

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Návrh a implementace webového rozhraní pro
telerehabilitaci

BAKALÁŘSKÁ PRÁCE

2021

Jan PETRŽÍLKA



Zadání bakalářské práce

Název:	Návrh a implementace webového rozhraní pro telerehabilitaci
Student:	RNDr. Jan Petržílka, Ph.D., MBA
Vedoucí:	doc. Ing. Patrik Kutílek, MSc., Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Cílem bakalářské práce je navrhnout a implementovat databázi a webovou aplikaci/rozhraní pro telerehabilitační systém. Backend webového rozhraní bude implementován v Javě. Přes rozhraní bude možné nahrát objemná data (například video) a bude možnost vyčkat na výsledky vypočtené výpočetním serverem. V rámci práce prozkoumejte současné softwarové technologie pro danou úlohu. Navrhněte vhodný databázový model pro danou problematiku. Navrhněte webový server, backend a frontend webové aplikace. Navrženou databázi i aplikaci implementujte a ověřte její funkčnost. Vytvořte testy ověřující funkčnost webové aplikace. Zdokumentujte implementovanou databázi a webovou aplikaci a vytvořte manuál k obsluze webové aplikace.

Bakalářská práce

NÁVRH A IMPLEMENTACE WEBOVÉHO ROZHRANÍ PRO TELEREHABILITACI

Jan Petržílka

Fakulta informačních technologií ČVUT v Praze
Katedra softwarového inženýrství
Vedoucí: doc. Ing. Patrik Kutílek, Ph.D.
Odborný konzultant: Ing. Jan Hejda, Ph.D.
5. května 2021

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Jan Petržílka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bez uplatněných zákonných licencí nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Petržílka, Jan. *Návrh a implementace webového rozhraní pro telerehabilitaci*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Obsah

Poděkování	xi
Prohlášení	xii
Abstrakt	xiii
Shrnutí	xiv
Seznam zkratk a pojmů	xv
1 Cíl práce	1
2 Architektura řešení	3
3 Řešení specifických požadavků	5
3.1 Dynamické generování formulářů	5
3.1.1 Využití Angularu pro odeslání souboru	5
3.1.2 Využití Angularu pro pořízení videonahrávky	6
3.1.3 Přímé využití základních API	6
3.1.4 Generovaný front end	6
3.2 Přenos souboru	7
3.2.1 Teoretická východiska	7
3.2.2 Varianty řešení	11
3.2.3 Implementované řešení	12
3.3 Pořízení videonahrávky	23
3.3.1 Technické možnosti	23
3.3.2 Implementované řešení	26
4 Projektový postup	31
4.1 Analýza požadavků	31
4.2 Implementace řešení	31
5 Popis vypracovaného řešení	33
5.1 Popis řešení	33
5.1.1 Popis datového modelu	33
5.1.2 Architektura aplikace	41
5.1.3 Využití dědičnosti	48
5.1.4 Případy užití	53
5.1.5 Popis zajištění bezpečnosti	56
5.1.6 Testování	59
5.2 Demonstrační instalace	62
5.3 Projektová dokumentace	62
5.3.1 Uživatelská příručka	64
5.3.2 Instalační příručka	64

5.3.3	Návrh specifikace	64
5.4	Licenční požadavky	64
5.4.1	Vývojové a pomocné nástroje	64
5.4.2	Vlastní aplikace	65
5.4.3	Komponenty uživatelského rozhraní	65
5.4.4	Implementace nahrávání videa	66
5.4.5	Tipy a triky	66
5.4.6	Nepoužité komponenty	66
6	Závěr	67
	Literatura	68
A	Souhlas s užitím kódu	77
B	Uživatelská příručka	81
B.1	Společné stránky	82
B.1.1	Úvodní stránka	82
B.1.2	Přihlašovací stránka	82
B.1.3	Stránka pro obnovení hesla	83
B.1.4	Stránka pro nastavení hesla	84
B.1.5	Stránka pro změnu hesla	84
B.2	Stránky uživatele s rolí klient	85
B.2.1	Stránka „Údaje“	85
B.2.2	Stránka „Nová úloha“	86
B.2.3	Stránka „Úlohy“	89
B.3	Stránky uživatele s rolí terapeut	91
B.3.1	Stránka „Úlohy“	91
B.3.2	Stránka „Nová úloha“	92
B.3.3	Stránka „Klienti“	92
B.3.4	Stránka „Nový klient“	95
B.4	Stránky uživatele s rolí správce	95
B.4.1	Stránka „Nový uživatel“	96
B.4.2	Stránka „Přehled uživatelů“	97
B.4.3	Stránka „Změna terapeuta“	98
B.4.4	Stránka „Nová úloha“	99
B.4.5	Stránka „Přehled úloh“	99
B.4.6	Stránka „Konfigurace“	100
B.4.7	Stránka „Šablony úloh“	104
B.4.8	Stránka „Lokalizace“	109
B.4.9	Stránka „Log“	111
B.4.10	Stránka „Test přenosu dat“	112
C	Instalační příručka	115
C.1	Instalace aplikace	115
C.1.1	Instalace databáze	115
C.1.2	Instalace Java Runtime Environmentu	116
C.1.3	Spuštění aplikace	116
C.1.4	Reverzní proxy	117
C.1.5	Logování	118
C.1.6	Sestavení aplikace	118
C.2	Administrace aplikace	118
C.2.1	Přístupové údaje ke službě reCaptcha	118

C.2.2	Přístupové údaje k e-mailovému serveru	118
C.2.3	Přenos souborů	119
D	Návrh specifikace	133
D.1	Obsah dokumentu	134
D.1.1	Seznam verzí	134
D.1.2	Otevřené body	135
D.2	Zadání práce	135
D.3	Předmět práce (manažerské shrnutí)	135
D.4	Funkční požadavky	136
D.4.1	Případy užití	136
D.4.2	Struktura uživatelů	151
D.4.3	Nepodporované funkčnosti	152
D.4.4	Datový model	153
D.5	Technické, servisní a nefunkční požadavky	154
D.5.1	Bezpečnostní požadavky	154
D.5.2	Velikost datových souborů	154
D.5.3	Podporované prohlížeče	156
D.5.4	Komponenta pro nahrávání videa	156
D.5.5	Výběr webové kamery	156
D.5.6	Generování uživatelského rozhraní	156
D.5.7	E-mailový SMTP server	157
D.5.8	Integrační testování	157
D.5.9	Lokalizace	157
D.5.10	Migrace dat	157
D.6	Výstupy práce	157
D.7	Předpokládaný harmonogram	157
E	Obsah příloženého média	159
F	Kontaktní údaje	161

Seznam obrázků

2.1	Třívrstvá architektura webové aplikace	3
3.1	Metoda okénka	7
3.2	Supervýkonný modem O2 Smart Box	8
3.3	Změřená rychlost připojení přes modem O2	9
3.4	Standardní odeslání 5 MB přes modem O2	9
3.5	Ping na <code>www.fit.cvut.cz</code> během standardního odesílání dat	10
3.6	Ping na <code>www.fit.cvut.cz</code> během měření rychlosti	11
3.7	Řízení dat souboru	13
3.8	Rozdělení souboru na sektory	14
3.9	Stavy sektoru souboru	15
3.10	Řízení přenosu dat	15
3.11	Zjednodušený náčrt toku dat	16
3.12	Aplikační odeslání 5 MB přes modem O2	17
3.13	Ping na <code>www.fit.cvut.cz</code> během aplikačního odesílání dat	18
3.14	Aplikační odesílání dat 38 vteřin přes mobilní připojení T-Mobile	18
3.15	Změřená rychlost mobilního připojení T-Mobile	19
3.16	Aplikační odesílání dat 38 vteřin přes kabelové připojení Vodafone	19
3.17	Aplikační odesílání dat 38 vteřin přes kabelové připojení Vodafone se simulovaným zpožděním 1 vteřina	20
3.18	Změřená rychlost kabelového připojení Vodafone	20
3.19	Spuštění přenosu	22
3.20	Zastavení přenosu	22
3.21	Pořízení videonahrávky pomocí HTML Media Capture	23
3.22	Vývojový stupeň specifikace MediaStream Recording API	25
3.23	Zapnutí MediaRecording API v Safari na iPadu 2018	25
3.24	Požadované rozlišení videa	26
3.25	Reportování rozlišení videa	27
3.26	Proces výběru formátu videonahrávky	28
5.1	Lokalizační data	34
5.2	Data o šablonách	35
5.3	Data o úlohách	38
5.4	Pomocná data	40
5.5	Architektura aplikace	42
5.6	Použití dědičnosti u cachování	48
5.7	Použití dědičnosti u monitoringu	49
5.8	Použití dědičnosti pro metody Lazy vs. Eager	49
5.9	Dědičnost nastavení filtru	50
5.10	Dědičnost filtrování a stránkování	51
5.11	Dědičnost řazení položek	52
5.12	Deduplikace kódu pomocí dědičnosti	52
5.13	Diagram případů užití uživatele v roli klient	53

5.14	Diagram případů užití uživatele v roli terapeut	54
5.15	Diagram případů užití uživatele v roli správce	55
5.16	SonarQube – nastavení „FIT way“	59
5.17	SonarQube – výstup fakultní instalace	60
5.18	SonarQube – nastavení lokální instalace	61
5.19	SonarQube – výstup lokální instalace	61
5.20	Počet zapsání (commit) nového kódu v čase	63
5.21	Růst počtu řádek zdrojového kódu v čase	63
5.22	Odkazy na zdroje v kódu aplikace	66
A.1	Souhlas s užitím kódu	79
B.1	Úvodní stránka aplikace	82
B.2	Přihlašovací stránka	83
B.3	Obnovení hesla	83
B.4	E-mail s linkem na obnovení hesla	84
B.5	Nastavení hesla	84
B.6	Změna hesla	85
B.7	Editace osobních údajů	86
B.8	Výběr typu úlohy	86
B.9	Formulář úlohy	87
B.10	Pořízení videonahrávky	88
B.11	Přehled úloh klienta	89
B.12	Vyhodnocení úlohy klienta	90
B.13	Zpráva o vyhodnocení úlohy	91
B.14	Přehled úloh klientů terapeuta	92
B.15	Přehled klientů terapeuta	92
B.16	Editace údajů klienta	93
B.17	Přihlášení terapeuta jako klienta	94
B.18	Odeslání linku na úlohu	94
B.19	Link na provedení úlohy bez přihlášení	95
B.20	Založení nového uživatele	96
B.21	Link na nastavení prvního hesla	97
B.22	Přehled uživatelů	97
B.23	Změna terapeuta	99
B.24	Přehled úloh z pohledu správce	99
B.25	Konfigurace – část 1	101
B.26	Konfigurace – část 2	102
B.27	Konfigurace – část 3	103
B.28	Aplikace s vypnutou lokalizací	103
B.29	Šablony úloh	104
B.30	Šablona úlohy	107
B.31	Editace komponenty šablony	108
B.32	Html stránky	109
B.33	Editace html stránky	110
B.34	Log	111
B.35	Test přenosu dat	113
C.1	Configurace reverzní proxy	117
C.2	Schéma souboru clientData.json	123
C.3	Schéma souboru taskData.json	128
C.4	Příklad souboru clientData.json	129
C.5	Příklad souboru taskData.json	130

C.6	Příklad souboru resultData.json	131
D.1	Případy užití klienta	136
D.2	Aplikační menu klienta	137
D.3	Editace údajů klienta	137
D.4	Změna hesla	138
D.5	Výběr typu úlohy	138
D.6	Plnění úlohy	139
D.7	Přehled úloh	140
D.8	Detail úlohy	141
D.9	Případy užití terapeuta	142
D.10	Aplikační menu terapeuta	143
D.11	Vytvoření nového klienta	143
D.12	Přehled klientů	144
D.13	Přehled úloh klientů terapeuta	145
D.14	Zobrazení úlohy terapeutem	146
D.15	Případy užití správce	147
D.16	Aplikační menu správce	147
D.17	Přehled uživatelů	148
D.18	Hromadná změna terapeuta	149
D.19	Přehled úloh	149
D.20	Definice úloh pro klienty	150
D.21	Definice úlohy	151
D.22	Nastavení systému	152
D.23	Logický datový model	153
D.24	Fyzický datový model	154
D.25	Přihlášení uživatele	155
D.26	Žádost o obnovu hesla	155
D.27	Ilustrace ke správě šablon	156

Seznam tabulek

3.1	Přehled aplikačního přenosu dat	21
3.2	Ověření přenosu na dálku	21
3.3	Vývojové stupně dle W3C	25
5.1	Lokalizační data	33
5.2	Pole tabulky html_page	33
5.3	Pole tabulky html_element	34
5.4	Pole tabulky html_element_text	34
5.5	Data o šablonách	36
5.6	Pole tabulky task_template	36
5.7	Pole tabulky task_template_name	36
5.8	Pole tabulky task_template.component	36
5.9	Pole tabulky task_template.component_name	36
5.10	Pole tabulky task_template.component_detail	37

5.11	Data o úlohách	39
5.12	Pole tabulky user	39
5.13	Pole tabulky response	39
5.14	Pole tabulky response_name	40
5.15	Pole tabulky link	40
5.16	Pomocná data	41
5.17	Pole tabulky config	41
5.18	Pole tabulky log	41
5.19	Balíčky tříd	43
5.20	Obsah balíčku anyone	43
5.21	Obsah balíčku client	43
5.22	Obsah balíčku therapist	44
5.23	Obsah balíčku admin	44
5.24	Obsah balíčku service	44
5.25	Obsah balíčku support	45
5.26	Obsah balíčku model	46
5.27	Obsah balíčku dao	46
5.28	Obsah balíčku repository	47
5.29	Obsah balíčku dto	47
5.30	Obsah balíčku config	47
5.31	Přiřazení systémových rolí	56
5.32	Hlavní adresy stránek aplikace nebo významných služeb	57
5.33	Připomínky z uživatelského testování	62
B.1	Stavy uživatelů	81
B.2	Filtrování úloh	89
B.3	Filtrování klientů	93
B.4	Filtrování uživatelů	98
B.5	Systémová nastavení	100
B.6	Filtrování šablon	105
B.7	Atributy šablon	105
B.8	Atributy komponent	106
B.9	Typy komponent šablony	106
B.10	Filtrování logů	111
B.11	Atributy testu přenosu dat	112
C.1	Výchozí přístupové údaje	116
C.2	Připojení školního e-mailového serveru	119
C.3	Výstupní soubory	119
C.4	Vstupní soubory	120
C.5	Typy komponent šablony	120
D.1	Změny implementace oproti návrhu specifikace	134
D.2	Seznam verzí	135
D.3	Otevřené body	135
D.4	Výstupy práce	157
D.5	Harmonogram práce	158

Chtěl bych poděkovat vedoucímu práce doc. Ing. Patriku Kutílkovi, Ph.D., za to, že mně umožnil vypracování této práce. Děkuji odbornému konzultantovi Ing. Janu Hejdovi, Ph.D., za diskuse o obsahu práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. května 2021

.....

Abstrakt

Obsahem práce je vypracování webové aplikace pro telerehabilitaci dle požadavků vedoucího práce. Aplikace umožňuje uživateli s rolí správce definovat šablony formulářů a uživateli s rolí klient tyto šablony použít k vložení informací o provedení úkolu. Šablona může obsahovat videonahrávku. Aplikace postoupí vyplněný formulář se souborovými přílohami ke zpracování návazné aplikaci, která není předmětem této práce, a klientovi zobrazí informace o výsledku zpracování. Uživatelé v roli terapeut mají přiřazené uživatele v roli klient a mají přístup k úkolům svých klientů. Aplikace nad rámec obvyklých požadavků obsahuje funkčnosti pro pořízení videonahrávky s využitím Media Recording API a pro přenos velkých souborů o velikosti stovek MB z webového prohlížeče na server. Pro zabránění zahlcení linky při přenosu dat na server je použit vlastní algoritmus řízení toku dat. Uživatelské rozhraní je provedeno česky a anglicky s plně parametrizovanou lokalizací. Serverová část je implementovaná v Javě, jako databáze je použit PostgreSQL.

Klíčová slova webová aplikace, telerehabilitace, webový přenos souborů, zahlcení linky, řízení toku dat, pořízení videonahrávky, Media Recording API, jazyková lokalizace

Abstract

The content of the work is the development of a web application for telerehabilitation according to the requirements of the thesis supervisor. The application allows a user with the administrator role to define templates, and a user with the client role can use these templates to enter information about a task. The template may contain a video recording. The application forwards the completed form with file attachments for processing to a related application, which is not the subject of this work, and displays information to the client about the result of processing. Users with the therapist role have assigned clients and they have access to their tasks. In addition to the usual requirements, the application includes functionality for video recording based on Media Recording API and for transferring large files of size of hundreds of MB from a web browser to a server. A custom data flow control algorithm to avoid congestion collapse is used to transfer data to the server. The user interface is made in Czech and English with fully parameterized localization. The server part is implemented in Java, PostgreSQL is used as the database.

Keywords web application, telerehabilitation, web file transfer, congestion collapse, data flow control, video recording, Media Recording API, language localization

Motivace

Odbornou motivací mojí práce bylo v rámci bakalářské práce dodat funkční webovou aplikaci se serverovou částí implementovanou v jazyce Java podle požadavků zadavatele a s využitím znalostí a postupů získaných během studia. Mojí osobní motivací pak bylo využití mého studijního úsilí k praktickému účelu. Výběr tématu jsem volil tak, abych mohl tento praktický účel předpokládat.

Cíl práce

Cílem práce je vytvořit webovou aplikaci podle požadavků zadavatele a vyřešit problémy související s následujícími požadavky, které překračují rámec obvyklé funkčnosti webové aplikace:

- Pořízení videonahrávky v prohlížeči
- Přenos velkého souboru z prohlížeče na server

Těmto cílům také odpovídá struktura a obsah práce – v práci popisují způsob řešení specifických požadavků uvedených výše, projektový postup a výslednou aplikaci.

Postup

V době zadání práce měl vedoucí práce ucelenou představu o většině hlavních případů užití. V průběhu řešení jsem se proto soustředil zejména na popis požadavků ve formě návrhu specifikace webové aplikace, vyřešení technických problémů a dodání požadovaného softwarového díla. Postup řešení jsem popsal v této práci.

Výsledky práce

Struktura obsahu práce odpovídá struktuře požadavků v zadání práce.

Výstupem práce jsou následující artefakty:

- Návrh specifikace webové aplikace
- Uživatelská příručka
- Instalační příručka s popisem souborů vyměňovaných s návaznou aplikací
- Aplikace a její demonstrační instalace na doméně telecam.cz

Tyto výstupy jsem v této práci doplnil následujícími popisy:

- Popis možností a řešení přenosu souborů
- Popis možností a řešení pořízení nahrávky
- Popis projektového postupu
- Popis datového modelu
- Popis struktury aplikace
- Popis zajištění bezpečnosti
- Popis implementovaných a provedených testů s výsledky

Jednotlivé popisy jsem sestavil co nejušporněji tak, aby obsahovaly výstupy podle požadavků zadavatele, dokumentovaly hlavní rysy aplikace a zachycovaly významné milníky, rozhodnutí a výsledky.

Závěr

Webovou aplikaci jsem implementoval dle požadavků zadavatele způsobem popsaným v této práci.

Seznam zkratek a pojmů

Obecné zkratky a pojmy

Cachování	Takový způsob práce s často používanými daty, jejichž příprava je výpočetně náročná, kdy aplikace jednou připravená data uchová pro příští použití a následně použije tato uložená data s cílem snížit dobu odezvy aplikace.
Captcha	Zkratka slovního spojení Completely Automated Public Turing test to tell Computers and Humans Apart; jedná se o test ověřující, že uživatelem není automat.
Cookies	Cookies jsou krátké textové soubory, jež jsou nezbytné pro správnou funkci internetových stránek.
Json	Zkratka slovního spojení JavaScript Object Notation; jedná se o formát textového souboru umožňující strukturovaně uložit objektová data.
GMT čas	Čas časové zóny Greenwich Mean Time, referenční čas, který se nemění přechodem na letní čas.
Google reCaptcha	Veřejně dostupná služba společnosti Google implementující Captcha test.
Lokalizace	Vlastnost aplikace umožňující zobrazení uživatelského rozhraní v různých jazycích.
Mb	1.000.000 bitů.
Mb/s	Jednotka rychlosti přenosu dat určující počet Mb za 1 vteřinu
MB	megabyte, 1.000.000 bytů.
REST služba	Služba využívající architektonický styl Representational state transfer umožňující vytvářet interaktivní aplikace.

Aplikačně specifické pojmy

element html stránky	Pole html stránky, které má definovanou hodnotu pro každý jazyk uživatelského rozhraní.
html stránka	Šablona stránky uživatelského rozhraní nebo jeho části používaná pro nastavení vícejazyčných textů. Jednotlivé texty jsou nastavení pomocí elementů html stránky.
komponenta šablony	Část šablony určující jedno pole formuláře pro úlohu.
šablona	Soubor polí používaný aplikací pro zobrazení formuláře. Šablona může být použita pro zobrazení formuláře s poli specifickými pro danou úlohu, pro zobrazení formuláře s poli s osobními údaji používanými pro úlohy nebo pro zobrazení výsledků zpracování úlohy.
úloha	Záznam aktivity uživatele s rolí klient prostřednictvím formuláře založeného na šabloně. Typ úlohy odpovídá typu šablony. Každá úloha spočívá ve vyplnění polí podle šablony úlohy a polí podle šablony pro osobní data klienta. Úlohy zpracovává jiná aplikace, která není předmětem této práce.

Kapitola 1

Cíl práce

Cílem práce je vytvořit webovou aplikaci podle požadavků zadavatele.

V době zadání práce měl zadavatel ucelenou představu o požadované funkčnosti aplikace. Tuto představu zde stručně popíši. Všechny zde popisované funkční aspekty jsem v aplikaci implementoval.

Aplikaci používají uživatelé v rolích správce, terapeut nebo klient. Ústředním případem užití je zobrazení formuláře s úlohou k vyplnění uživateli v roli klient. Tento formulář má dvě části – osobní údaje a údaje specifické pro konkrétní úlohu. Součástí polí specifických pro danou úlohu může být pořízení jedné nebo více videonahrávek. Klient tento formulář vyplní – případně pořídí požadované videonahrávky – a odešle na server.

Server přijatou úlohu spolu s přílohami postoupí ke zpracování další aplikaci. Výsledky zpracování jsou předány zpět aplikaci řešené v této práci a ta je zobrazí klientovi.

Klient může nezávisle na plnění úloh editovat svoje osobní údaje, které se následně použijí při vyplňování nejbližší následující úlohy.

Formuláře pro vyplňování úloh jsou založeny na šablonách, které správce aplikace může měnit – šablona obsahuje pole různých typů s definovaným pořadím. Jednotlivá pole mohou být nastavená jako povinná a/nebo jako předvyplňovaná podle osobních údajů nebo předchozích provedení úlohy týměž uživatelem.

Specifickými požadavky bylo

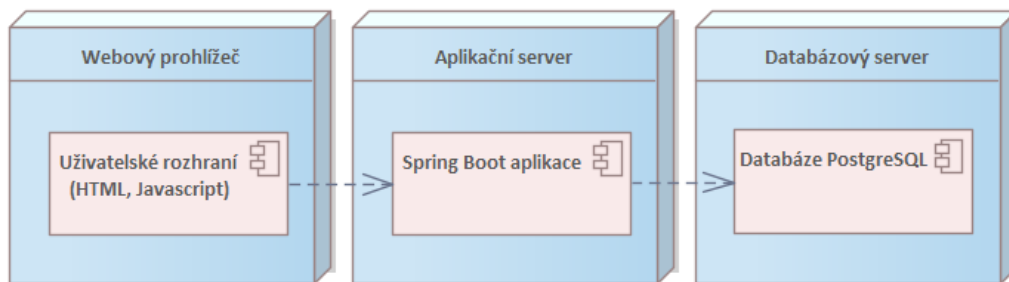
- vyřešit pořízení videonahrávky (nebo nahrávek) v internetovém prohlížeči tak, aby si klient mohl nahrávku (nebo nahrávky) před odesláním prohlédnout a teprve potom odeslat
- a zajistit spolehlivý přenos těchto nahrávek v předpokládané velikosti stovek MB na webový server.

Videonahrávka je pořizována v požadovaném minimálním rozlišení 1280 x 1024 bodů se snímkovací frekvencí 25 snímků za vteřinu.

Bylo požadováno, aby alespoň klientská část aplikace byla provedena česky a anglicky.

Architektura řešení

Struktura webové aplikace, která je výstupem této práce, vychází ze standardní třívrstvé architektury zachycené na obrázku 2.1. Uživatel používá uživatelské rozhraní ve webovém prohlížeči. Uživatelské rozhraní používá REST služby aplikačního serveru. Aplikační server zpracovává data podle implementované aplikační logiky. Data jsou uložena v databázi.



■ **Obrázek 2.1** Třívrstvá architektura webové aplikace

Technické možnosti implementace specifických požadavků kladených na uživatelské rozhraní jsem spolu se zvoleným řešením popsal v kapitole 3.

Jako aplikační server jsem použil Tomcat [1] zapouzdřený ve Spring Boot [2] aplikaci. Hlavním důvodem mého rozhodnutí bylo to, že toto řešení nabízí potřebnou funkčnost a současně usnadňuje nasazení a provozování aplikace, neboť nevyžaduje samostatnou instalaci a správu aplikačního serveru. Dalším důvodem byla moje zkušenost s tímto řešením z předmětů „Softwarový projekt 1“ a „Softwarový projekt 2“, o kterou jsem se mohl opřít. Alternativně jsem uvažoval o aplikačních serverech GlassFish [3] a WildFly [4] nebo o frameworku Quarkus [5], ale použití ani jedné z těchto možností by dle mého názoru nebylo oproti použití serveru Tomcat s frameworkem Spring Boot výhodnější.

Databází je PostgreSQL [6]. Podobně jako v případě aplikačního serveru stála moje volba zejména na mých zkušenostech z výše uvedených předmětů a dalších projektů. Alternativou bylo využití databáze MySQL [7]. Použití obou těchto databází by dle mého názoru bylo pro požadovanou aplikaci vhodné, v případě MySQL by se ale jednalo o moje první použití této databáze ve spojení s Javou, což by představovalo zbytečné riziko.

Řešení specifických požadavků

Dva požadavky na funkčnost aplikace, a totiž (1) schopnost pořídit videonahrávku v prohlížeči uživatele v požadovaném rozlišení a kódování, umožnit uživateli tuto nahrávku shlédnout a (2) následně odeslat na server, vybočují z běžných funkcností webových aplikací a vyžadovaly proto hlubší analýzu. Tuto analýzu představuji v této kapitole.

Před rozborem přenosu dat a pořízení videonahrávky uvádím ještě poznámky k rozhodnutí o způsobu, jakým bude aplikace generovat formuláře pro provádění úloh. Tyto tři aspekty se totiž ovlivňují navzájem a pro úspěšné řešení tématu je potřeba jejich řešení navzájem přizpůsobit.

3.1 Dynamické generování formulářů

Při zahájení analytické fáze práce jsem vycházel z předpokladu, že nahrání souboru na server a pořízení videonahrávky bude zajištěno triviálním voláním nějaké komponenty použitého frameworku a že těžiště analytického úsilí bude spočívat v nalezení nejvhodnější datové reprezentace formulářů pro provádění úlohy a jejich zobrazení v prohlížeči uživatele. Předpokládal jsem, že frontendová část aplikace bude napsána s využitím frameworku Angular [8], se kterým jsem měl minimální zkušenosti ze semestrální práce z předmětu „Enterprise java“, a očekával jsem, že svoje zkušenosti rozšířím.

3.1.1 Využití Angularu pro odeslání souboru

Angular používání dynamických formulářů podporuje [9] a zbývá tedy vyřešit odeslání souborů a pořízení videonahrávky. Použití Angularu znamená psát kód uživatelského rozhraní v Type Scriptu [10], [11]. To může otevřít cestu k využití již hotových komponent, které uvažovanou úlohu spolehlivě vyřeší standardním způsobem.

Pro odeslání souboru s využitím Angularu / Type Scriptu je možné podle [12] použít například komponentu „angular-file-uploader“ [13], která je standardem v této oblasti. Vzhledem ke svým minulým profesním zkušenostem jsem při používání cizích komponent k zajištění klíčových aplikačních funkcností mimořádně obezřetný. Je potřeba mít přiměřenou míru jistoty, že použitá komponenta bude fungovat i pro uvažovaný případ, a že i v budoucnosti bude zachována její kompatibilita s ostatními požadavky.

Uvedená komponenta pro odeslání souboru je navržena pro odeslání souboru ze souborového systému, v řešeném případě ale bude potřeba odeslat data uložená přímo v prohlížeči. Řešení nepochybně existuje [14], ale způsob použití komponenty k účelu, pro něj není běžně používána, může být zdrojem problémů.

Výchozí nastavení komponenty uvádí maximální velikost souboru 20 MB. V řešeném případě je potřeba přenášet soubory o řád větší. Samozřejmě je možné maximální velikost souboru změnit, ale opět se tak dostanu do situace, kdy komponentu používám mimo její obvyklé limity.

Navíc jsem si nebyl jistý, jak se komponenta (nebo její budoucí verze) vyrovná s problémem popsaným v odstavci 3.2.1.1, resp. jsem předpokládal, že se bude chovat jako referenční řešení, které pro uvažované použití není vhodné.

Výsledkem předběžné analýzy byl závěr, že řešení tímto způsobem je možné, ale že riziko implementačních nebo budoucích problémů je značné.

3.1.2 Využití Angularu pro pořízení videonahrávky

V případě pořízení videonahrávky byla situace obdobná jako v případě odeslání souboru. Existuje například komponenta „ngx-webcam“ [15] umožňující zobrazit obraz a pořídít snímek z kamery s využitím „Media Capture and Streams API (Media Stream)“ [16], ale toto API je primárně určeno pro pořízení streamu z kamery a jeho přímý přenos na jiné zařízení. Pravděpodobně existuje cesta, jak úpravou implementace pořídít videonahrávku (viz například [17] a [18]), ale rozhodně se nejedná o přímočaré řešení.

Jiná komponenta „extendable-media-recorder“ [19] založená na Type Scriptu v určitém rozsahu zpřístupňuje „MediaStream Recording API“ nakonec použité v implementaci aplikace, ale jedná se o další mezivrstvu s určitými vlastnostmi, závislostmi na dalších komponentách, chybami nebo licenčními požadavky.

Podobně jako v případě přenosu dat, je také pořízení videonahrávky klíčovou funkcí aplikace a je zde riziko, že použití nějaké komponenty by mohlo omezit implementaci současných nebo budoucích požadavků zadavatele a provozovatele aplikace.

3.1.3 Přímé využití základních API

Na základě analytických úvah popsaných výše jsem dospěl k názoru, že využití Angularu nebo komponent, které využívají API prohlížeče k přenosu dat nebo pořízení nahrávky, do projektu přináší zbytečnou míru komplexity a neakceptovatelné riziko nedodání řešení v termínu.

Aplikace proto nepoužívá k odeslání dat a pořízení videonahrávky žádné dodatečné komponenty a je založena na přímém využití javascriptových API internetového prohlížeče.

3.1.4 Generovaný front end

Od rozhodnutí nepoužít Angular a Type Script se bezprostředně odvíjí související rozhodnutí generovat frontendovou část aplikace s využitím aplikační logiky serverové části. Frontendová část aplikace je proto aplikačně generovaná serverovou částí aplikace způsobem inspirovaným technologií „JSP pages“ [20], přestože takové řešení znamená, že případné změny uživatelského rozhraní mohou vyžadovat nové sestavení serverové části aplikace a proto toto řešení může být považováno za technický dluh [21]. Šablony html stránek a jejich částí jsou v aplikaci uloženy jako resources s jazykově specifickými poli, jejichž hodnota je pro jednotlivé jazykové varianty uživatelského rozhraní uložena v aplikační databázi a je definovaná uživatelem s rolí správce.

Moje výše uvedená rozhodnutí také odrážejí moji pouze základní zkušenost s Angularem a Type Scriptem a dále obava, že kombinace různých komponent a eventuální nutnost jejich úpravy nebo doplnění vlastním kódem v Javascriptu by mohla dále zvýšit složitost řešení. Je možné, že kdybych měl větší zkušenostmi s Angularem a Type Scriptem, tak bych se rozhodl jinak.

3.2 Přenos souboru

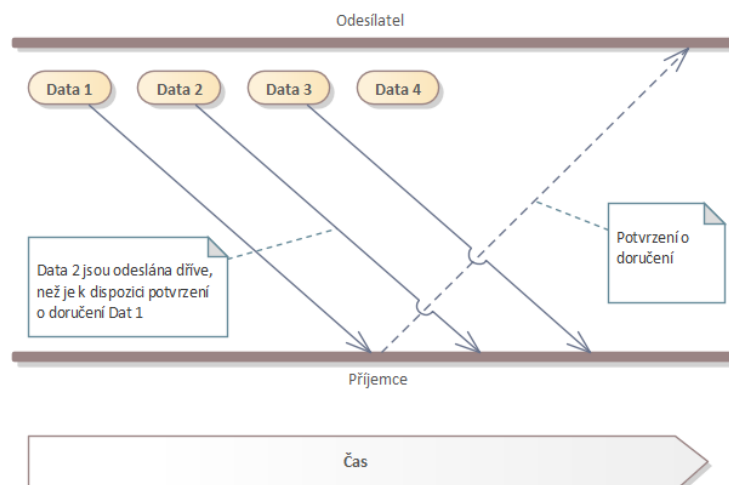
Aplikace musí být schopna přenést z prohlížeče uživatele soubor o velikosti řádově stovek megabytů. Jedná se o klíčovou funkčnost aplikace, při jejímž selhání aplikace ztrácí svoji použitelnost. Jedná o přenos videonahrávky pořízené přímo prohlížečem, může se tedy jednat o neopakovatelná nebo obtížně reprodukovatelná data.

Pro úspěšné řešení jsem stanovil následující kritéria:

- Vzhledem k tomu, že aplikaci po dokončení práce předám provozovateli a nebudu ji nadále podporovat nebo udržovat, musí být řešení co nejjednodušší.
- Řešení musí poskytovat dostatečnou míru interaktivity pro uživatele i na linkách s nízkou rychlostí přenosu.
- Řešení musí využívat dostupnou přenosovou kapacitu linky, ale nesmí ji přenášenými daty zahltil. Je důležitější, aby řešení fungovalo spolehlivě na linkách s malou rychlostí přenosu a dlouhou dobou odezvy, než aby byla využita plná kapacita rychlých linek.
- Vzhledem k očekávaným dobám přenosu i desítek minut musí být možné přenos přerušit a obnovit a řešení si musí poradit i s přerušením spojení.
- Přenos dat bude probíhat výhradně v uživatelském prohlížeči – není tedy potřeba, aby všechny situace byly řešeny automaticky, a je možné počítat s tím, že uživatel může stav přenosu korigovat.

3.2.1 Teoretická východiska

Základní metodou první volby pro přenos většího množství dat je metoda okénka (sliding window method) [22], [23]. Základním principem této metody, zachyceným na obrázku 3.1, je to, že odesílatel odesílá další datové rámce ještě před přijetím potvrzení o dokončení přenosu rámců předchozích. Pokud úspěšný přenos některého z rámců není potvrzen, je pokus o přenos opakován. Šíře okénka určuje maximální počet datových rámců, které mohou být odeslány bez přijetí potvrzení o doručení předchozích rámců.



■ Obrázek 3.1 Metoda okénka

Tato metoda funguje za podmínky, že je zajištěno, že při přenosu nedojde k zahlcení žádného prvku na trase mezi odesílatelem a příjemcem. Aby k tomu nedošlo, používá každý aktivní síťový prvek nějaký způsob řízení toku dat – a zajistí tak, že prvek neodesílá data v případě, kdy následující prvek není schopen data přijmout a zpracovat. Tento mechanismus je v rozdělení podle referenčního ISO/OSI modelu [24] zajištěn linkovou vrstvou na linkové úrovni pro jednotlivé části spojení a dále transportní vrstvou na úrovni komunikace mezi odesílatelem a příjemcem.

Při odesílání dat z prohlížeče máme k dispozici pouze aplikační vrstvu založenou na TCP/IP protokolu transportní vrstvy (protokol UDP není v prohlížeči k dispozici, viz například [25]). Aplikace musí řídit odesílání dat takovým způsobem, aby data mohla být linkou průběžně přenášena s využitím její dostupné kapacity (možná technická řešení diskutují v části 3.2.2).

Před návrhem řešení jsem ověřil funkčnost běžných aplikací používaných k odesílání souborů podobné velikosti jako v případě připravované aplikace přes různá v ČR běžně dostupná připojení k Internetu:

- Kabelové připojení UPC/Vodafone.
- Mobilní připojení poskytované operátorem O2, Vodafone nebo T-Mobile prostřednictvím modemu v mobilním telefonu nebo USB modemu.
- Běžné VDSL připojení pomocí standardního modemu VDSL2 – modemu Zyxel VMG3312-T20A [26].
- Prémiové VDSL připojení pomocí prémiového „Supervýkonného modemu O2 Smart Box“ na obrázku 3.2 převzatého z webu společnosti O2 [27].

Výsledkem bylo zjištění, že na spolehlivé řízení přenosu dat na lince v odchozím směru nelze spoléhat a že bude potřeba vyřešit problém popsany níže.

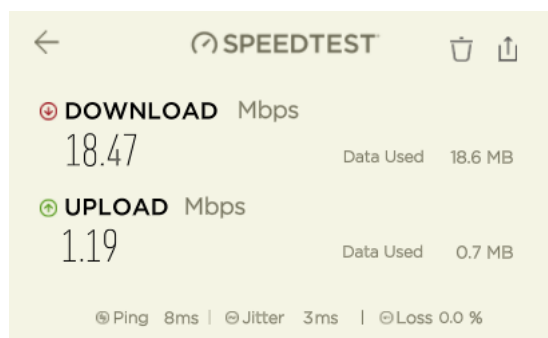


■ **Obrázek 3.2** Supervýkonný modem O2 Smart Box, obrázek převzatý z [27]

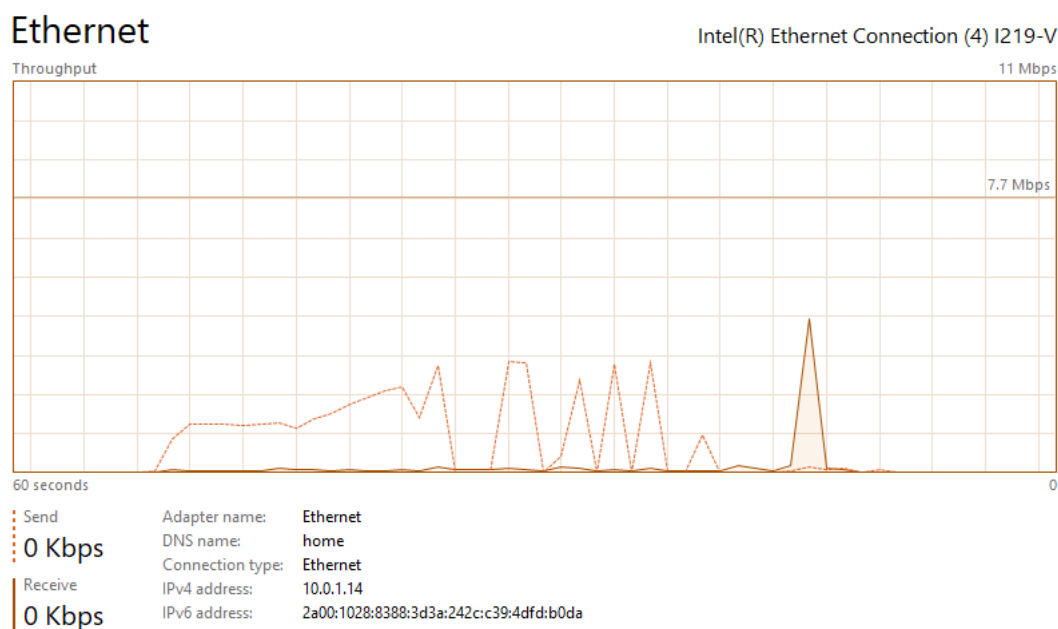
3.2.1.1 Zahlcení linky

Data prezentovaná níže byla pořízena na notebooku Lenovo T480 s operačním systémem Microsoft Windows 10 Professional verze 10.0.19042 Build 19042, kdy interní síťová karta Intel(R) Ethernet Connection (4) I219-V byla připojena kabelem přímo k routeru O2 Smart Box. Žádný další uživatel router současně nepoužíval. Router zpřístupňoval VDSL připojení společnosti O2 s obchodním názvem „Internet HD Bronzový“ s deklarovanou rychlostí připojení 20/2 Mb/s. Skutečná rychlost připojení v době měření je zachycena na obrázku 3.3 [28]. Z prohlížeče Chrome verze 89.0.4389.114 (Official Build) (64-bit) byl odeslán soubor o velikosti 5.086.580 bytů do úložiště Google Drive v rámci placeného tarifu G Suite Business. Obdobné chování modemu O2

Smart Box lze snadno reprodukovat také v jiné konfiguraci – s jinou aplikací a s jiným počítačem nebo mobilním zařízením.



■ **Obrázek 3.3** Změřená rychlost připojení přes modem O2



■ **Obrázek 3.4** Standardní odeslání 5 MB přes modem O2

Na obrázku 3.4 vidíme průběh odesílání dat tak, jak byl zaznamenán aplikací Task Manager operačního systému Windows 10. Po počátečním hladkém průběhu dochází k nárůstu odesílaných dat nad kapacitu linky následovaném výraznými oscilacemi. Na konci přenosu je stránka prohlížeče aktualizována a na záznamu se tato aktualizace projeví jako příchozí data, která jsou vyznačena tmavou výplní pod křivkou. Z celkové doby přenosu dat usuzuji, že odesílání dat modemem O2 Smart Box celou dobu probíhalo rychlostí přibližně 1,15 až 1,25 Mb/s, která odpovídá rychlosti linky. Pozorování nesevědí o tom, že by při přenosu dat docházelo ve větší míře k jejich ztrátě a nutnosti jejich přenos opakovat. Údaje je potřeba interpretovat v kontextu doby odezvy serveru www.fit.cvut.cz, která byla paralelně měřena stejným připojením a je zachycena na obrázku 3.5. Jednotlivé zprávy o vypršení doby čekání na odpověď jsou vypisovány v intervalu přibližně 5 vteřin.

Domnívám se, že pozorovaná data při použití uvedené konfigurace svědčí o selhání řízení toku dat a následném zahlcení vyrovnávací paměti modemem nebo jeho dalších mechanismů pro řízení

toku dat. (Při použití modemu Zyxel VMG3312-T20A na stejné lince v jinak stejné konfiguraci měření popisovaný jev nenastal.) Přenos dat sice probíhá, efektivně je ale znemožněna paralelní komunikace. Z pohledu těchto dalších komunikací je to stav podobný stavu, který je podle [23] popisován jako „congestion collapse“. Pokud paralelně probíhá například internetový hovor, je přerušen. V případě déle trvajících přenosu dat a paralelní komunikaci více aplikací využívajících dostupnou kapacitu linky dokonce často dochází k úplnému rozpadu spojení.

```
Reply from 2001:718:2:2908::212: time=8ms
Reply from 2001:718:2:2908::212: time=8ms
Reply from 2001:718:2:2908::212: time=481ms
Reply from 2001:718:2:2908::212: time=604ms
Reply from 2001:718:2:2908::212: time=704ms
Reply from 2001:718:2:2908::212: time=798ms
Reply from 2001:718:2:2908::212: time=874ms
Reply from 2001:718:2:2908::212: time=956ms
Reply from 2001:718:2:2908::212: time=1106ms
Reply from 2001:718:2:2908::212: time=1127ms
Reply from 2001:718:2:2908::212: time=1449ms
Reply from 2001:718:2:2908::212: time=2141ms
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 2001:718:2:2908::212: time=3142ms
Reply from 2001:718:2:2908::212: time=18ms
Reply from 2001:718:2:2908::212: time=8ms
Reply from 2001:718:2:2908::212: time=8ms
```

■ **Obrázek 3.5** Ping na `www.fit.cvut.cz` během standardního odesílání dat

Na obrázku 3.13 je doba paralelně měřené odezvy serveru `www.fit.cvut.cz` v průběhu měření rychlosti. Na průběhu doby odezvy vidíme dvě zvýšení – první odpovídá testu stahování, druhé testu odesílání. Test odesílání byl příliš krátký, aby se efekt mohl projevit naplno, ale i zde vidíme růst doby odezvy až na 1,6 vteřiny.

```
Reply from 2001:718:2:2908::212: time=7ms
Reply from 2001:718:2:2908::212: time=8ms
Reply from 2001:718:2:2908::212: time=52ms
Reply from 2001:718:2:2908::212: time=56ms
Reply from 2001:718:2:2908::212: time=60ms
Reply from 2001:718:2:2908::212: time=56ms
Reply from 2001:718:2:2908::212: time=68ms
Reply from 2001:718:2:2908::212: time=70ms
Reply from 2001:718:2:2908::212: time=72ms
Reply from 2001:718:2:2908::212: time=87ms
Reply from 2001:718:2:2908::212: time=9ms
Reply from 2001:718:2:2908::212: time=9ms
Reply from 2001:718:2:2908::212: time=9ms
Reply from 2001:718:2:2908::212: time=741ms
Reply from 2001:718:2:2908::212: time=1147ms
Reply from 2001:718:2:2908::212: time=1385ms
```

```
Reply from 2001:718:2:2908::212: time=1601ms
Reply from 2001:718:2:2908::212: time=840ms
Reply from 2001:718:2:2908::212: time=8ms
Reply from 2001:718:2:2908::212: time=8ms
```

■ **Obrázek 3.6** Ping na `www.fit.cvut.cz` během měření rychlosti

Při použití Supervýkonného modemu O2 Smart Box ke kontinuálnímu odesílání dat roste doba odezvy při potvrzování odeslaných dat způsobem, který standardní odesílatel patrně (jedná se o čirou spekulaci) neinterpretuje jako hrozící zahlcení, ale jako prosté prodloužení doby odezvy, na což reaguje rozšířením posuvného okénka a zvýšením odchozího datového toku, které vyústí v chování popsané výše. Aplikační kód tedy v tomto případě nesmí spoléhat na standardní řízení toku dat, musí sám detekovat prodlužování doby odezvy a včas datový tok omezit, protože uživatelé připravované aplikace mohou používat pro připojení k Internetu právě toto zařízení na lince s podobnými vlastnostmi.

Nezbývá mi tedy nic jiného, než pomocí javascriptového kódu implementovat řízení datového toku, které zabrání vzniku popsaného jevu. Jedinou snadno měřitelnou veličinou, o kterou se při návrhu algoritmu mohu opřít, je doba mezi odesláním dat a přijetím potvrzení o jejich doručení. Podle [29] se jedná o dobu obrátky (RTT, round trip time).

Standardní způsob řešení je zhruba popsán ve [23] a spočívá v dynamickém snižování počtu odeslaných datových rámců (zužování posuvného okénka) v případě prodlužování doby odpovědi příjemce (resp. neobdržení odpovědi do určeného času) za současného prodlužování čekání na odpověď („Multiplicative Decrease Congestion Avoidance“). V okamžiku uvolnění linky pak dochází pozvolna k opačnému procesu („slow-start recovery“).

V tomto případě bohužel nemáme přístup přímo k datům na jiné než TCP/IP úrovni. Aplikační kód na obou stranách spojení získává o situaci pouze zprostředkované informace, které jsou filtrovány celou kaskádou vyrovnávacích pamětí podél přenosové linky. Navíc aplikační data odeslaná pomocí TCP/IP protokolu – pokud nedojde k úplnému rozpadu spojení – nakonec k cíli dorazí, a odesílatel nemá informace o tom, kolik pokusů na přenos bylo spotřebováno.

3.2.2 Varianty řešení

Komunikace frontendové části aplikace se serverem může probíhat pomocí HTTP požadavků [30], [31] nebo s využitím rozhraní WebSockets [32], [33], [34]. V případě komunikace pomocí HTTP požadavků mohou být data přenášena v těle požadavku v textové podobě nebo jako binární soubor s minimálním objemem dat přenášených navíc kromě datového obsahu souboru [30], [35].

Hlavním architektonickým rozdílem http požadavků a WebSocketů je to, že WebSokety umožňují oboustrannou komunikaci, čili že server může prohlížeči uživatele odeslat zprávu mimo pořadí požadavků z prohlížeče. To by eventuálně šlo využít k pravidelnému přenosu zpráv ze serveru o přijatých datech.

Nejzajímavější odlišností je to, že aplikace odesílající data pomocí WebSocketů může monitorovat využití výstupního bufferu [36] a tak zajistit, že aplikace odesílá data přesně v tom tempu, v jakém je přenosová linka schopna tato data vstřebat. **Je to přesně ta metoda, která zajistí, že rychlost odesílání dat aplikací je řízeno rychlostí odesílání dat síťovým adaptérem.** Pokud by odesílání dat bylo implementováno tímto způsobem, data by na běžných linkách a s využitím běžného hardwaru byla odesílána optimální rychlostí. Tímto způsobem by patrně mohlo vzniknout standardní řešení, které bude mít standardní projevy. V případě problému popisovaného výše je ale potřeba monitorovat zpracování dat na modemu, protože na měřených datech je vidět, že problém vznikl po odeslání takového množství dat z počítače, které

modem nebyl sto zpracovat (zde předpokládám, že použitý software, který způsobil zahlcení linky, je naprogramován tak, aby k tomu v běžných konfiguracích nedocházelo a že tedy monitoruje využití výstupního bufferu nebo používá k dosažení obdobného výsledku jiné prostředky).

Historickou odlišností také bylo to, že komunikace přes WebSockety, na rozdíl od komunikace s využitím http požadavků, nemusela otevřít pro každý datový přenos nové spojení. Tento rozdíl je již nyní zastřen tím, že soudobé prohlížeče odesílají http požadavky přes jedno spojení a v případě potřeby jej opětovně naváží.

Použití WebSocketů zvyšuje složitost řešení a vyžaduje širší znalostní základnu pro případné aplikační změny. Proto jsem se rozhodl, že se pokusím implementovat odesílání dat pomocí http požadavků, a až v případě neúspěchu přistoupím k využití WebSocketů. V případě bakalářské práce je navíc dle mého názoru vhodné experimentovat s různými přístupy a nikoliv nutně implementovat standardní řešení. Výsledná implementovaná aplikace je založena pouze na využití http požadavků a data jsou přenášena formou binárních souborů [35], [37].

3.2.3 Implementované řešení

Terminologické poznámky: V textu níže používám pojem datový rámec ve smyslu aplikačního datového rámce, tedy jako velikost najednou přenášených aplikačních dat bez ohledu na to, jakým způsobem jsou data po lince skutečně přenášena. Užitečnou zátěží resp. rychlostí přenosu užitečné zátěže pak mám na mysli to, že neuvažuji přenos řídicích a podpurných dat, ale pouze obsah přenášených souborů.

3.2.3.1 Proměnlivé časování

Popisovat zde celou cestu k výslednému řešení nemá valnou hodnotu, z různých slepých uliček uvedu pouze tu nejzajímavější, jejíž prozkoumání se na výsledném řešení významným způsobem projevilo. Původně jsem předpokládal, že aplikace bude posílat datové rámce stejné velikosti v různých časových odstupech podle aktuálně detekované kapacity linky. Tato idea se ukázala jako neschůdná.

Hlavním problémem bylo to, že používané datové rámce musely být dostatečně malé, aby jejich odeslání na pomalé datové lince – s rychlostí 1 Mb/s – trvalo dostatečně krátkou dobu, aby nedošlo k zahlcení již jedním datovým rámcem a aby uživatel mohl sledovat postup odesílání dat v reálném čase, a přitom tak velké, aby fixní čas na zpracování jednoho datového rámce – cca 30 ms (tolik trvá zpracování datového rámce o nulové velikosti na serveru) – zbytečně nezpomaloval dobu přenosu. Například při použití linky s rychlostí 0,5 Mb/s vede požadavek doby odeslání dat za 0,25 vteřiny na velikost datového rámce cca 15000 bytů. To ovšem znamená, že při rychlosti 50 Mb/s bude každou vteřinu odesláno 400 datových rámců, jejichž celkový fixní čas zpracování na serveru je 12 vteřin.

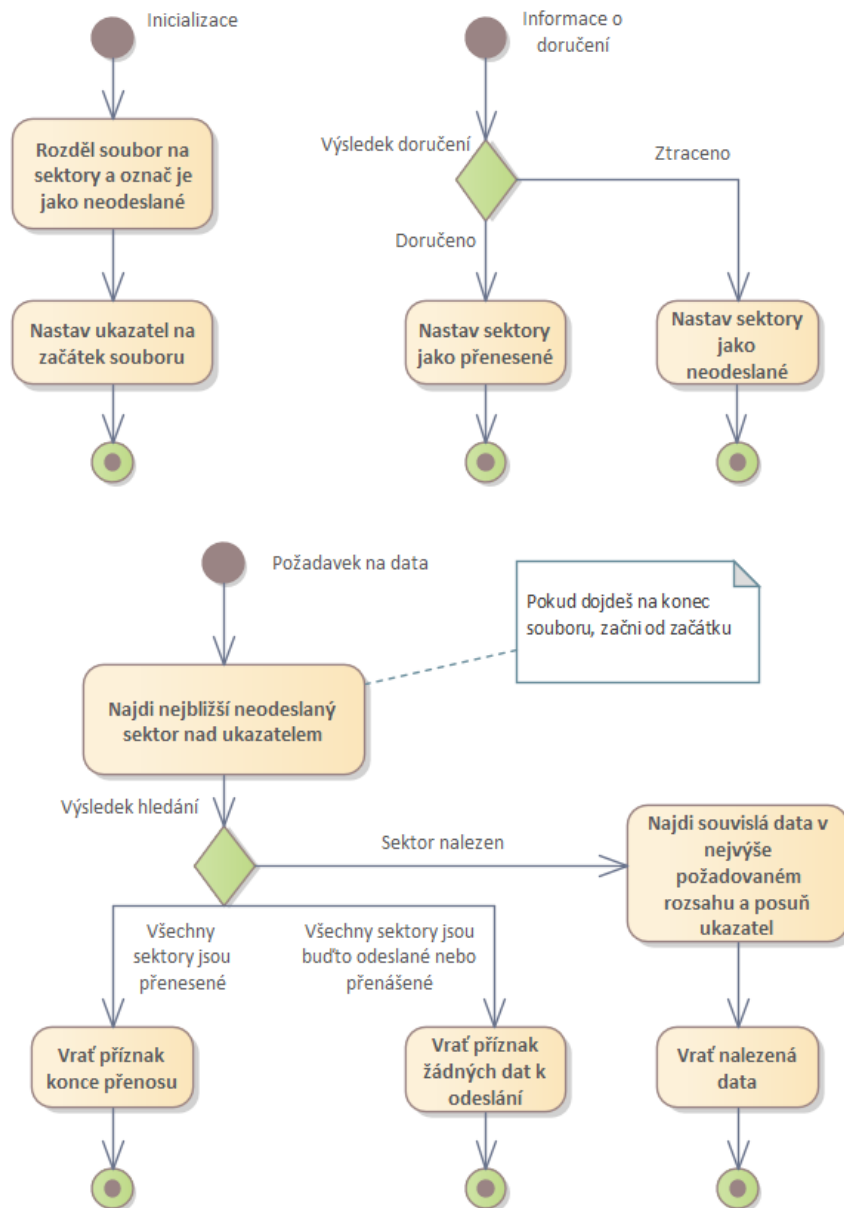
Pokud má aplikace být schopna odesílat data po lince s rychlostí 0,5 Mb/s a reagovat na její případné zahlcení a současně využít kapacitu linky s rychlostí 50 Mb/s, musí používat datové rámce proměnlivé velikosti. Je možné, že tento problém by elegantně vyřešilo použití přenosu dat založeného na využití WebSocketů, protože jednotlivé přenosy dat mají zjednodušený proces zpracování a například se při každém dílčím přenosu neposílají cookies, ale tuto variantu jsem neimplementoval.

3.2.3.2 Proměnlivá velikost datových rámců

V této části práce popisují výsledné řešení pro přenos dat.

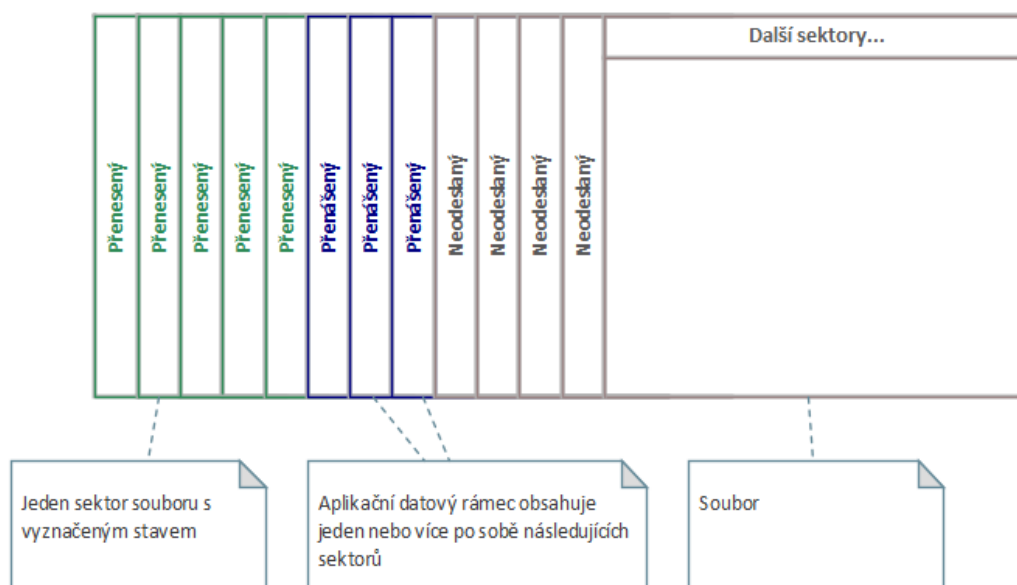
Algoritmus pro odesílání dat má dvě části – část řídicí data souboru k odeslání a část řídicí přenos dat z prohlížeče na server.

Část řídicí data souboru k odeslání na obrázku 3.7 pracuje následujícím způsobem.



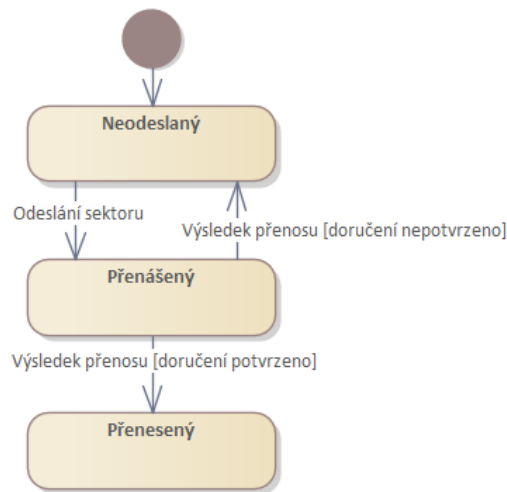
■ Obrázek 3.7 Řízení dat souboru

1. Obsah souboru rozdělí na sektory po 8192 bytech tak, jak je naznačeno na obrázku 3.8 a všechny sektory označí jako neodeslané. Jednotlivé sektory prochází stavy na obrázku 3.9. Ukazatel procházení souboru nastaví na začátek a čeká na požadavky na data k odeslání.
2. Požadavek na data k odeslání obsahuje nejvyšší počet sektorů, které mohou být najednou odeslány. Algoritmus najde nejbližší dosud neodeslaný sektor za ukazatelem procházení souboru a vrátí nejvýše tolik na sebe navazujících dosud neodeslaných sektorů, kolik může být najednou odesláno. Odeslané sektory označí jako přenášené. Pokud při hledání prvního dosud neodeslaného sektoru dorazí na konec souboru, pokračuje v hledání znovu od začátku. Pokud při hledání zjistí, že všechny sektory jsou již odeslané, vrátí zprávu, že přenos byl dokončen. Pokud při hledání zjistí, že všechny sektory jsou buďto odeslané nebo přenášené, vrátí zprávu, že aktuálně není co přenášet.
3. Jakmile získá informaci o doručení některého datového rámce, označí sektory z toho rámce jako přenesené nebo neodeslané podle toho, zda informace o doručení obsahovala potvrzení o doručení nebo informaci o nedoručení dat. Vzhledem k tomu, že přenos probíhá s využitím TCP/IP protokolu, není informace o nedoručení dat běžným operačním stavem.

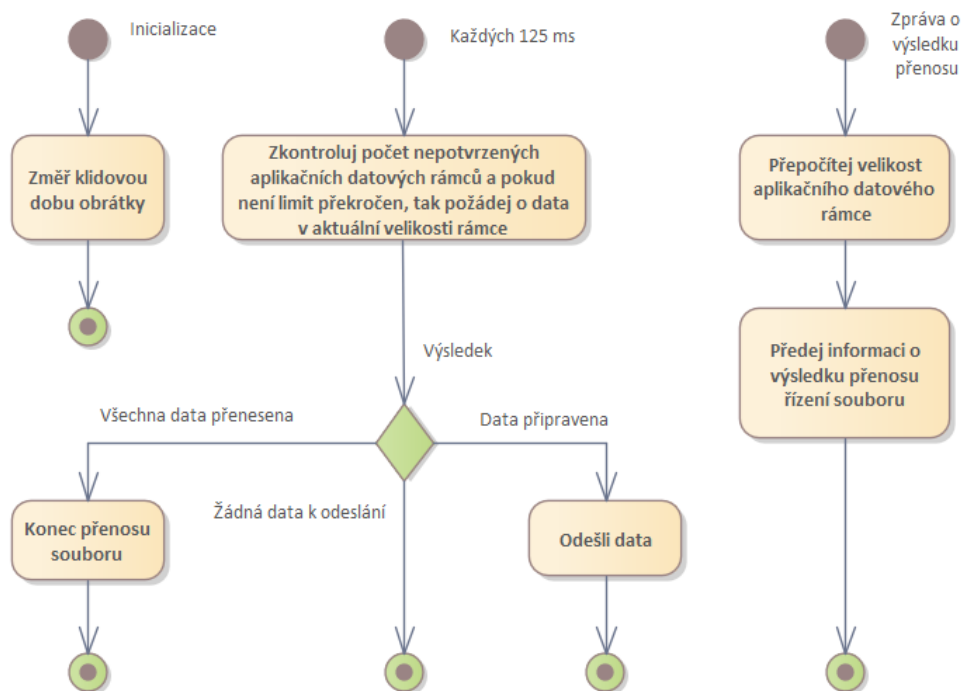


■ **Obrázek 3.8** Rozdělení souboru na sektory

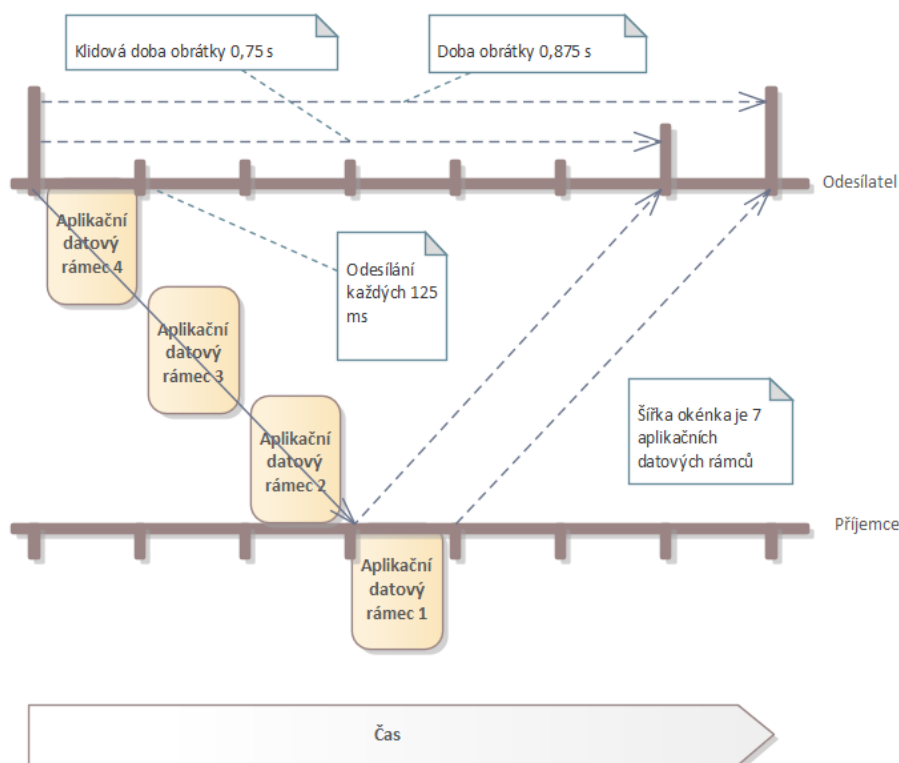
Část řídicí přenos dat zachycená na obrázku 3.10 spouští pokus o odeslání dat každých 125 ms a velikost odesílaných dat určuje tak, aby čas zpracování datového rámce určený jako aktuální doba obrátky nad dobu obrátky změřenou před započítáním přenosu odpovídal **dvojnásobku** doby mezi odesláním datových rámců. Současně je omezen počet nepotvrzených rámců jako počet rámců odpovídající klidové době obrátky (době obrátky na nezatížené lince) zvýšený o jeden rámeček. To znamená, že algoritmus je navržen tak, aby odesílal datové rámce 4 – 8 krát za vteřinu. Po spuštění algoritmu se ustálí dynamická rovnováha a přenos dat probíhá tak, jak je schematicky načrtnuto na obrázku 3.11 (oproti skutečným poměrům při přenosu dat je situace na obrázku zjednodušená).



■ **Obrázek 3.9** Stavy sektoru souboru



■ **Obrázek 3.10** Řízení přenosu dat

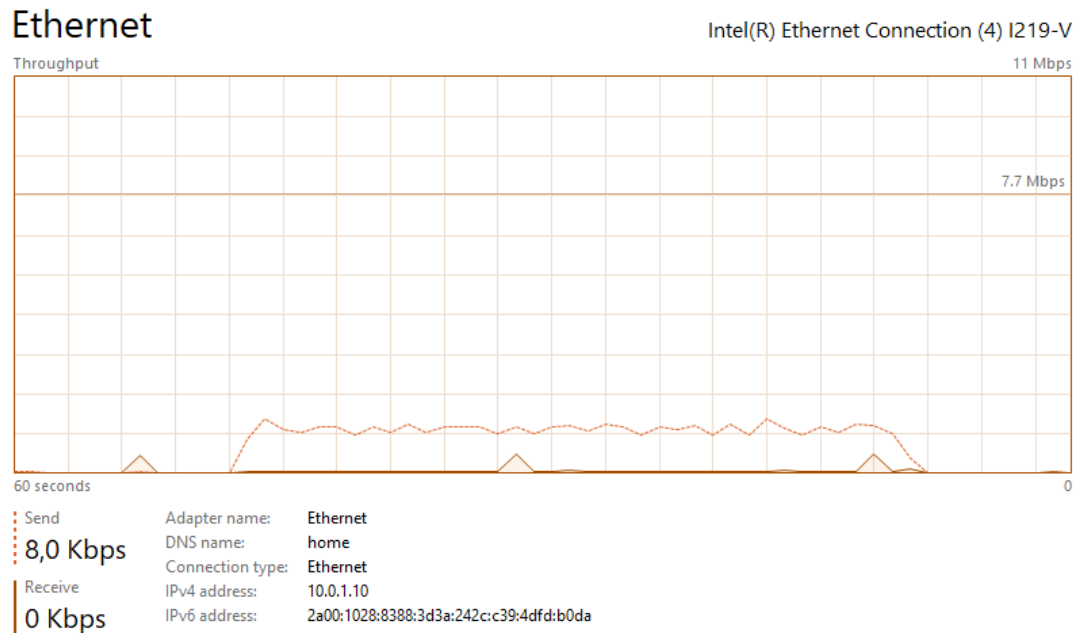


■ **Obrázek 3.11** Zjednodušený náčrt toku dat

Výše popsáný princip je implementován tímto způsobem:

1. Kód zjistí klidovou dobu obrátky před započítáním přenosu dat. Pokud doba obrátky při přenosu klesne pod klidovou dobu obrátky, uloženou dobu obrátky aktualizuje.
2. Výchozí velikost datového rámce je nastavena na 1 sektor.
3. Kód v cyklu 125 ms kontroluje, že není překročen počet nepotvrzených rámců a požádá o data k odeslání podle aktuálního nastavení velikosti datového rámce a data odešle. Pokud není co odeslat, odeslání v tomto cyklu vynechá a pokračuje až dalším cyklem. Pokud je doručení všech sektorů potvrzeno, algoritmus končí.
4. Pokud je přijata informace o ztrátě datového rámce, je aktuální velikost datového rámce snížena na jednu polovinu.
5. Pokud je přijata informace o doručení datového rámce, kód spočítá optimální velikost datového rámce z doby odeslání potvrzeného rámce, ta ale nesmí přesáhnout devítinásobek velikosti původního datového rámce. Tuto optimální velikost použije k úpravě aktuální velikosti datového rámce tak, že nová velikost datového rámce je určena jako součet 90 % aktuální velikosti datového rámce a 10 % optimální velikosti datového rámce. Pokud je změřená doba obrátky po odečtení klidové doby obrátky nula nebo záporná, optimální velikost datového rámce nelze spočítat a aktuální velikost datového rámce je zvětšena na dvojnásobek.
6. Nejvyšší povolená velikost datového rámce je $1024 * 8192$ bytů. Tím je omezena maximální rychlost přenosu na 268 – 537 MB/s podle aktuální frekvence odesílání datových rámců.

Nyní můžeme zopakovat experiment z odstavce 3.2.1.1. Výsledek je na obrázcích 3.12 a 3.13. Za 38 vteřin bylo odesláno 5,23 MB dat ve 244 datových rámcích. Průměrná rychlost z přenosu užitečné datové zátěže byla 1.1 Mb/s a průměrná velikost datového rámce byla 21400 bytů.



■ **Obrázek 3.12** Aplikační odeslání 5 MB přes modem O2

```

Reply from 2001:718:2:2908::212: time=9ms
Reply from 2001:718:2:2908::212: time=8ms
Reply from 2001:718:2:2908::212: time=116ms
Reply from 2001:718:2:2908::212: time=177ms
Reply from 2001:718:2:2908::212: time=197ms
Reply from 2001:718:2:2908::212: time=220ms
Reply from 2001:718:2:2908::212: time=82ms
Reply from 2001:718:2:2908::212: time=110ms
Reply from 2001:718:2:2908::212: time=247ms
Reply from 2001:718:2:2908::212: time=159ms
Reply from 2001:718:2:2908::212: time=196ms
Reply from 2001:718:2:2908::212: time=232ms
Reply from 2001:718:2:2908::212: time=267ms
Reply from 2001:718:2:2908::212: time=244ms
Reply from 2001:718:2:2908::212: time=166ms
Reply from 2001:718:2:2908::212: time=200ms
Reply from 2001:718:2:2908::212: time=170ms
Reply from 2001:718:2:2908::212: time=179ms
Reply from 2001:718:2:2908::212: time=123ms
Reply from 2001:718:2:2908::212: time=154ms
Reply from 2001:718:2:2908::212: time=212ms
Reply from 2001:718:2:2908::212: time=254ms
Reply from 2001:718:2:2908::212: time=294ms
Reply from 2001:718:2:2908::212: time=158ms
Reply from 2001:718:2:2908::212: time=193ms
Reply from 2001:718:2:2908::212: time=108ms

```

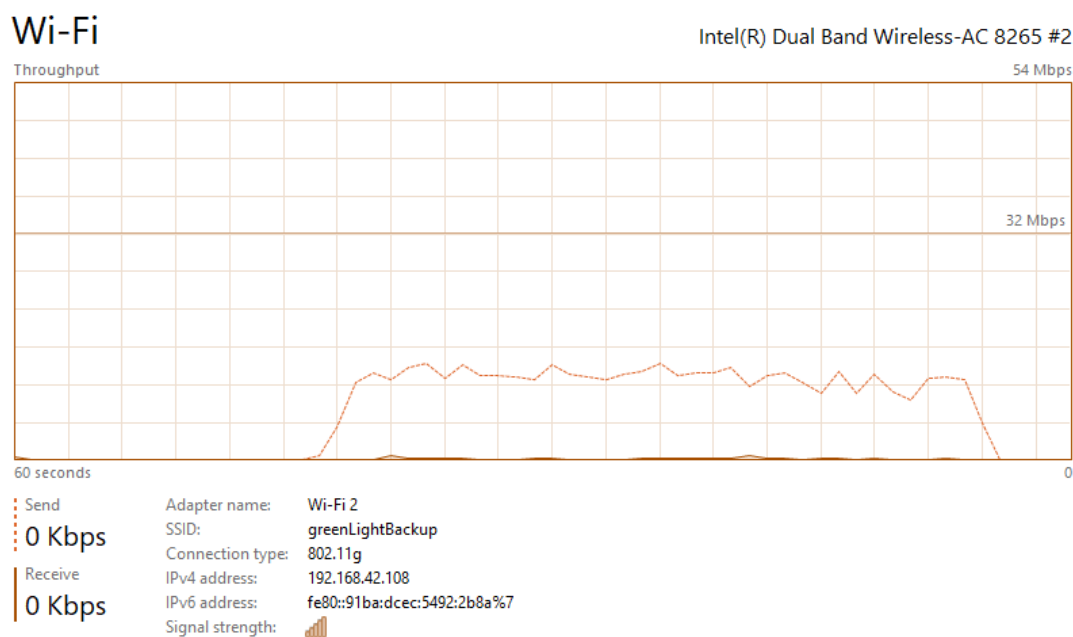
```

Reply from 2001:718:2:2908::212: time=141ms
Reply from 2001:718:2:2908::212: time=228ms
Reply from 2001:718:2:2908::212: time=255ms
Reply from 2001:718:2:2908::212: time=175ms
Reply from 2001:718:2:2908::212: time=90ms
Reply from 2001:718:2:2908::212: time=233ms
Reply from 2001:718:2:2908::212: time=233ms
Reply from 2001:718:2:2908::212: time=95ms
Reply from 2001:718:2:2908::212: time=131ms
Reply from 2001:718:2:2908::212: time=104ms
Reply from 2001:718:2:2908::212: time=195ms
Reply from 2001:718:2:2908::212: time=210ms
Reply from 2001:718:2:2908::212: time=266ms
Reply from 2001:718:2:2908::212: time=9ms
Reply from 2001:718:2:2908::212: time=8ms

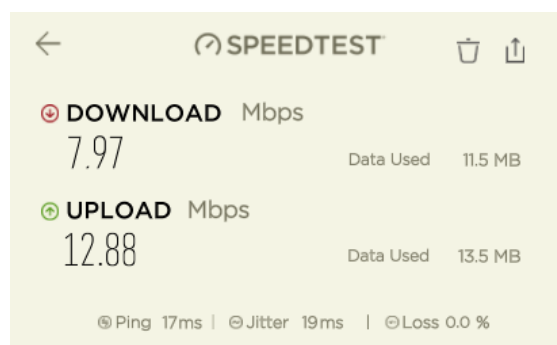
```

■ **Obrázek 3.13** Ping na `www.fit.cvut.cz` během aplikačního odesílání dat

Průběh aplikačního přenosu dat pomocí mobilního připojení T-Mobile je zachycen na obrázku 3.14, změřená rychlost pak na obrázku 3.15. Za 38 vteřin bylo odesláno 51,18 MB dat ve 231 datových rámcích. Průměrná rychlost z přenosu užitečné datové zátěže byla 10,7 Mb/s a průměrná velikost datového rámce byla 222.000 bytů.

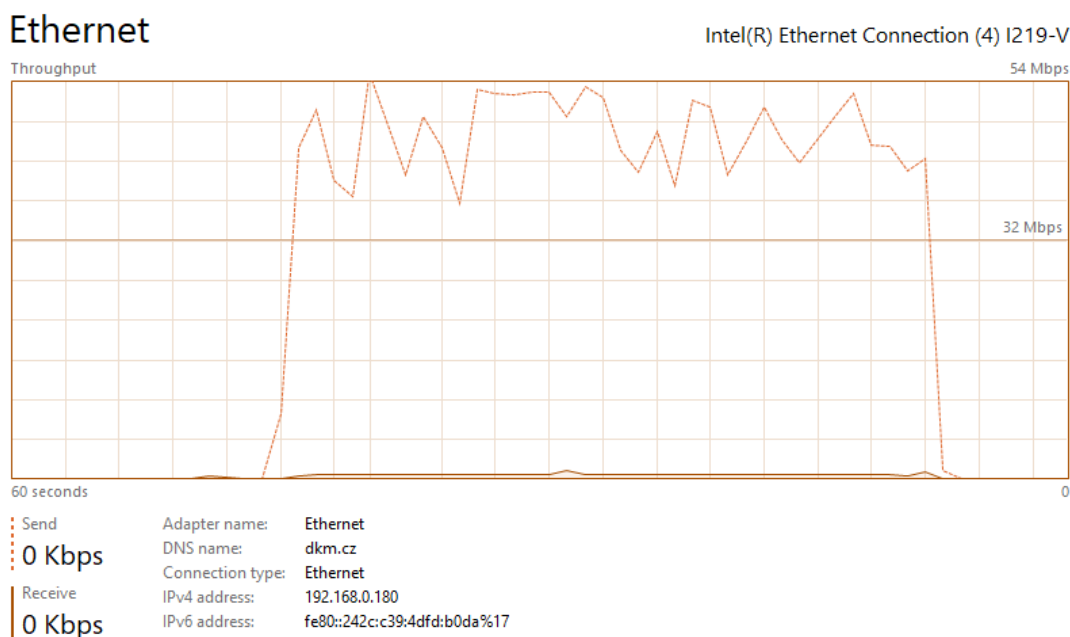


■ **Obrázek 3.14** Aplikační odesílání dat 38 vteřin přes mobilní připojení T-Mobile

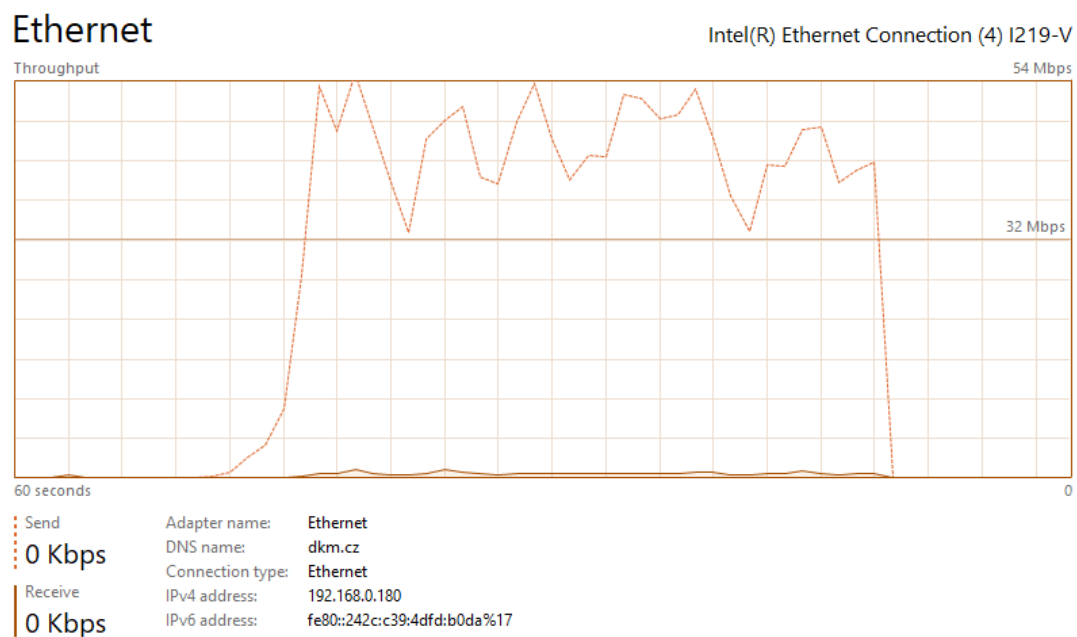


■ **Obrázek 3.15** Změřená rychlost mobilního připojení T-Mobile

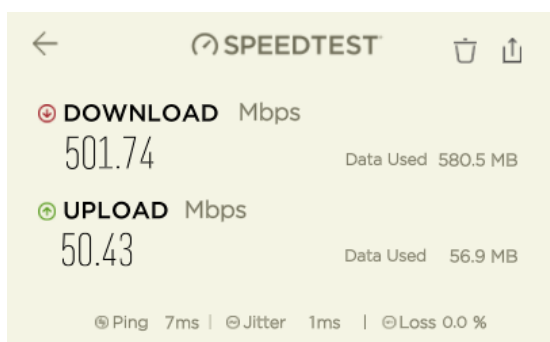
Průběh aplikačního přenosu dat pomocí kabelového připojení operátora Vodafone je zachycen na obrázku 3.16 a se simulovaným zpožděním 1 vteřina na obrázku 3.17, změřená rychlost pak na obrázku 3.18. Za 38 vteřin bylo odesláno 204 MB dat ve 243 datových rámcích. Průměrná rychlost měřená z přenosu užitečné datové zátěže byla 42,7 Mb/s a průměrná velikost datového rámce byla 840.000 bytů. Při simulovaném zpoždění 1 vteřina bylo za 38 vteřin odesláno 178 MB dat ve 263 datových rámcích. Průměrná rychlost měřená z přenosu užitečné datové zátěže byla 36,1 Mb/s a průměrná velikost datového rámce byla 677.000 bytů.



■ **Obrázek 3.16** Aplikační odeslání dat 38 vteřin přes kabelové připojení Vodafone



■ **Obrázek 3.17** Aplikační odesílání dat 38 vteřin přes kabelové připojení Vodafone se simulovaným zpožděním 1 vteřina



■ **Obrázek 3.18** Změřená rychlost kabelového připojení Vodafone

Souhrnné výsledky jsou uvedeny v tabulce 3.1 včetně měření přes linku ADSL, které není výše popsáno.

■ **Tabulka 3.1** Přehled aplikačního přenosu dat za prvních 38 vteřin přenosu

Připojení	Rychlost linky [Mb/s]	Dosažená rychlost užitečné zátěže [Mb/s]	Průměrná velikost datového rámce [byte]	Počet rámců
ADSL	0,60	0,51	13,400	183
VDSL	1,2	1,1	21,400	244
Mobilní	12,9	10,7	222.000	231
Kabelové	50,4	42,7	840.000	243
Kabelové + zpoždění 1 s	50,4	36,1	667.000	263

Chování odesílání dat na větší vzdálenosti bylo ověřeno pomocí aplikace Hotspot Shield Premium [38], kdy data mezi prohlížečem v Praze a webovým serverem v Hluboké nad Vltavou byla směrována přes světové metropole uvedené v tabulce 3.2. Přenos dat trval vždy 38 vteřin. Jednotlivé sloupce mají následující význam:

- Metropole: Místo, přes které data tekla zjištěné podle IP adresy přípojného bodu a služby GeoIP.com [39].
- Ping: Doba odezvy na ping na aplikační server. Nejedná se o dobu odezvy na http požadavek.
- Rychlost spojení: Rychlost změřená odesláním souboru na webový server pomocí scp, nejedná se tedy o rychlost linky, ale o rychlost změřenou pro užitečnou zátěž. Rychlost byla shora omezena rychlostí připojení k Internetu, která byla 50 Mb/s.
- Aplikační rychlost: Rychlost změřená při aplikačním odesílání dat pro užitečnou zátěž.

■ **Tabulka 3.2** Ověření přenosu na dálku

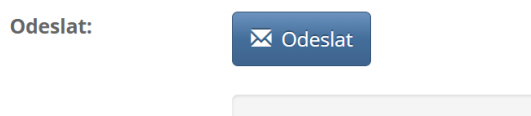
Metropole	Ping	Rychlost spojení [Mb/s]	Aplikační rychlost [Mb/s]
Přímé spojení	11	47	43
Vídeň	28	46	39
Hamburg	40	46	38
Tel Aviv	43	20	20
Zurich	47	44	37
Jerevan	49	46	43
Santiago	253	23	13
Nassau	259	23	23
Ho Chi Minh	345	16	16
Los Angeles	349	17	15
Kathmandu	462	11	12
Soul	548	14	12
Auckland	573	11	12

3.2.3.3 Odolnost vůči změnám parametrů linky

Použitý algoritmus není odolný pro případ významného zvýšení klidové doby obrátky během přenosu jednoho souboru. Pokud, například vlivem rekonfigurace sítě, dojde k prodloužení doby odezvy, které nesouvisí s tokem dat, algoritmus tuto změnu vyhodnotí jako přetížení linky a rychlost přenosu sníží. Problém je možné velmi snadno vyřešit pravidelným přerušováním přenosu

dat a recalibrací klidové doby obrátky. Toto řešení ale není implementováno, protože je to dle mého názoru zbytečné. V běžné praxi bude k takovým případům docházet jen zcela výjimečně, a pokud k tomu přeci jen dojde a zpomalení bude tak významné, že to bude uživatele obtěžovat, přenos přeruší a pustí jej znovu, čím dojde k nastavení aktualizovaného parametru pro dobu obrátky.

Před započítáním přenosu dat je zobrazeno tlačítko na obrázku 3.19. Jakmile na něj uživatel klikne, změní se tak, jak se znázorněno na obrázku 3.20. Aplikace při přenosu souborů indikuje postup odesílání dat. Pokud uživatel klikne na tlačítko „Stop“, zobrazení se přepne zpět do původní podoby. Po opětovném kliknutí na tlačítko odeslání formuláře aplikace provede nové změření klidové doby obrátky a v přenosu dat pokračuje tam, kde byl přenos předtím přerušen – již přenesená data nejsou znovu přenášena. Obdobně je možné postupovat při přerušení spojení nebo změně způsobu připojení k Internetu.



■ **Obrázek 3.19** Spuštění přenosu



■ **Obrázek 3.20** Zastavení přenosu

3.2.3.4 Minimální rychlost přenosu

Přenos souborů tímto způsobem byl ověřen ve více různých konfiguracích a k okamžiku dokončení práce nejsou známy případy, kdy by implementované řešení s použitím běžně dostupných linek selhalo. Algoritmus tak, jak je implementován, nepracuje dobře pro linky s rychlostí přenosu nižší než 0,075 MB/s (různé rychlosti jsem simuloval pomocí routeru Cisco RV260W, který umožňuje odchozí rychlost omezit). V takovém případě by bylo vhodné velikost datových rámců snížit pod stávající minimální velikost 8192 bytů. Taková linka ale stejně není k přenosu objemů dat používaných touto aplikací prakticky použitelná.

3.2.3.5 Použitý hardware

Aplikaci jsem vyvíjel a testoval s využitím následujících zařízení pro připojení k Internetu:

- Notebook Lenovo T480 s OS Microsoft Windows 10 Professional (interní síťová karta Intel(R) Ethernet Connection (4) I219-V a USB karta I-TEC USB 3.0 Metal Gigabit Ethernet)
- Routeru Cisco RV260W, Cisco RV340, Zyxel VMG3312-T20A, Vodafone Station, O2 Smart Box a TP-LINK TD-W8961NB

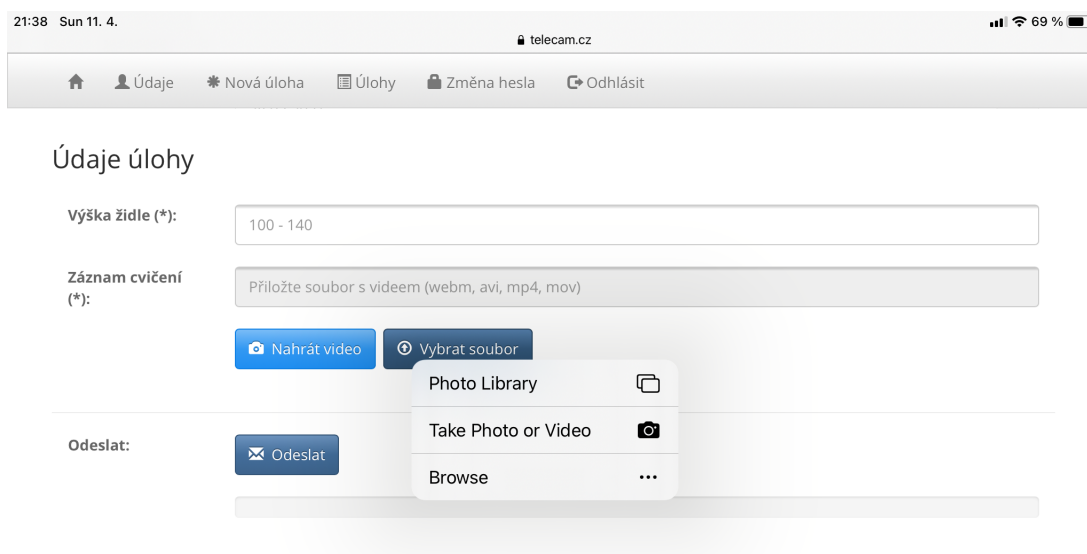
- Mobilní telefon Apple SE 2016 a tablet Apple iPad 2018
- Přípojný Wi-Fi body Ubiquiti UniFi UAP-AC-PRO a Cisco WAP361
- Více různých USB modemů Huawei E3372H-320 a mobilní router Teltonika LTE Router RUT240

3.3 Pořízení videonahrávky

Zadavatel požadoval, aby uživatel pořídil v internetovém prohlížeči videonahrávku, tuto videonahrávku si mohl přehrát a následně odeslat na server. Zadavatel dále požadoval, aby rozlišení záznamu bylo minimálně 1280 x 1024 bodů při snímkovací frekvenci minimálně 25 snímků za vteřinu s tím, že ideální je používat fixní snímkovací frekvenci. Preferovanými kodeky pro kódování videa jsou (v tomto pořadí) h.264, h.265, vp9 a vp8.

3.3.1 Technické možnosti

Možností, jak pořídít videonahrávku v prohlížeči, je celá řada. Podobně, jako v případě odesílání dat na server, jsem hledal řešení, které je co nejjednodušší, nevyžaduje použití dalších komponent a představuje co nejnižší riziko implementačních komplikací.



■ **Obrázek 3.21** Pořízení videonahrávky pomocí HTML Media Capture

3.3.1.1 HTML Media Capture

Základní možností pořízení videonahrávky je použití HTML Media Capture [40], [41]. Demonstrační implementace je na adrese <https://addpipe.com/html-media-capture-demo/> [42]. Na webové stránce je tlačítko „Vybrat soubor“ a při kliknutí je otevřena aplikace zařízení, na kterém jsou stránky zobrazeny, pro pořízení videonahrávky. Tato možnost jsem použil také ve výsledné webové aplikaci, jak je zachyceno na obrázku 3.21.

Tento způsob má dva základní nedostatky. Aplikace nemá kontrolu nad formátem pořízeného záznamu a metoda je podle [43] spolehlivě dostupná pouze na mobilních prohlížečích. Pro splnění požadavků zadavatele není toto řešení postačující.

3.3.1.2 WebRTC

WebRTC (Web Real-Time Communication) je technologie, která umožňuje webovým aplikacím a webům zaznamenávat a volitelně streamovat zvuková a/nebo obrazová média a také vyměňovat libovolná data mezi prohlížeči bez nutnosti použití prostředníka. Sada standardů, které zahrnují WebRTC, umožňuje sdílet data a provádět telekonference peer-to-peer, aniž by bylo nutné, aby uživatel instaloval doplňky nebo jiný software třetích stran [44].

Možnosti, které WebRTC nabízí, jsou popsány v přehledovém příspěvku [45]. Ve spojení s „Media Capture and Streams API (Media Stream)“ (je to totéž jako „Media Streams API“ a jako „MediaStream API“) je API související s WebRTC, které poskytuje podporu pro streamování zvukových a obrazových dat. Poskytuje rozhraní a metody pro práci s datovými proudy a jejich stopami, omezení spojená s datovými formáty, úspěšná a chybná zpětná volání při asynchronním použití dat a událostmi, které se aktivují během procesu [46].

WebRTC umožňuje výměnu médií v reálném čase peer-to-peer mezi dvěma zařízeními. Spojení je navázáno prostřednictvím procesu zjišťování a vyjednávání, který se nazývá signalizace [47].

Tyto nepochybně velmi silné stránky jsou patrně vyváženy kompromisy ve kvalitě videa, které je pořizováno tak, aby odpovídalo dostupné šířce přenosového pásma. Podle [43] a [48] dochází při použití WebRTC k pořízení videozáznamu k problémům s kvalitou záznamu. Nicméně předpokládám, že při správném použití k účelu, pro které WebRTC **není určeno**, totiž k záznamu v rámci prohlížeče, je možné tento problém účinně adresovat.

Cesta k použití WebRTC k pořízení videozáznamu v prohlížeči nepochybně existuje. Článek [18] uvádí dvě možnosti, a to využití knihoven VideoJS [49] a VideoJS Record [50] nebo knihovny RecordRTC [17]. Popis knihovny RecordRTC je doplněn demonstračním použitím [51], které má přibližně ty vlastnosti, které jsou od cílového řešení požadovány.

V přehledovém článku [43] z roku 2017 autoři komerčního řešení pro pořizování videonahrávek na webových stránkách a jejich přenos na webový server vysvětlují, proč se rozhodli svoje řešení založit právě na WebRTC a nepoužít „MediaStream Recording API“ [52], [53], na kterém je aplikace z této bakalářské práce založena. Jejich hlavní obavou byla nedostatečná budoucí podpora v prohlížečích.

WebRTC ve **spojení s vhodnou knihovnou pro pořízení nahrávky v prohlížeči** je cestou, jak požadavky zadavatele této práce splnit. Zbývá tedy rozhodnout, kterou z možných knihoven bude nejlepší použít, případně zda je schůdné jít o úroveň níže a funkčnost knihovny nahradit vlastním kódem. Tohoto rozhodnutí jsem byl ušetřen, respektive bylo triviální, protože MediaStream Recording API jako součást soudobých prohlížečů je právě takovou knihovnou, kterou je ve spojení s WebRTC vhodné použít [54].

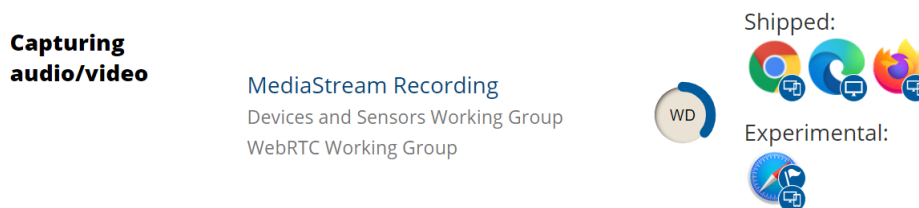
3.3.1.3 MediaStream Recording API

„MediaStream Recording API“ [52], [53], [54], [55], [56], někdy označované jako Media Recording API nebo MediaRecorder API, je podle [52] úzce spojeno s Media Capture and Streams API a WebRTC API. MediaStream Recording API umožňuje zachytit data generovaná objektem MediaStream nebo HTMLMediaElement pro analýzu, zpracování nebo uložení na disk. Podle [52] je také překvapivě snadné s ním pracovat.

Demonstrační implementace pořízení videonahrávky připravená autory WebRTC dostupná na stránce <https://webrtc.github.io/samples/src/content/getusermedia/record/> [57], [58] je implementována právě s využitím MediaStream Recording API. Velmi pěkné demo je k dispozici také na stránce <https://quickblox.github.io/javascript-media-recorder/sample/> [59]. Demonstrační implementace na webu <https://addpipe.com/media-recorder-api-demo/> [60] byla inspirací pro dílčí části kódu použitého k pořízení záznamu z kamery v této bakalářské práci.

Podle vyjádření konsorcia W3C definujícího internetové standardy ze září 2020 [61] je specifikace MediaStream Recording API na vývojovém stupni WD (viz obrázek 3.22 a tabulka

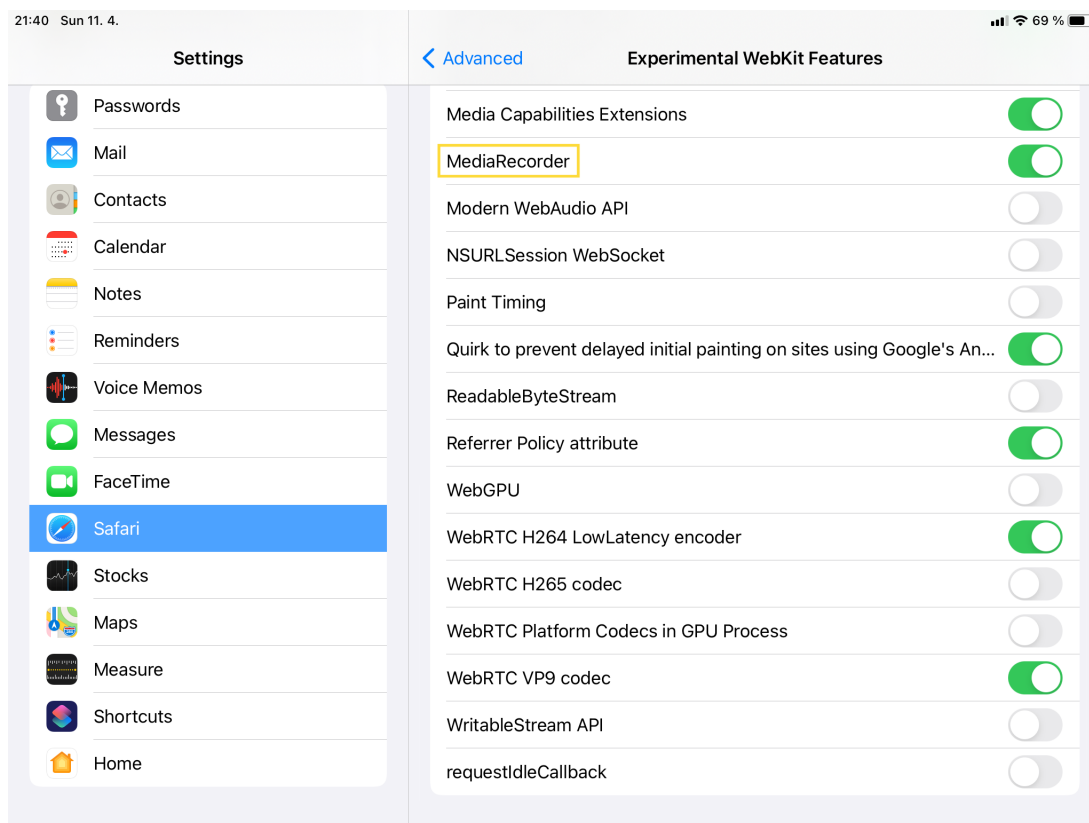
3.3) a dodané pro prohlížeče Chrome a Firefox. Na Safari je k dispozici jako **experimentální vlastnost** [63], kterou je potřeba zapnout tak, jak je ukázáno na obrázku 3.23.



■ **Obrázek 3.22** Vývojový stupeň specifikace MediaStream Recording API, převzato z [61]

■ **Tabulka 3.3** Vývojové stupně dle W3C podle [62]

Stupeň	Význam	Překlad
WD	Working Draft	Pracovní verze
CR	Candidate Recommendation	Kandidující doporučení
PR	Proposed Recommendation	Navržené doporučení
REC	W3C Recommendation	Doporučení W3C



■ **Obrázek 3.23** Zapnutí MediaRecording API v Safari na iPadu 2018

Riziko budoucí dostupnosti a podpory API formulované v [43] je dle mého názoru vyváženo jednoduchostí implementace, resp. tím, že k implementaci není potřeba použít nějakou knihovnu,

kteřá poskytuje podobnou míru abstrakce jako právě toto API. Autoři [43] problém řešili vývojem vlastního řešení založeného na WebRTC. Toto řešení je komerčně dostupné za poplatek [64] a nepochybně existují i další podobná komerční řešení; některá řešení dostupná bez poplatku jsem uvedl výše. Dle mého názoru je optimální aplikaci nyní zprovoznit s využitím MediaStream Recording API, protože je to ze všech možností patrně ta nejjednodušší. Pokud by se v budoucnosti ukázalo, že aplikace potřebuje jiné řešení nebo že řešení založené na MediaStream API není dostatečně podporované v prohlížečích, bude potřeba implementaci změnit. V době této změny bude mít provozovatel řešení již zkušenosti z použití MediaStream Recording API a bude při rozhodování o budoucím řešení moci tyto zkušenosti využít.

3.3.2 Implementované řešení

Implementované řešení je založeno na použití MediaStream Recording API a proto využívá i související rozhraní Media Capture and Streams API a WebRTC API. Dílčí části aplikačního kódu byly inspirovány demonstrační implementací na stránce <https://addpipe.com/media-recorder-api-demo/> [60]. Při použití WebRTC API je podle [65] doporučeno použít knihovnu adapter.js [66], která interceptuje volání API prohlížeče a standardizuje je. Při implementaci aplikace v této bakalářské práci tato knihovna **nebyla** použita. Pokud by došlo k potížím s kompatibilitou některých prohlížečů, její použití může být velmi snadným řešením takové situace. Knihovna je k dispozici pod licencí typu BSD [67].

Na začátku nahrávání je s využitím výše uvedených API možné stanovit omezení na rozlišení videa nebo snímkovací frekvenci určením minimálních, optimálních a maximálních hodnot tak, jak je zřejmé z příkladu na obrázku 3.24.

```
const constraints = {
  video: {
    width: {
      min: 640,
      ideal: 1280,
      max: 1280
    },
    height: {
      min: 320,
      ideal: 1024,
      max: 1024
    },
    framerate: {
      min: 25,
      ideal: 25,
      max: 50
    }
  }
};
navigator.mediaDevices.getUserMedia(constraints);
```

■ **Obrázek 3.24** Požadované rozlišení videa

Prohlížeč tyto hodnoty v určitém rozsahu respektuje, v závislosti na dostupném rozlišení zařízení, okamžitém stavu zařízení a okamžitém stavu prohlížeče může použít odlišné hodnoty. Na stejném zařízení může nahrávání při různých pokusech a z uživatelského hlediska stejných podmínek vést k různým výsledkům. Aplikace tak, jak je nyní implementována, požaduje hodnoty z příkladu výše, skutečně použité rozlišení nahrávky neověřuje a spolu s předanými daty reportuje

doplňkové informace o použitém nastavení tak, jak je ukázáno na příkladu na obrázku 3.25. Tyto údaje jsou aplikaci poskytnuty výše uvedenými API.

```
"settings": {
  "videoSettings": {
    "facingMode": "user",
    "aspectRatio": 1.25,
    "colorTemperature": 4600,
    "frameRate": 30,
    "contrast": 50,
    "width": 1280,
    "height": 1024
  },
  "selectedMimeType": "video/webm;codecs=h264",
  "recordedMimeType": "video/x-matroska;codecs=avc1,opus"
},
```

■ **Obrázek 3.25** Reportování rozlišení videa

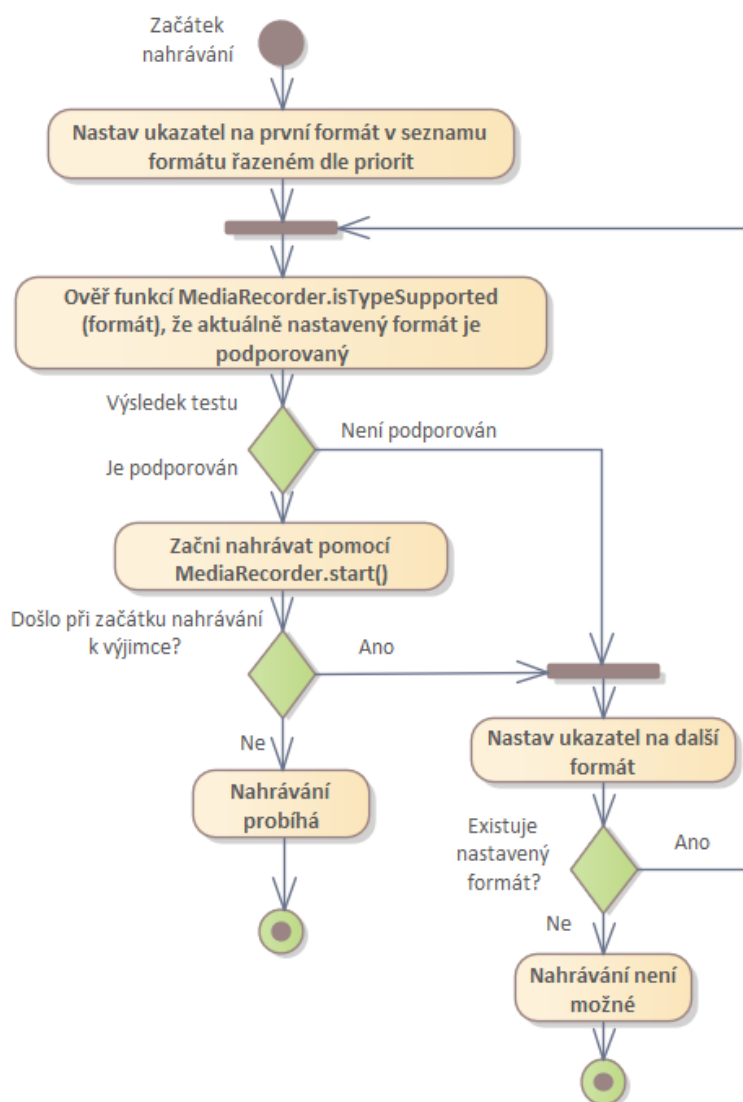
Údaj v poli `selectedMimeType` je formát, který byl při nahrávání zvolen postupem popsaným níže, v poli `recordedMimeType` je pak formát, který byl prohlížečem při nahrávání uveden.

Výběr formátu nahrávání aplikace, která je předmětem této práce, provádí ve dvou po sobě následujících krocích. Nejprve ověří, že prohlížeč resp. zařízení formát nahrávání podporují pomocí funkce na příkladu níže.

```
MediaRecorder.isTypeSupported('video/webm;codecs=h264')
```

Pokud je formát podporovaný, aplikace se pokusí začít nahrávat. Pokud je formát skutečně podporovaný, nahrávání začne, v opačném případě dojde k výjimce, která je aplikací zachycena, a aplikace prověří možnost použít další formát v pořadí podle priority. Celý proces výběru formátu je zachycen na obrázku 3.26. Nyní aplikace používá následující formáty s prioritou určenou tímto pořadím:

1. `video/webm;codecs=h264`
2. `video/webm;codecs=h265`
3. `video/webm;codecs=vp9`
4. `video/webm;codecs=vp8`
5. `video/mp4;codecs=avc1`
6. `video/webm`
7. `video/mp4`



■ **Obrázek 3.26** Proces výběru formátu videonahrávky

Pořízení nahrávky z uživatelského hlediska je popsáno v uživatelské příručce v příloze B.

3.3.2.1 Použitý hardware

Funkčnost jsem implementoval s využitím následujících kamer:

- Kamera notebooku Lenovo T480 (OS Microsoft Windows 10 Professional)
- Kamera Ausdom AW615
- Kamera Logitech Webcam C925e

- Kamera mobilního telefonu Apple SE 2016
- Kamera tabletu Apple iPad 2018
- Kamera tabletu Lenovo Yoga 3 (OS Android)
- Virtuální kamera ManyCam [68]
- Virtuální kamera Logitech capture [69]

Projektový postup

V této kapitole popisuji projektový postup od zadání tématu práce až po její dokončení.

4.1 Analýza požadavků

Na tématu bakalářské práce jsem se s vedoucím bakalářské práce dohodl 13. října 2020. S odborným konzultantem Ing. Janem Hejdou, Ph.D., jsme si od 19. října nastavili týdenní pravidelné schůzky, kdy jsem dostával informace o požadavcích na aplikaci, která je předmětem této práce, a měl příležitost diskutovat svoje otázky.

Zadavatel měl poměrně přesnou a vnitřně konzistentní představu o požadované funkčnosti aplikace, což proces analýzy požadavků velmi zjednodušilo a usnadnilo.

Pokud to bylo vhodné, jako například v případě návrhu logického a fyzického datového modelu nebo v případě modelu případů užití jednotlivých aktérů, jsem si před schůzkou připravil návrh řešení, abychom mohli diskutovat konkrétní detaily.

V polovině prosince roku 2020 jsem připravil návrh specifikace aplikace v příloze D a započal implementační práce.

4.2 Implementace řešení

Do konce roku 2020 jsem dosáhl základní funkčnosti všech klíčových prvků budoucí aplikace – na serveru existovala databáze s uloženým heslem uživatele aplikace, nad ní běžela aplikace s možností přihlášení uživatele, aplikace byla připojena k e-mailovému serveru, zdrojový kód zárodku aplikace byl uložen v GitLabu [70] a při sloučení kódu do hlavní větve probíhala kompilace s testováním včetně statické analýzy nástrojem SonarQube [71].

V průběhu ledna jsem našel způsob jak implementovat trvalé přihlášení uživatele s využitím Spring Security Frameworku [72], do aplikace jsem zakomponoval Bootstrap framework [73] s jQuery [74], notifikace pomocí komponenty toastr [75] a zvládl jsem použití cookies ve Spring Boot [2] aplikaci.

V závěru ledna, již po zkuškovém období, jsem během jednoho týdne implementoval lokalizační funkčnosti, na kterých je založeno vícejazyčné zobrazení uživatelského rozhraní.

První únorový týden jsem věnoval implementaci algoritmu pro odesílání velkých souborů na server. Bylo to pro mě poměrně náročné, protože algoritmus vyžadoval použití asynchronních funkcí v Javascriptu [76], [77], [78], s čímž jsem do té doby neměl žádné zkušenosti (jedná se o javascriptovou náhradu paralelního zpracování dat s ohledem na javascriptový model běhu kódu, kdy quasi-paralelně probíhá více různých procesů, které se navzájem ovlivňují a navzájem se nesmí blokovat).

Následující dva týdny jsem věnoval implementaci správy šablon úloh a komponent šablon úloh, včetně jejich zobrazení a odeslání a zobrazení seznamu vypracovaných úloh.

V závěru února jsem něco málo přes týden implementoval pořizování videonahrávky přes webovou kameru. Zde jsem s výhodou využil svoje nově nabyté znalosti asynchronních javascriptových funkcí, protože bez nich by implementace pořízení videonahrávky nebyla vůbec možná.

Tím byly všechny části práce, které vyžadovaly výzkum neznámých oblastí a získávání nových znalostí v základním rozsahu dokončeny a já jsem mohl svoje úsilí napřít k dokončení ostatních částí aplikace.

Během března po sobě v rychlém sledu následovala implementace správy uživatelů včetně editace osobních údajů a přihlašování se do role jiného uživatele, příprava rozhraní pro konfiguraci aplikace, implementace filtrování a stránkování pro zobrazení seznamu uživatelů, šablon, úloh a logů (stránku s logy jsem implementoval nad rámec požadavků a použil jsem ji pro testování stránkování a filtrování, protože v té době měly logy již cca 40.000 záznamů, kde každý záznam odpovídá volání REST funkce – tedy kliknutí na tlačítko na uživatelském rozhraní).

V závěru března jsem implementoval import vyhodnocení úlohy, aplikační logiku zajišťující řízení přístupu k objektům, odstranil nálezy ze statické analýzy kódu a kód aplikace pokryl automatizovanými testy.

V samém závěru března proběhlo uživatelské testování, viz odstavec 5.1.6.3.

V dubnu jsem odstranil zbylé chyby v kódu aplikace a připravil text práce.

Popis vypracovaného řešení

5.1 Popis řešení

Vypracovaná aplikace je založena na využití frameworku Spring [79] a jeho částí Spring Boot [2] a Spring Security [72]. Serverová část je napsaná v Javě 11, databáze je PostgreSQL [6]. Frontendová část využívá framework Bootstrap 3 [73] s jQuery [74]. V uživatelském rozhraní jsou použity komponenty toastr [75] pro notifikaci uživatele a Datepicker [80] pro zobrazení kalendáře. Grafická šablona vychází ze šablony Spacelab [81].

5.1.1 Popis datového modelu

Aplikační data jsou uložena v databázi. V této kapitole popisují všechny datové tabulky, které aplikace k ukládání dat používá.

5.1.1.1 Lokalizace

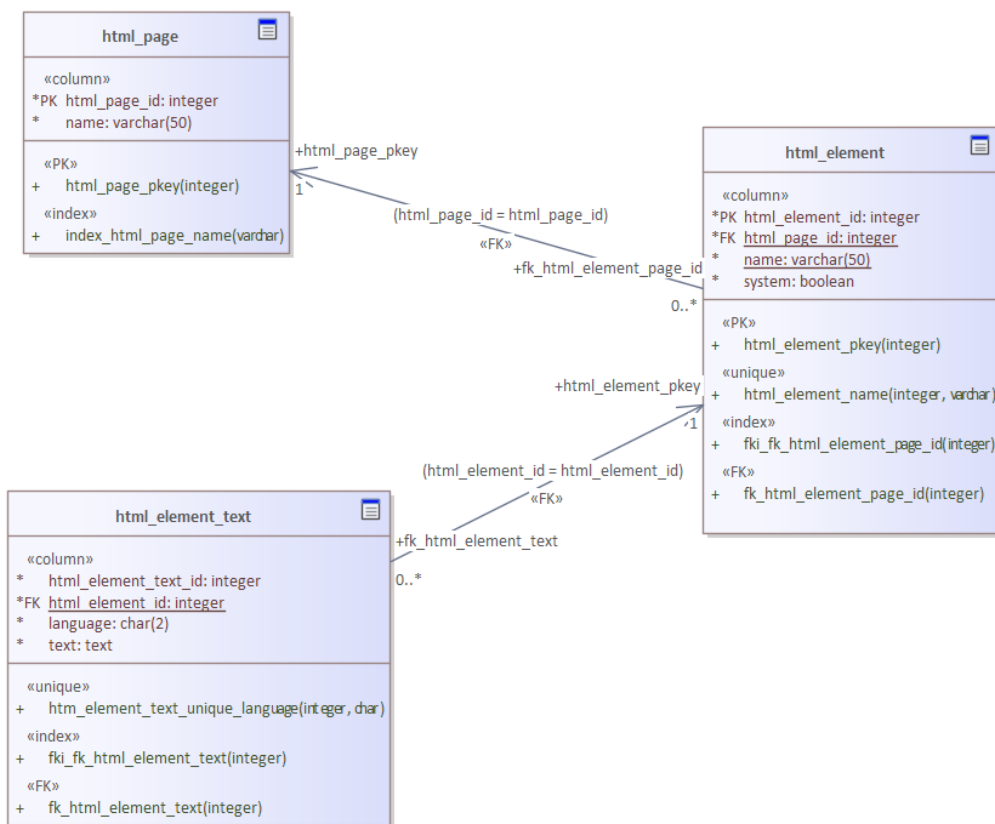
Lokalizační údaje jsou uloženy v tabulkách na obrázku 5.1. Jednotlivé datové tabulky jsou popsány v tabulce 5.1, jednotlivá pole datových tabulek pak v tabulkách 5.2, 5.3 a 5.4. Šablony html stránek a jejich částí jsou v aplikaci uloženy jako resources.

■ **Tabulka 5.1** Lokalizační data

Tabulka	Obsah
html_page	Html stránka nebo její část
html_element	Prvek html stránky
html_element_text	Text prvku v jednotlivých jazycích

■ **Tabulka 5.2** Pole tabulky html_page

Pole	Význam
html_page_id	Primární klíč
name	Identifikátor pro použití v kódu



■ **Obrázek 5.1** Lokalizační data

■ **Tabulka 5.3** Pole tabulky `html.element`

Pole	Význam
<code>html_element_id</code>	Primární klíč
<code>html_page_id</code>	Cizí klíč
<code>name</code>	Identifikátor pro použití v kódu
<code>system</code>	Indikátor systémového prvku, který musí být vždy lokalizován

■ **Tabulka 5.4** Pole tabulky `html.element_text`

Pole	Význam
<code>html_element_text_id</code>	Primární klíč
<code>html_element_id</code>	Cizí klíč
<code>language</code>	Identifikátor jazyka
<code>text</code>	Text prvku

5.1.1.2 Šablony

Data o šablonách jsou uložena v tabulkách na obrázku 5.2. Jednotlivé datové tabulky jsou popsány v tabulce 5.5, jednotlivá pole datových tabulek pak v tabulkách 5.6, 5.7, 5.8, 5.9 a 5.10.



■ Obrázek 5.2 Data o šablonách

■ **Tabulka 5.5** Data o šablonách

Tabulka	Obsah
task_template	Šablony úloh
task_template_name	Názvy šablon úloh v jednotlivých jazycích
task_template_component	Komponenty šablon úloh
task_template_component_name	Názvy komponent šablon úloh v jednotlivých jazycích
task_template_component_detail	Statické texty, html texty nebo meze číselných polí komponent šablon úloh

■ **Tabulka 5.6** Pole tabulky task_template

Pole	Význam
task_template_id	Primární klíč
date_created	Datum vytvoření šablony
active	Příznak aktivní šablony
custom_order	Pořadí šablony
client_data	Příznak použití šablony pro klientská data
identifier	Unikátní identifikátor

■ **Tabulka 5.7** Pole tabulky task_template_name

Pole	Význam
task_template_name_id	Primární klíč
task_template_id	Cizí klíč
language	Identifikátor jazyka
name	Název šablony

■ **Tabulka 5.8** Pole tabulky task_template_component

Pole	Význam
task_template_component_id	Primární klíč
task_template_id	Cizí klíč
custom_order	Pořadí komponenty šablony
component_type	Typ komponenty
mandatory	Příznak povinného pole
pre_filler	Příznak pole, které je předvyplněno
identifier	Unikátní identifikátor

■ **Tabulka 5.9** Pole tabulky task_template_component_name

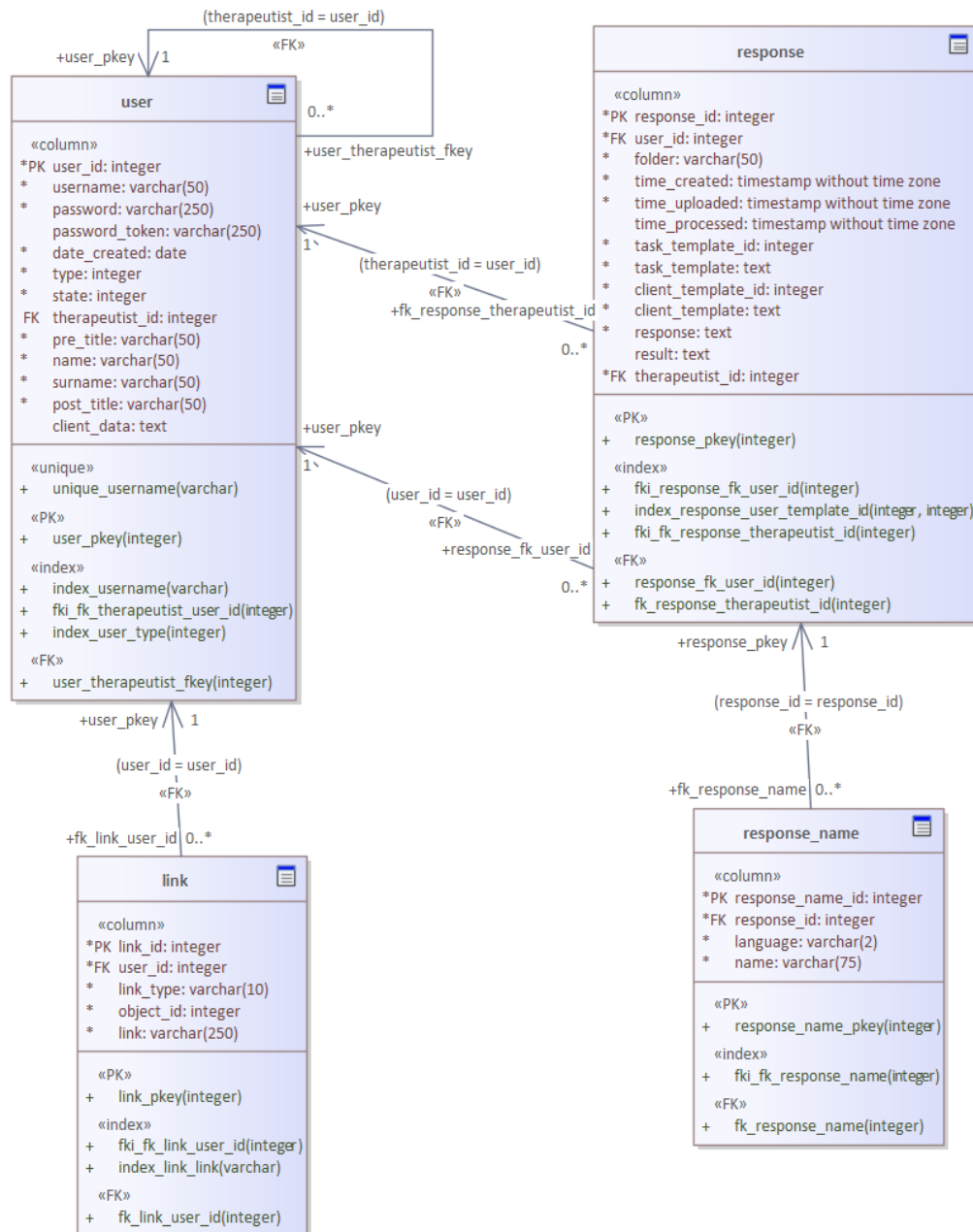
Pole	Význam
task_template_component_name_id	Primární klíč
task_template_component_id	Cizí klíč
language	Identifikátor jazyka
name	Název komponenty

■ **Tabulka 5.10** Pole tabulky task_template.component_detail

Pole	Význam
task_template.component_detail_id	Primární klíč
task_template.component_id	Cizí klíč
detail_type	Typ detailu – používané hodnoty jsou „staticText“ pro statický text nebo html text, a „valueFrom“ a „valueTo“ pro meze číselného rozsahu
language	Identifikátor jazyka; pro rozsah numerického pole je použit jazyk „XX“
detail	Údaje dle typu detailu

5.1.1.3 Úlohy

Údaje o úlohách jsou uloženy v tabulkách na obrázku 5.1. Jednotlivé datové tabulky jsou popsány v tabulce 5.11, jednotlivá pole datových tabulek pak v tabulkách 5.12, 5.13, 5.14 a 5.15.



■ Obrázek 5.3 Data o úlohách

■ **Tabulka 5.11** Data o úlohách

Tabulka	Obsah
user	Uživatelé aplikace
response	Vypracované úlohy
response_name	Název vypracované úlohy používaný pouze pro filtrování úloh v uživatelském rozhraní
link	Link na šablonu pro vypracování úlohy nebo na vyhodnocenou úlohu, aplikace udržuje nejvýše 100 linků jednoho uživatele

■ **Tabulka 5.12** Pole tabulky user

Pole	Význam
user_id	Primární klíč
username	E-mailová adresa uživatele
password	Řetězec odvozený od hesla uživatele
password_token	Řetězec používaný k nastavení prvního hesla
date_created	Datum vytvoření uživatele
type	Typ uživatele: 0 – správce, 1 – terapeut, 2 – klient
state	Stav uživatele: 0 – blokový, 1 – aktivní, 2 – neaktivní
therapist_id	v případě klienta primární klíč jeho terapeuta
pre_title	Tituly před jménem
name	Jméno
surname	Příjmení
post_title	Tituly za jménem
client_data	Json podoba dat o klientovi podle šablony používané pro osobní data klientů

■ **Tabulka 5.13** Pole tabulky response

Pole	Význam
response_id	Primární klíč
user_id	Cizí klíč identifikující klienta, který úlohy provedl
folder	Název složky, ve které jsou uloženy soubory úlohy
time_create	Čas vytvoření formuláře pro provedení úlohy klientem
time_uploaded	Čas dokončení úlohy klientem
time_processed	Čas načtení výsledků úlohy – pokud má pole hodnotu null, je úloha ve stavu „Čeká na zpracování“, jinak je ve stavu „Zpracovaná“
task_template_id	Klíč identifikující šablonu úlohy
task_template	Json podoba šablony úlohy
client_template_id	Klíč identifikující šablonu použitou pro osobní data klienta
client_template	Json podoba šablony úlohy použité pro osobní data klienta
response	Data o provedení úlohy v podobě, jaká přišla z prohlížeče
result	Data o vyhodnocení úlohy
therapist_id	Cizí klíč identifikující terapeuta klienta v době provedení úlohy

■ **Tabulka 5.14** Pole tabulky response_name

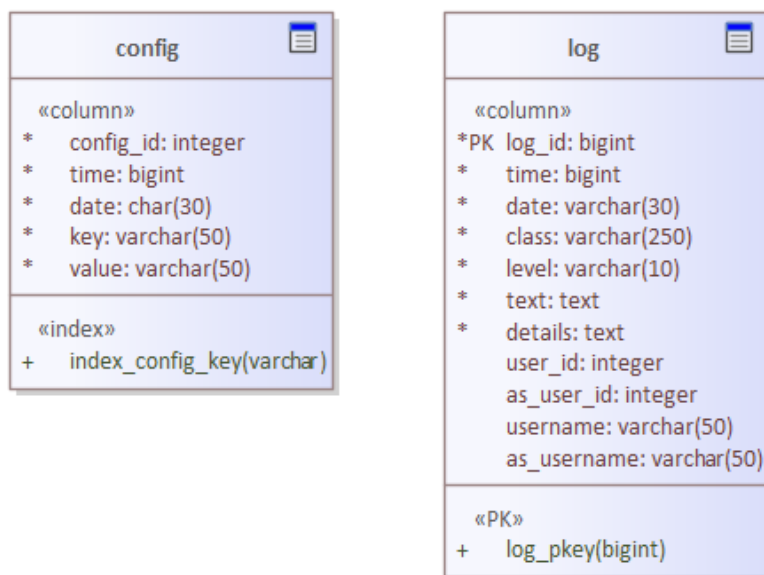
Pole	Význam
response_name_id	Primární klíč
response_id	Cizí klíč
language	Identifikátor jazyka
name	Název úlohy

■ **Tabulka 5.15** Pole tabulky link

Pole	Význam
link_id	Primární klíč
user_id	Cizí klíč
link_type	Určení typu linku – používané hodnoty jsou „taskLink“ pro link na úlohu k provedení a „viewLink“ pro link na vyhodnocenou úlohu
object_id	Klíč šablony nebo úlohy
link	Link

5.1.1.4 Pomocná data

Pomocná data jsou uložena v tabulkách na obrázku 5.4. Jednotlivé datové tabulky jsou popsány v tabulce 5.16, jednotlivá pole datových tabulek pak v tabulkách 5.17 a 5.18.



■ **Obrázek 5.4** Pomocná data

■ **Tabulka 5.16** Pomocná data

Tabulka	Obsah
config	Konfigurační údaje
log	Aplikační log

■ **Tabulka 5.17** Pole tabulky config

Pole	Význam
config_id	Primární klíč
time	Čas posledního uložení hodnoty jako systémový čas v časové zóně GMT v milisekundách
date	Čas v časové zóně GMT posledního uložení hodnoty jako textový řetězec
key	Klíč odvozený od názvu položky na stránce pro konfiguraci aplikace uživatelského rozhraní správce aplikace
value	Hodnota konfigurační položky

■ **Tabulka 5.18** Pole tabulky log

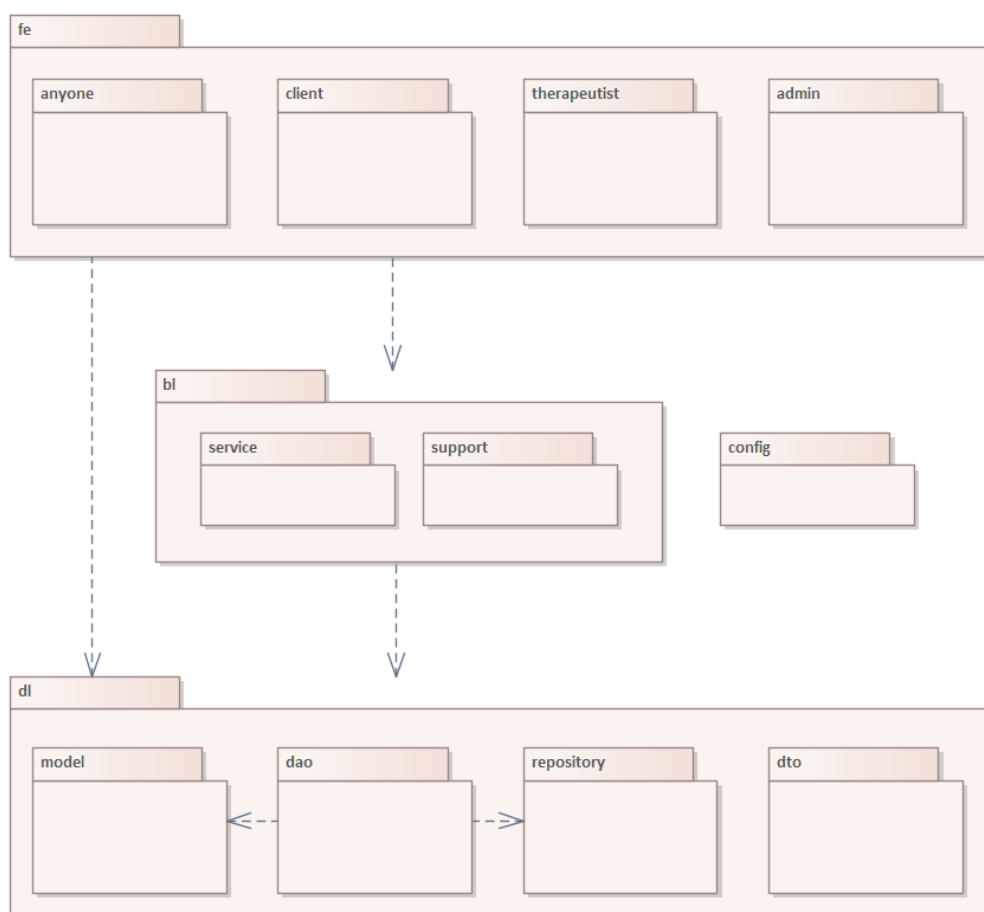
Pole	Význam
log_id	Primární klíč
time	Čas jako systémový čas v časové zóně GMT v milisekundách
date	Čas v časové zóně GMT jako textový řetězec
class	Třída, kde požadavek záznamu vznikl
level	Typ záznamu, používané hodnoty jsou „info“ a „error“
text	Text záznamu
details	Podrobnosti záznamu
user_id	Klíč identifikující přihlášeného uživatele, pokud je známý
as_user_id	Klíč identifikující uživatele, za kterého je uživatel přihlášen, pokud je známé
username	E-mailová adresa přihlášeného uživatele, pokud je známá
as_username	E-mailová adresa uživatele, za kterého je uživatel přihlášen, pokud je známá

5.1.2 Architektura aplikace

Architektura aplikace vychází z třívrstvé aplikační architektury:

- fe („front end“) představuje prezentační vrstvu
- bl („business layer“) představuje business vrstvu
- dl („data layer“) představuje datovou vrstvu

Architektura aplikace je zachycena na obrázku 5.5. Jednotlivé balíčky jsou popsány v tabulce 5.19.



■ **Obrázek 5.5** Architektura aplikace

■ **Tabulka 5.19** Balíčky tříd

Balíček	Obsah
fe/anyone	REST controllery pro stránky, které jsou přístupné bez přihlášení uživatele
fe/client	REST controllery pro stránky, které jsou přístupné klientům, terapeutům a správcům aplikace
fe/therapist	REST controllery pro stránky, které jsou přístupné terapeutům a správcům aplikace
fe/admin	REST controllery pro stránky, které jsou přístupné správcům aplikace
bl/service	Třídy obsahující metody pro obsluhu front endu aplikace a zpracování aplikačních dat
bl/support	Pomocné třídy
dl/model	Třídy objektově relačního modelu popisující tabulky v databázi
dl/dao	Třídy pro manipulaci s daty v databázi
dl/repository	Rozhraní CrudRepository pro práci s daty v databázi
dl/dto	Pomocné datové objekty používané v http požadavcích a odpovědích
config	Konfigurace aplikace

Obsah jednotlivých balíčků jsou popsány v tabulkách 5.20, 5.21, 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28, 5.27 a 5.30. Mimo balíčky stojí třída Application, která aplikaci inicializuje a eventuálně vytvoří správce uživatele s pevným názvem uživatelského účtu, pokud není v aplikaci definován.

■ **Tabulka 5.20** Obsah balíčku anyone

Třída	Obsah
Error	Obsluha stránky zobrazující chybovou hlášku
Homepage	Obsluha úvodní stránky aplikace
Login	Obsluha přihlašovací stránky
LoginBlocked	Obsluha výchozí stránky Spring Security Frameworku pro přihlášení, která uživatele přesměruje na aplikační stránku pro přihlášení
Logout	Obsluha odhlášení z aplikace
PasswordReset	Obsluha stránky obnovení hesla

■ **Tabulka 5.21** Obsah balíčku client

Třída	Obsah
ChangePassword	Obsluha stránky pro změnu hesla
CreateResponse	Obsluha stránky vypracování úlohy
NewTask	Obsluha stránky výběru typu úlohy
Responses	Obsluha stránky zobrazení vypracovaných úloh
SetPassword	Obsluha stránky nastavení hesla
ViewResponse	Obsluha stránky zobrazující vypracovanou úlohu

■ **Tabulka 5.22** Obsah balíčku therapist

Třída	Obsah
EditUser	Obsluha stránky editace údajů klienta nebo uživatele
Users	Obsluha stránky zobrazující klienty nebo uživatele

■ **Tabulka 5.23** Obsah balíčku admin

Třída	Obsah
Admin	Obsluha výchozí stránky správce aplikace
ChangeTherapist	Obsluha stránky hromadné změny terapeuta
Component	Obsluha stránky editace komponenty
Config	Obsluha stránky konfigurace aplikace
Page	Obsluha stránky s jednou html stránkou pro potřeby lokalizace
Pages	Obsluha stránky zobrazující html stránky pro potřebu lokalizace
ShowLog	Obsluha stránky zobrazující aplikační log
Template	Obsluha stránky editace šablony
Templates	Obsluha stránky zobrazení šablon
TestFileUpload	Obsluha stránky ověření odesílání data na server

■ **Tabulka 5.24** Obsah balíčku service

Třída	Obsah
ComponentService	Zpracování dat o komponentách šablon
DatabaseUserDetailsService	Implementace rozhraní UserDetailsService pro potřeby Spring Security Frameworku
PageMakerService	Metody zajišťující lokalizaci aplikace
ResponseService	Zpracování dat o úlohách
TaskController	Třída zajišťující stavové informace (uložené ve třídě Task) o provádění úloze – třída má zásadní význam pro řízení zpracování formulářů úloh: vede jejich seznam spolu s názvy složek pro přílohy; její scope je session
TaskTemplateService	Zpracování dat o šablonách úloh
UserService	Zpracování dat o uživateli

■ **Tabulka 5.25** Obsah balíčku support

Třída	Obsah
CaptchaValidator	Validace Google reCaptcha testu
Constant	Třída obsahující aplikační konstanty
Crypto	Metody používané pro zajištění bezpečnosti aplikace
Language	Číselník jazyků
LocalizationCache	Obecná třída zajišťující obecnou cache
LocalizationPageCache	Cache obsahující lokalizované stránky
LocalizationResourceCache	Cache obsahující lokalizované prvky html stránek
LocalizationTransformationCache	Cache obsahující transformační pravidla html stránek – seznam prvků html stránek
Logger	Třída používaná pro zápis aplikačního logu
LoginMonitor	Třída, která vede záznamy o počtu neúspěšných přihlášení v čase
MailingService	Třída zajišťující e-mailové služby
PageFilter	Třída zajišťující filtrování a stránkování stránek
Result	Číselník výsledků zpracování dat o komponentách šablon
Scheduler	Třída zajišťující pravidelně prováděné činnosti – udržování velikosti aplikačního logu a import souborů o zpracovaných úlohách
SpeedMonitor	Třída, která vede záznamy o objemu přijatých dat v čase
StaticLogger	Třída poskytující přístup k logu
Task	Třída, která vede údaje o prováděné úloze
UserSession	Pomocná třída vedoucí informace o aktuální session uživatele
VolumeMonitor	Obecná třída umožňující vést časové záznamy o událostech

■ **Tabulka 5.26** Obsah balíčku model

Třída	Obsah
Config	Popis tabulky config
HtmlElement	Popis tabulky html_element
HtmlElementText	Popis tabulky html_element_text
HtmlPage, HtmlPageEager, HtmlPageLazy	Popis tabulky html_page a varianty, které používají pro tabulky s cizím klíčem z této tabulky přístup Eager nebo Lazy
Link	Popis tabulky link
Listed	Rozhraní pro třídy definující pořadí
ListedIdentifier	Třída implementující rozhraní Listed a poskytující identifikátor
Log	Popis tabulky log
Name	Rodičovská třída tříd obsahujících atribut name
Response	Třída popisující tabulku response
ResponseName	Třída popisující tabulku response_name
TaskTemplate, TaskTemplateEager, TaskTemplateLazy	Popis tabulky task_template a varianty, které používají pro tabulky s cizím klíčem z této tabulky přístup Eager nebo Lazy
TaskTemplateComponent	Popis tabulky task_template_component
TaskTemplateComponentDetail	Popis tabulky task_template_component_detail
TaskTemplateComponentName	Popis tabulky task_template_component_name
TaskTemplateName	Popis tabulky task_template_name
Timed	Rodičovská třída tříd obsahujících atributy time a date
User, UserEager a UserLazy	Popis tabulky user a varianty, které používají pro tabulky s cizím klíčem z této tabulky přístup Eager nebo Lazy

■ **Tabulka 5.27** Obsah balíčku dao

Třída	Obsah
ConfigDao, FilterDao, HtmlElementDao, HtmlPageDao, LinkDao, ResponseDao, ResponseNameDao, TaskComponentDao, TaskTemplateDao, UserDao	Třídy zajišťující přístup k tabulkám config, html_element, html_page, link, response, response_name, task_template, task_template_component, user
FilterDao	Abstraktní třída zajišťující filtrování při zobrazení seznamů
LogDao	Třída zajišťující přístup k tabulce log pro třídy, které mají přístup k session uživatele
SystemLogDao	Třída zajišťující přístup k tabulce log pro třídy, které nemají přístup k session uživatele
ListedDao	Abstraktní třída zajišťující změnu pořadí šablon nebo komponent

■ **Tabulka 5.28** Obsah balíčku repository

Třída	Obsah
ConfigRepository, HtmlElementRepository, HtmlElementTextRepository, HtmlPageEagerRepository, HtmlPageLazyRepository, LinkRepository, LogRepository, ResponseNameRepository, ResponseRepository, TaskTemplateComponentDetailRepository, TaskTemplateComponentNameRepository, TaskTemplateComponentRepository, TaskTemplateEagerRepository, TaskTemplateLazyRepository, TaskTemplateNameRepository, UserRepositoryEager, UserRepositoryLazy	CrudRepository pro třídy, od jejichž názvu byly názvy tříd odvozeny

■ **Tabulka 5.29** Obsah balíčku dto

Třída	Operace, kde se třída používá
ChangePasswordDto	Změna hesla
ChangeTherapistDto	Hromadná změna terapeuta
CreateElementDto	Vytvoření prvku na html stránce
CreateResponseDto	Odeslání formuláře s vypracovanou úlohou
EditUserDto	Editace uživatele nebo klienta
EmailCredentialsDto	Uložení nebo ověření přístupových údajů k e-mailovému serveru
FilterDto	Společná pole pro filtrování seznamů
LoginDto	Přihlášení uživatele
LogsFilterDto	Filtrování logů
NewPageDto	Vložení nové html stránky
NewTaskTemplateComponentDto	Vložení nové komponenty šablony nebo její změna
NewTaskTemplateDto	Vložení nové šablony
OutputDto	Obecná odpověď serveru uživatelskému rozhraní; obsahuje výsledek operace, zprávu k zobrazení a eventuálně další detaily jako například nový obsah stránky
PasswordResetDto	Obnovení hesla
RenamePageDto	Přejmenování html stránky
RenameTaskTemplateDto	Přejmenování šablony
ResponsesFilterDto	Filtrování provedených úloh
SaveConfigDto	Uložení konfiguračního klíče a hodnoty
SaveElementDto	Uložení prvku html stránky
SetPasswordDto	Nastavení hesla
TaskTemplatesFilterDto	Filtrování šablon úloh
UsersFilterDto	Filtrování uživatelů

■ **Tabulka 5.30** Obsah balíčku config

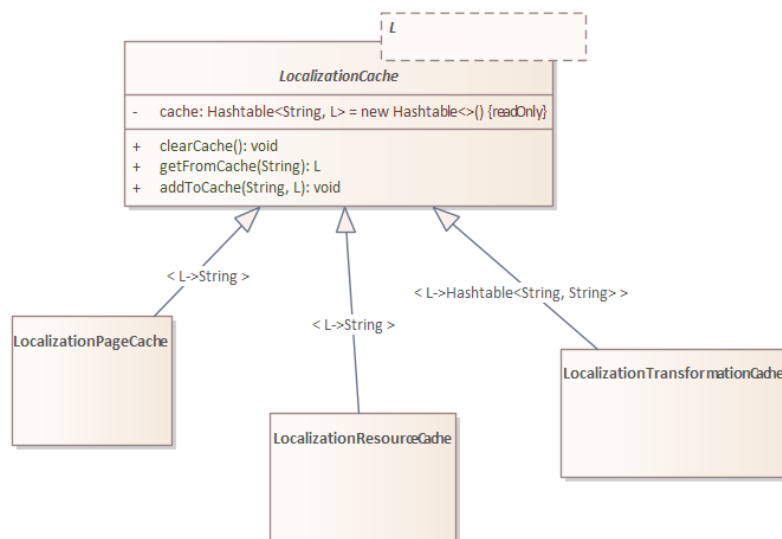
Třída	Obsah
AppConfig	Třída povolující použití Spring Framework scheduleru
SecurityConfiguration	Třída s konfigurací Spring Security Frameworku

5.1.3 Využití dědičnosti

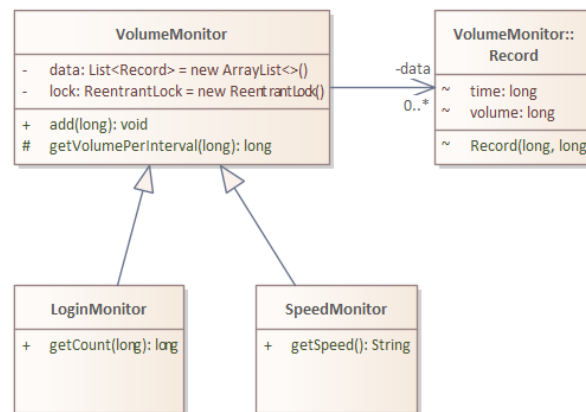
V této části práce popisuji, jak jsem v aplikaci využil dědičnost. V některých případech jsem dědičnost použil ke zjednodušení kódu a zvýšení jeho přehlednosti, zatímco v jiných případech bylo jediným důvodem splnění limitu nejvýše 3 % detekované duplicity kódu ve statické analýze kódu.

5.1.3.1 Caching a monitoring

Velmi přímočaré využití dědičnosti jsem implementoval v případě cachování a monitoringu četnosti událostí. Obecná třída implementuje obecný algoritmus, který je u dědicích tříd použit v případě cachování (na obrázku 5.6) k uložení lokalizovaných html stránek, jejich částí nebo transformačních pravidel, v případě monitoringu (na obrázku 5.7) stejný algoritmus rodičovské třídy používají třídy pro sledování počtu nesprávných přihlášení během určitého času nebo sledování objemu přijatých dat).



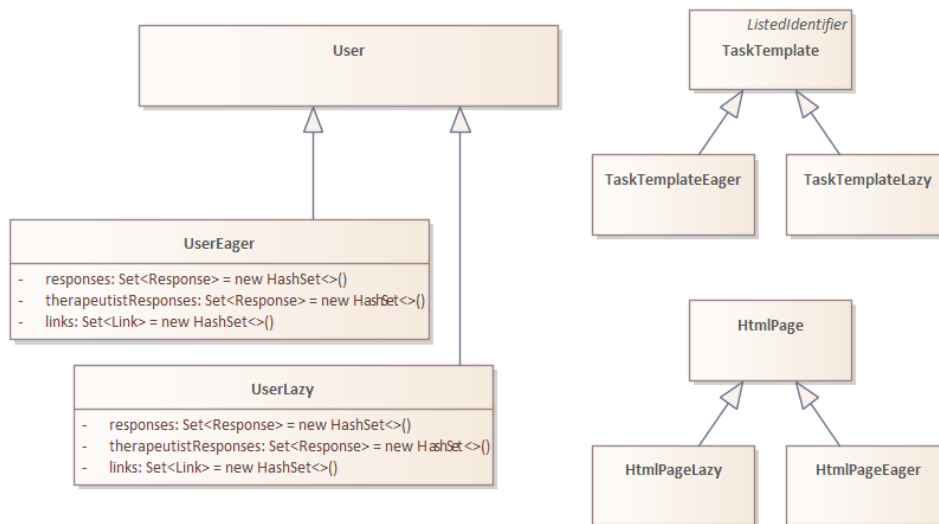
■ **Obrázek 5.6** Použití dědičnosti u cachování



■ **Obrázek 5.7** Použití dědičnosti u monitoringu

5.1.3.2 Lazy vs. Eager

Zcela jiné uplatnění dědičnosti jsem použil při implementaci načítání dat z databáze metodou Lazy nebo Eager (při použití metody Lazy jsou načtena související data až v případě jejich potřeby, zatímco při použití metody Eager jsou tato data načtena současně s původním záznamem). Stejná třída je z optimalizačních důvodů jednou implementována jako třída využívající metodu Lazy a jednou jako třída využívající metodu Eager. Obě tyto třídy dědí ostatní atributy z rodičovské třídy. Na obrázku 5.8 jsou detaily implementace zobrazeny pro třídu User, pro ostatní třídy je zachycen pouze jejich vzájemný vztah.

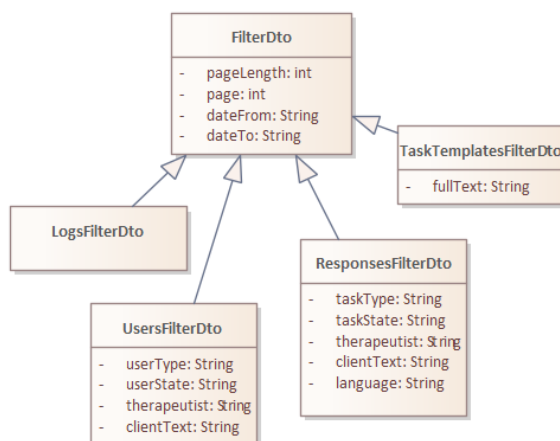


■ **Obrázek 5.8** Použití dědičnosti pro metody Lazy vs. Eager

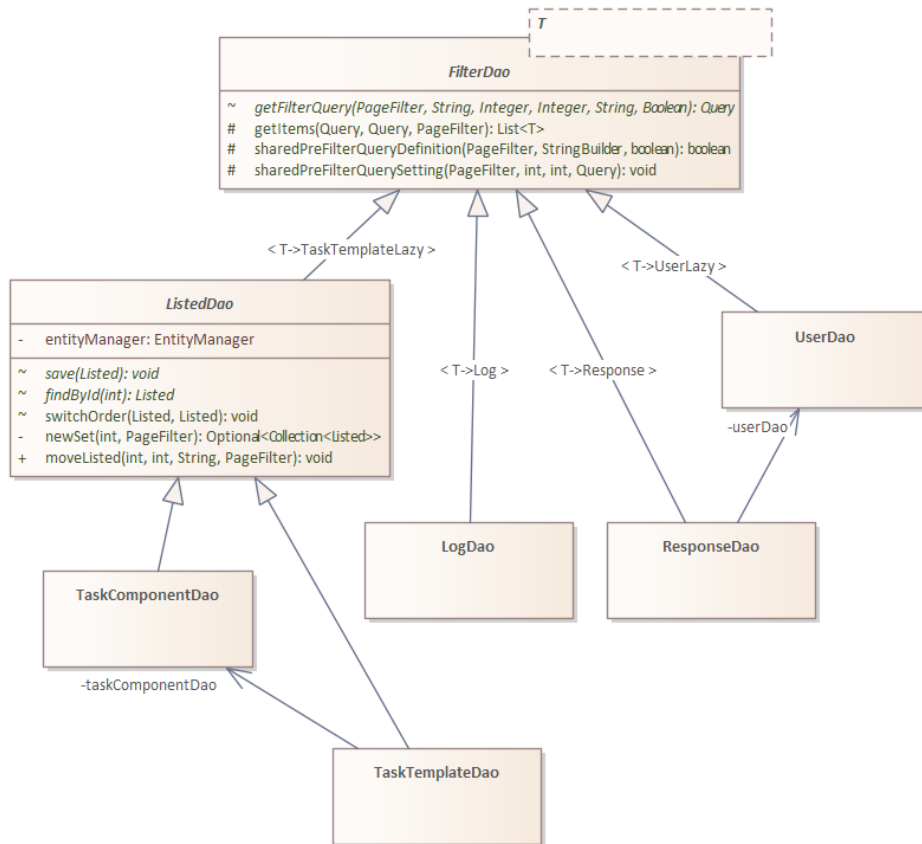
5.1.3.3 Filtrování a stránkování

V aplikaci je filtrování a stránkování použito pro zobrazení seznamu uživatelů, úloh, šablon a logů. Situace je komplikovaná funkcí řazení položek na zobrazení seznamu šablon a komponent

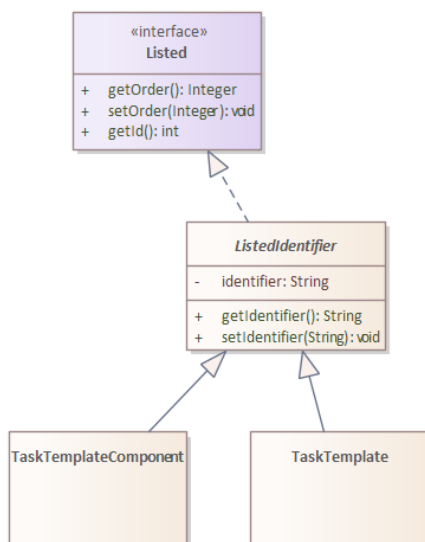
šablon. Při implementaci jsem použil dědičnost tříd pro přenos dat z uživatelského rozhraní na server (na obrázku 5.9), dědičnost tříd pro manipulaci s uloženými daty (na obrázku 5.10) a v případě implementace změny pořadí také dědičnost samotných datových tříd představujících šablony a jejich komponenty (na obrázku 5.11).



■ **Obrázek 5.9** Dědičnost nastavení filtru



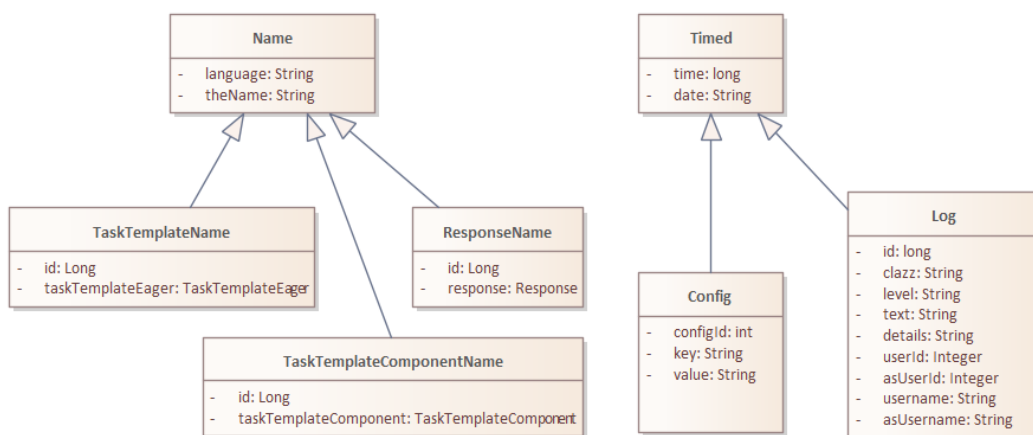
■ **Obrázek 5.10** Dědičnost filtrování a stránkování



■ **Obrázek 5.11** Dědičnost řazení položek

5.1.3.4 Deduplikace kódu

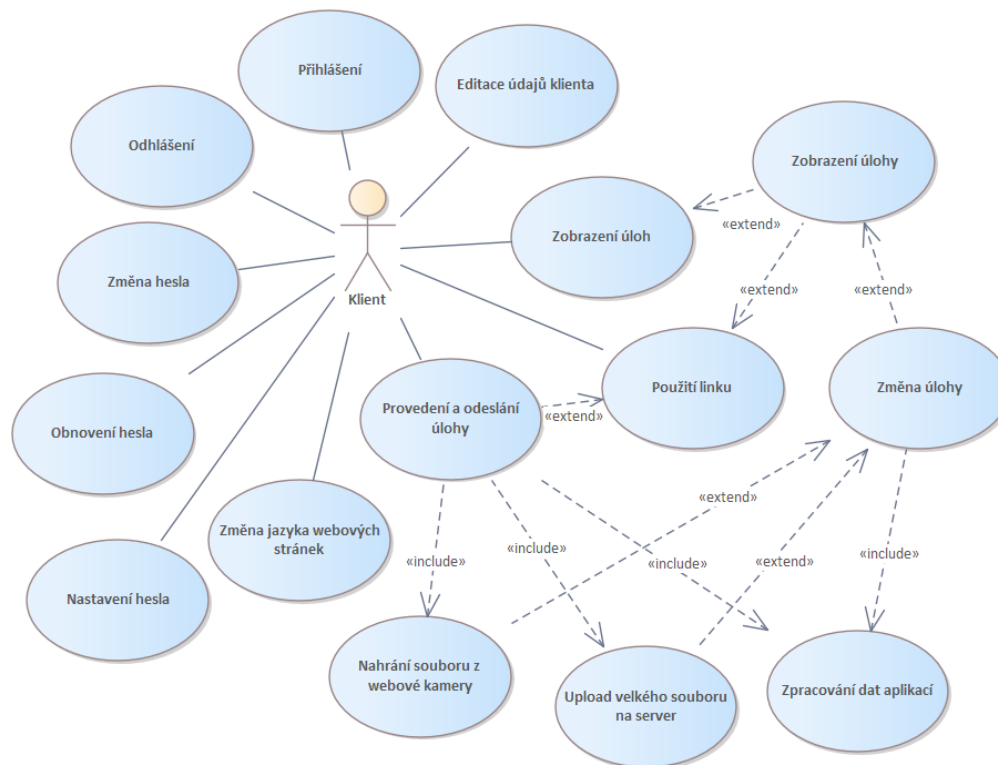
Pro třídy na obrázku 5.12 jsem dědičnost použil pouze k odstranění detekované duplicity kódu statickou analýzou kódu – třídy mají shodné atributy a proto je možné tuto detekovanou duplicitu kódu odstranit s využitím dědičnosti, přehlednost kódu se tím ale nezvyšuje.



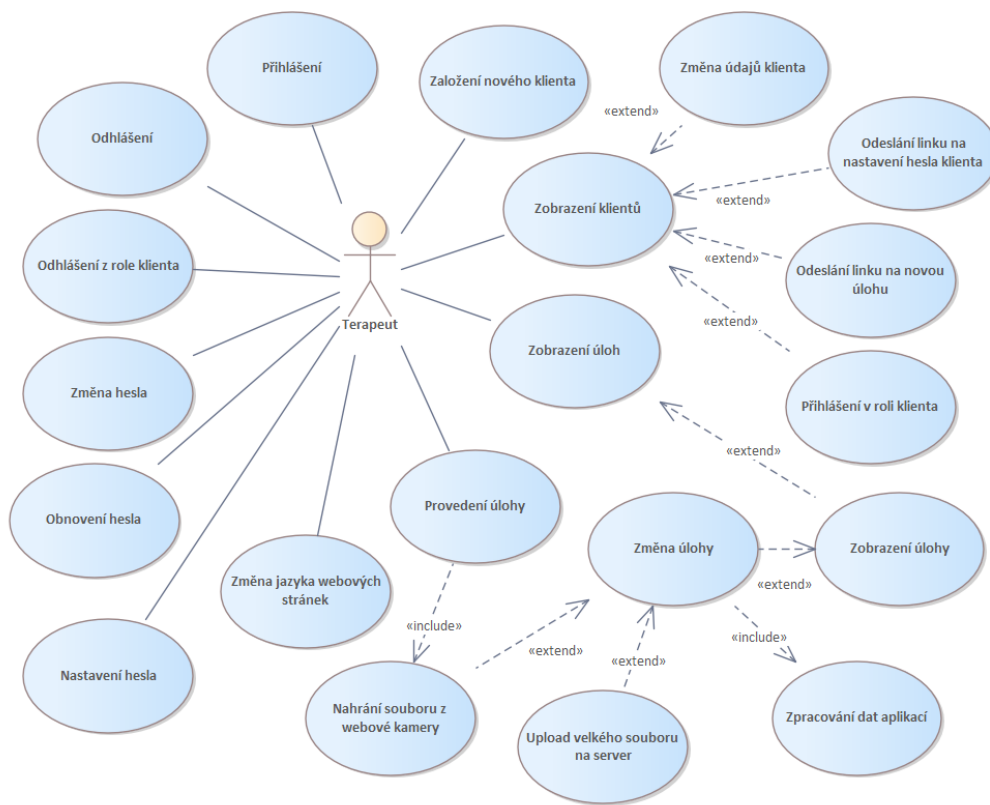
■ **Obrázek 5.12** Deduplikace kódu pomocí dědičnosti

5.1.4 Případy užití

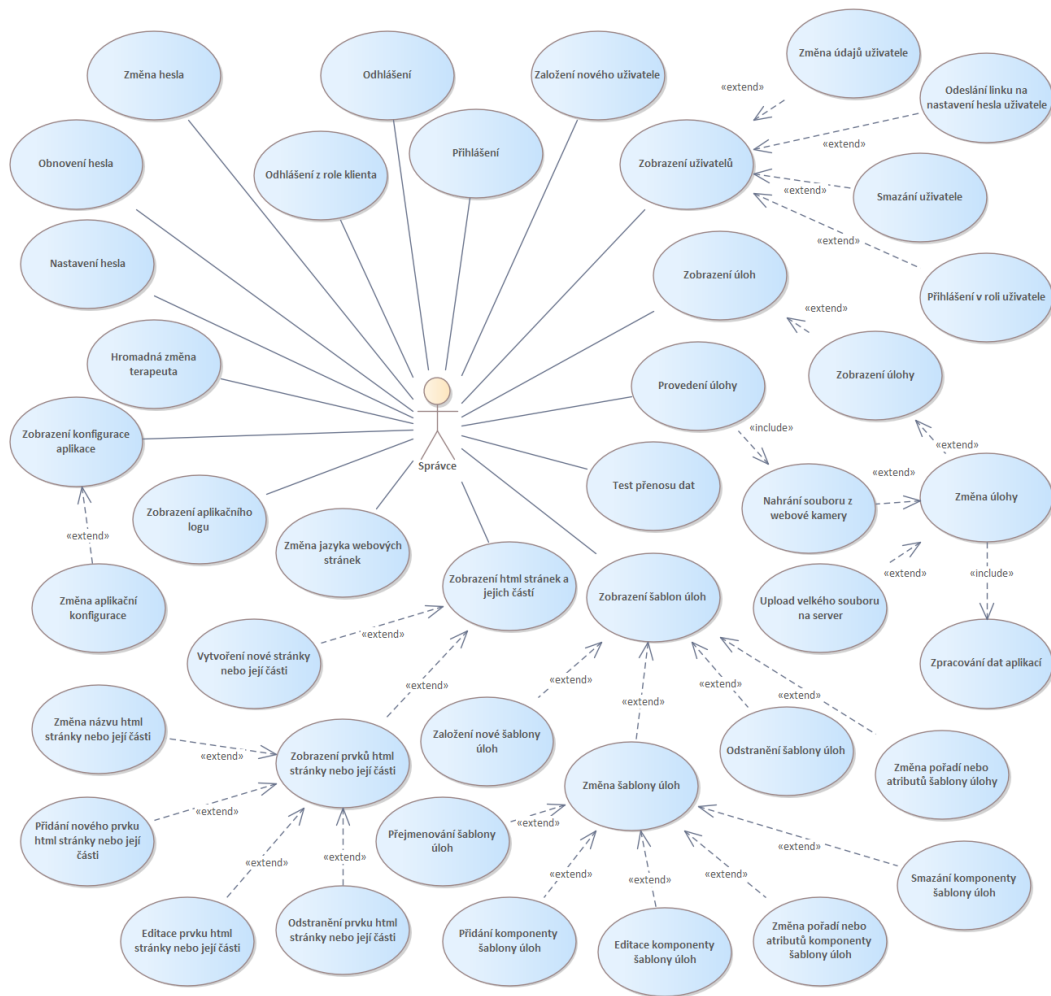
Pro snazší orientaci v části 5.1.5 jsou v této části zobrazeny diagramy případů užití pro uživatele v roli klient, terapeut nebo správce na obrázcích 5.13, 5.14 a 5.15. Podrobnější popis průběhu případů užití je k dispozici v návrhu specifikace v kapitole D a v uživatelské příručce v kapitole B.



■ **Obrázek 5.13** Diagram případů užití uživatele v roli klient



■ Obrázek 5.14 Diagram případů užití uživatele v roli terapeut



■ **Obrázek 5.15** Diagram případů užití uživatele v roli správce

5.1.5 Popis zajištění bezpečnosti

Zajištění bezpečnosti aplikace vychází z použití Spring Security frameworku. V této kapitole popisují jeho využití v aplikaci včetně doplňkových metod, které aplikace pro zajištění bezpečnosti používá.

5.1.5.1 Konfigurace Spring Security frameworku

Výchozí stránka frameworku pro vložení uživatelského jména a hesla je nahrazena vlastní stránkou pro přihlášení uživatele, která uživatelskou identifikaci frameworku předává.

V aplikaci vystupují uživatelé v jedné z následujících systémových rolí:

- ROLE_CLIENT
- ROLE_THERAPEUTIST
- ROLE_ADMIN

Tyto systémové role jsou uživatelům přiřazeny podle tabulky 5.31.

■ **Tabulka 5.31** Přiřazení systémových rolí

Typ / aplikační role uživatele	Systémová role
správce	ROLE_ADMIN, ROLE_THERAPEUTIST, ROLE_CLIENT
terapeut	ROLE_THERAPEUTIST, ROLE_CLIENT
klient	ROLE_CLIENT

Přístup na jednotlivé stránky aplikace je konfigurován tak, že stránky (a obecně adresy souvisejících REST služeb) pro správce začínají řetězcem `/admin`, stránky pro terapeuty řetězcem `/therapist` a stránky pro klienty řetězcem `/client`. Tím je zajištěno, že klienti mohou využívat pouze stránky určené pro klienty a že terapeuti nemohou využívat stránek určených pro správce.

V tabulce 5.32 je seznam všech aplikačních stránek. Některé stránky uvedené v tabulce mají více různých dalších adres (které v tabulce uvedeny nejsou), které mění jejich chování.

■ **Tabulka 5.32** Hlavní adresy stránek aplikace nebo významných služeb

Adresa	Obsah
/	Úvodní stránka aplikace
/login	Přihlašovací stránka
/passwordreset	Obnova hesla
/setpassword	Nastavení hesla
/logout	Nastavení hesla
/error	Stránka pro zobrazení chyby – například přístup na stránku bez potřebné systémové role
/admin	Úvodní stránka správce
/admin/newuser	Založení uživatele správcem
/admin/users	Zobrazení uživatelů správci
/admin/edit	Editace údajů uživatele správcem
/admin/changetherapist	Hromadná změna terapeuta správcem
/admin/newtask	Výběr nové úlohy správcem
/admin/createresponse	Provedení úlohy správcem
/admin/responses	Zobrazení úloh správcem
/admin/response	Zobrazení úlohy správcem
/admin/config	Konfigurace aplikace správcem
/admin/templates	Zobrazení šablon správcem
/admin/template	Zobrazení šablony správcem
/admin/component	Zobrazení komponenty správcem
/admin/pages	Zobrazení html stránek správcem
/admin/page	Zobrazení prvku html stránky správcem
/admin/showlog	Zobrazení logu správcem
/admin/testfileupload	Ověření rychlosti nahrávání souborů správcem
/admin/changepassword	Změna hesla správce
/therapist/responses	Zobrazení úloh terapeutem
/therapist/response	zobrazení úlohy terapeutem
/therapist/newtask	Výběr nové úlohy terapeutem
/therapist/createresponse	Provedení úlohy terapeutem
/therapist/clients	Zobrazení klientů terapeuta
/therapist/edit	Editace údajů klienta terapeuta
/therapist/newclient	Založení klienta terapeuta
/therapist/changepassword	Změna hesla terapeuta
/client/edit	Editace údajů klienta
/client/newtask	Výběr nové úlohy klientem
/client/createresponse	Provedení nové úlohy klientem
/client/responses	Zobrazení úloh klienta
/client/response	Zobrazení úlohy klienta
/client/changepassword	Změna hesla klienta
/createlinkedresponse	Vytvoření úlohy podle linku
/viewlinkedresponse	Zobrazení úlohy podle linku
/responsefile	Stažení přílohy úlohy
/filelinkedupload	Odeslání přílohy úlohy prováděné podle linku

5.1.5.2 Uživatelské heslo

Uživatelské heslo je uloženo v databázi aplikace jako hash vytvořený funkcí Spring Frameworku `BCryptPasswordEncoder()` [82] nad hashem SHA-512 hesla vytvořeným systémovou funkcí Javy

a převedeným do kódu Base64 systémovou funkcí Javy:

```
ulozeneHeslo = bCryptEncode(base64Encode(sha512(password)));
```

Tento způsob je zvolen proto, aby bylo možné zajistit trvalé přihlášení uživatele v prohlížeči s využitím Cookies a v prohlížeči nebylo uloženo heslo uživatele v čitelné podobě.

5.1.5.3 Trvalé přihlášení

Implementace trvalého přihlášení je založena na využití Cookies prohlížeče. V prohlížeči je uloženo heslo jako hash SHA-512 hesla uživatele převedený do Base64 kódování:

```
cookieUsername = base64Encode(username);
cookiePassword = base64Encode(sha512(password));
```

Při odhlášení jsou Cookies z aktuálního prohlížeče odstraněny. Při změně hesla uživatele uložené hodnoty přihlášení uživatele neumožňují.

5.1.5.4 Nastavení prvního hesla

Při založení uživatele aplikace odešle na určenou e-mailovou adresu link k nastavení prvního hesla uživatele. Link obsahuje informace, které aplikace použije k přihlášení uživatele v rámci frameworku Spring Security, a pak uloží nové heslo přihlášenému uživateli.

Aplikace před odesláním linku nejprve vygeneruje dočasné heslo uživatele jako náhodný 10 místný alfanumerický řetězec a převede jej do kódu Base64. Toto heslo je uloženo v databázi pomocí funkce `BCryptPasswordEncoder()` na místě určeném právě pro dočasná hesla, je tedy uloženo v jiném poli než hesla trvalá – toto uspořádání je zvoleno proto, aby při odeslání linku na obnovení hesla, které využívá shodný mechanismus, mohl uživatel dále používat svoje běžné heslo. Aplikace následně odesílá link, který má dvě části – identifikaci uživatele a heslo. Identifikace uživatele je hexadecimální reprezentací jeho e-mailové adresy a heslo je hexadecimální reprezentací kódu v Base64:

```
linkedUsername = hex(username);
linkedPassword = hex(base64Encode(randomString(10)));
```

Při použití linku aplikace ověří link pomocí Spring Security frameworku. Při ověření pozmění uživatelské jméno tak, že před něj předradí text „SP: “. Tento speciální řetězec následně rozpozná aplikační funkce, kterou Spring Security framework při kontrole hesla zavolá, a ta vrátí místo trvalého uživatelského hesla heslo dočasné, které bylo použito při generování linku.

5.1.5.5 Obnova hesla

Obnova hesla využívá shodného mechanismu jako nastavení prvního hesla popsané v odstavci 5.1.5.4.

5.1.5.6 Link na úlohu

Aplikace používá link na provedení úlohy a na zobrazení úlohy. V obou případech link jednoznačně identifikuje uživatele a objekt, který má být zobrazen – formulář pro vytvoření úlohy nebo formulář již provedené úlohy.

Linky jsou vedeny v databázi a aplikace uchovává nejvýše 100 linků jednoho uživatele.

Při generování linku je potřeba zajistit jednoznačnost a náhodnost linku. Link je proto tvořen hexadecimální reprezentací primárního klíče linku a náhodnou posloupností znaků:

```
link = hex(linkId + "-" + base64Encode(sha512(randomString(20))));
```

Při ověření linku je text linku nalezen v databázi linků.

5.1.5.7 Zajištění bezpečnosti aplikační logikou

Při zobrazení úlohy nebo příloh úloh aplikace kontroluje, že přihlášený uživatel nebo uživatel identifikovaný pomocí linku má k úloze přístup – je klientem, který úlohu provedl, jeho aktuálním terapeutem nebo správcem.

Při zobrazení údajů klienta aplikace kontroluje, že přihlášený uživatel má k údajům přístup – je klientem, jehož údaje jsou zobrazeny, nebo jeho aktuálním terapeutem nebo správcem.

Při odeslání příloh prováděné úlohy aplikace uloží soubor do složky, kterou vytvořil stejný uživatel, který soubor odesílá. Obdobně data úlohy aplikace přiřadí uživateli a uloží do složky, kterou vytvořil uživatel, který byl při vytvoření formuláře úlohy identifikován linkem nebo byl přihlášen do aplikace.

5.1.6 Testování

Součástí zadání je požadavek navrženou databází a aplikaci implementovat a ověřit její funkčnost.


Ověření funkce databáze a aplikace proběhlo na několika úrovních. Při vývoji jednotlivých funkcí jsem správnost implementace a soulad s požadavky ověřil. Během vývoje aplikace jsem funkčnost aplikace demonstroval odbornému konzultantovi Ing. Janu Hejdovi, Ph.D., a jednotlivé aspekty jsem s ním pravidelně konzultoval.

Podporovanými prohlížeči jsou prohlížeče Chrome a Firefox. Na zařízení iPad 2018 s operačním systémem iOS byla aplikace testována také s prohlížečem Safari – pro správnou funkci tohoto prohlížeče je potřeba v nastavení prohlížeče zapnout experimentální funkčnost „Media recorder“.

5.1.6.1 Statická analýza kódu

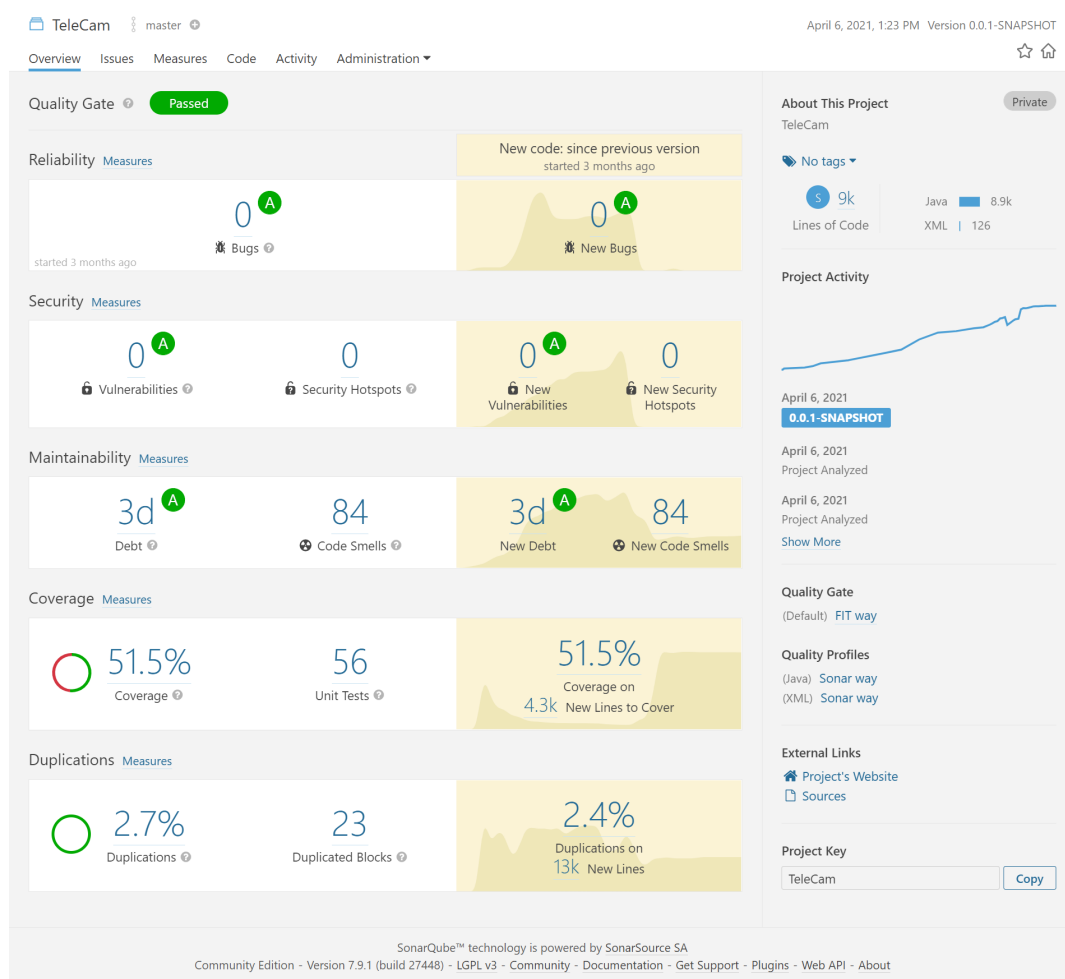
Kód aplikace má dvě části – serverová část je napsána v jazyce Java a frontendová část je napsána v html / Javascriptu. Serverová část v jazyce Java byla podrobena statické analýze s využitím nástroje SonarQube [71].

Test byl proveden na fakultní instalaci nástroje SonarQube s profilem „FIT way“ na obrázku 5.16.

Metric 	Operator	Error
Coverage on New Code	is less than	50.0%
Duplicated Lines on New Code	is greater than	3.0%
Maintainability Rating on New Code	is worse than	A
Reliability Rating on New Code	is worse than	A
Security Rating on New Code	is worse than	A

■ **Obrázek 5.16** SonarQube – nastavení „FIT way“

Všechny nálezy nástroje jsem prošel a všechny nálezy, kde podle mého názoru mělo smysl kód upravit, jsem kód upravil. V případech, kdy jsem nálezy ignoroval, by úprava podle mého názoru přehlednost kódu snížila. Dále jsem kód upravil tak, aby rozsah duplicit v kódu byl menší než požadovaný. Kód byl analyzován celý včetně tříd popisujících datové tabulky v databázi. Splnění požadovaného limitu nejvýše 3 procent duplicit si vyžádalo zavedení dědičnosti i v případech, kde bych se jí jinak vyhnul (viz odstavec 5.1.3.4). Výstup nástroje ke dni 6. dubna 2021, kdy byly dokončeny i uživatelské testy a všechny chyby odstraněny, je na obrázku 5.17.



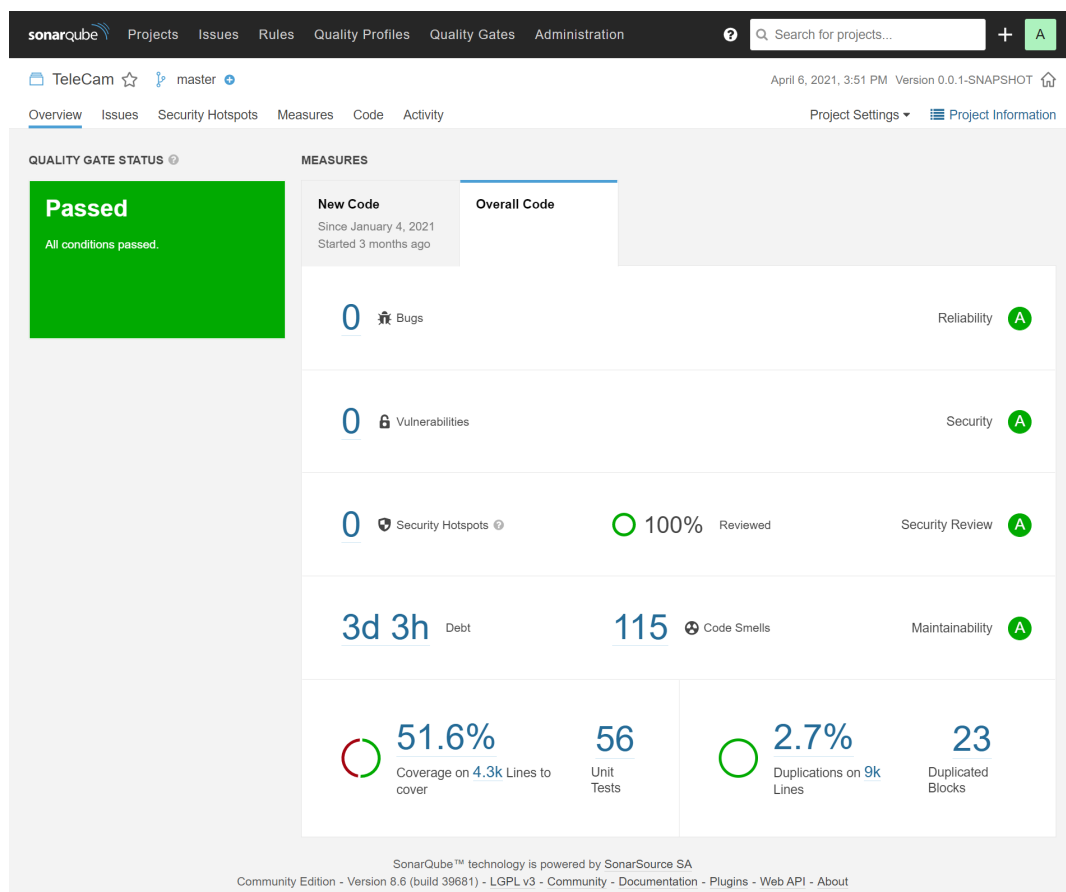
■ **Obrázek 5.17** SonarQube – výstup fakultní instalace

V průběhu implementace jsem používal také lokální instalaci nástroje SonarQube v jiné verzi, která běžela na mém počítači v Docker kontejneru [83]. Kritéria kvality jsem nastavil shodně jako na fakultní instalaci – minimální pokrytí testy jsem snížil z původních 80 % na limit 50 %. Ostatní nastavení jsem ponechal beze změny tak, jak je vidět na obrázku 5.18. Výsledky analýzy jsou zobrazeny na obrázku 5.19.

Conditions on New Code

Metric	Operator	Value	Edit	Delete
Coverage	is less than	50.0%		
Duplicated Lines (%)	is greater than	3.0%		
Maintainability Rating	is worse than	A		
Reliability Rating	is worse than	A		
Security Hotspots Reviewed	is less than	100%		
Security Rating	is worse than	A		

Obrázek 5.18 SonarQube – nastavení lokální instalace



Obrázek 5.19 SonarQube – výstup lokální instalace

Drobné odlišnosti výsledku analýzy fakultní a lokální instalace jsou způsobeny tím, že se jedná o jinou verzi nástroje.

5.1.6.2 Automatizované testy

Aplikace byla pokryta automatizovanými testy, které ověřují funkčnost implementované aplikace a databáze. Automatizované testy běží proti testové instanci databáze a ta musí obsahovat běžnou konfiguraci aplikace v rozsahu výchozí konfigurace lokalizace, šablony pro osobní údaje použité pro úlohy a konfigurace funkčního e-mailového serveru; v průběhu integračních testů je odeslána e-mailová zpráva na adresu `testingX@telecam.cz`, kde X je celé číslo. Proti jedné testové databázi může současně běžet nejvýše jedna instance testů, jinak může dojít k chybě.

Ačkoliv automatizované testy pokrývají více než 50 % kódu aplikace a ověřují funkčnost většiny hlavních funkcí aplikace, mají charakter spíše smoke testů, které sice odhalí neschopnost aplikace provést daný úkol, ale neověřují správnost dílčích aplikačních výstupů.

5.1.6.3 Uživatelské testy

Aplikace byla testována zadavatelem dne 31. března 2021 s využitím prostředí popsaného v odstavci 5.2. Z testování vzešlo 19 připomínek, které byly zaznamenány ve fakultním GitLabu jako issues. V tabulce 5.33 jsou tyto připomínky uvedeny ve skupinách po kategoriích připomínek.

■ **Tabulka 5.33** Připomínky z uživatelského testování

Počet	Kategorie	Stav
7	Chyba	Opraveno
5	Požadavek na změnu funkčnosti	Implementováno
3	Nesrozumitelná funkčnost	Funkčnost změněna
2	Podnět k možné změně funkčnosti	Diskutováno a uzavřeno
1	Požadavek uzavřen v rámci testování	Uzavřeno
1	Nejasné určení chyby	Diskutováno a uzavřeno jako nereprodukovatelné

Žádné další připomínky nebyly vzneseny.

5.2 Demonstrační instalace

Aplikace byla nasazena na Ubuntu Server 18.04 LTS s databází PostgreSQL verze 10.16.

Jako server byl použit virtuální privátní server společnosti Wedos nabízený pod obchodním názvem VPS SSD v kombinaci s balíčkem Profi. Hardwarová konfigurace byla 1 vyhrazené vlákno procesoru Xeon 1.70 GHz, 4 GB RAM DDR3 a SSD diskový prostor v konfiguraci RAID 10 sdílený s dalšími zákazníky. Na serveru současně běžely i další aplikace, ale žádná z nich server výrazně nezatěžovala. V této konfiguraci byla odezva aplikace na běžné operace jednoho přihlášeného uživatele přibližně do 100 ms s výjimkou prvních volání stránek nebo jejich částí, které od spuštění aplikace ještě nebyly v daném jazyce použity a proto nebyly cachovány. Zátěžové testy nebyly prováděny.

Jako e-mailový server byl použit server společnosti Active24 nabízený pod obchodním názvem balíček Firma produktu Webhosting Windows.

Data o vypracovaných úlohách byla sdílena s využitím Wedos disku. Přístup aplikace na Wedos disk byl zajištěn pomocí protokolu CIFS; sdílení souborů s návaznou aplikací pak pomocí protokolu rsync.

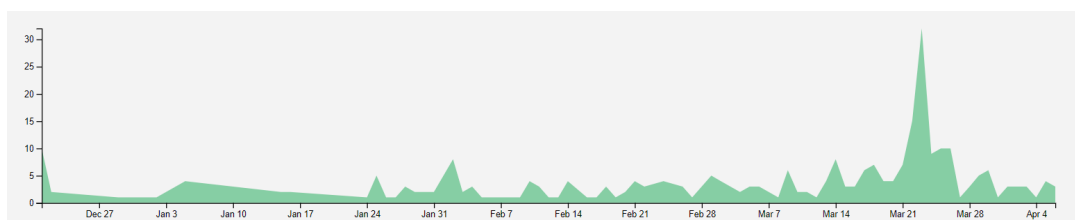
5.3 Projektová dokumentace

V analytické fázi projektu byl jediným artefaktem projektové dokumentace sdílený Google sheet s termíny a úkoly, které je potřeba splnit. Výstupem této fáze byl z hlediska projektové doku-

mentace návrh specifikace, který je v příloze D. Tento návrh specifikace shrnuje obsah diskusí s odborným konzultantem Ing. Janem Hejdou, Ph.D., (dále jen odborný konzultant) a ukotvil hlavní rysy aplikace. Dokument byl připomínkován odborným konzultantem a všechny jeho připomínky byly zapracovány.

Součástí dokumentu návrhu specifikace je i předpokládaný časový harmonogram implementace aplikace, dokončení a odevzdání bakalářské práce, který byl v následujících fázích v zásadě dodržen.

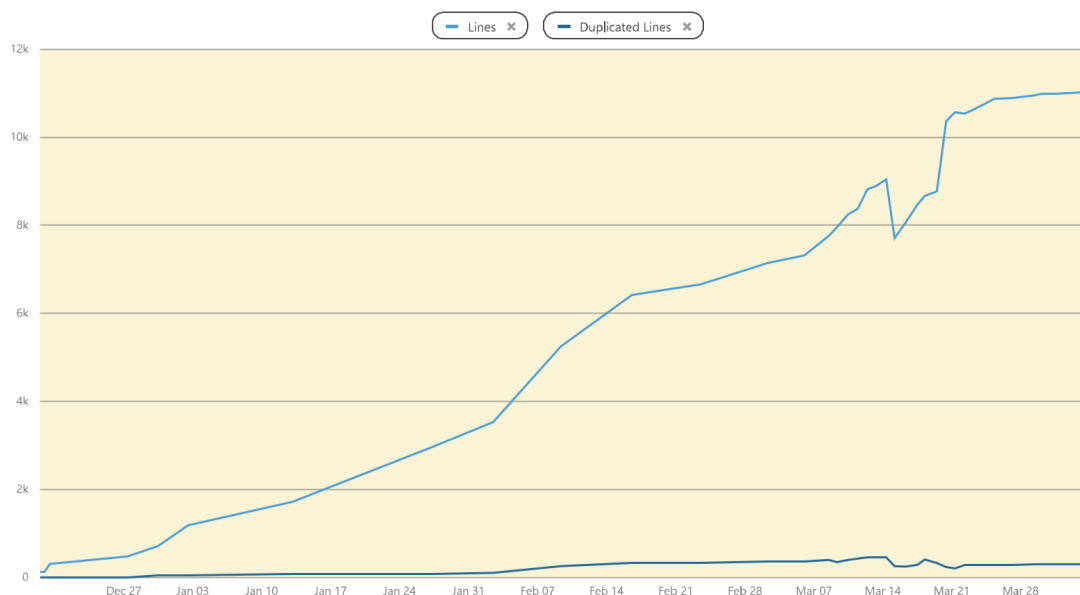
Po začátku implementace aplikace je implementační úsilí dokumentováno postupným růstem rozsahu kódu uloženého ve fakultním GitLabu, kam měl odborný konzultant přístup a kam také vložil ve formě issues připomínky vzešlé z uživatelského testování.



Obrázek 5.20 Počet zapsání (commit) nového kódu v čase

Na obrázku 5.20 je počet zapsání (commit) nového kódu do fakultního GitLabu v průběhu projektu. Prudký nárůst na konci března souvisí s odstraňováním issues nalezených statickou analýzou kódu. Obrázek je vygenerován fakultní instalací GitLabu.

Na obrázku 5.21 je znázorněn časový průběh počtu řádek zdrojového kódu aplikace a časový průběh duplicitních řádek kódu. Propad počtu v závěru března je způsobem dočasným vynětím tříd popisujících databázové objekty z analýzy. Obrázek je vygenerován fakultní instalací nástroje SonarQube.



Obrázek 5.21 Růst počtu řádek zdrojového kódu v čase

Jednotlivé dílčí funkčnosti byly implementovány do dílčích větví, které pak byly sloučeny do hlavní větve zdrojového kódu. Byla přitom využita funkčnost GitLabu Continuous Integration,

kdy nutnou podmínkou sloučení bylo zkompilování aplikace a bezchybné proběhnutí automatických testů. Po sloučení do hlavní větve automaticky následovala statická analýza nástrojem SonarQube.

Po dokončení implementace vznikl text této práce, který je včetně příloh a obrázků v GitLabu rovněž uložen.

5.3.1 Uživatelská příručka

Uživatelská příručka je k dispozici v příloze B.

5.3.2 Instalační příručka

Instalační příručka je k dispozici v příloze C. Instalační příručka obsahuje také popis souborů o provedených úlohách a jejich vyhodnocení, které jsou vyměňovány s návaznou aplikací.

5.3.3 Návrh specifikace

Návrh specifikace je k dispozici v příloze D.

5.4 Licenční požadavky

Aplikace není zatížena žádnými licenčními požadavky, které by znemožňovaly její další úpravy nebo komerční použití. V této části práce jsou uvedeny použité nástroje, komponenty nebo části kódu. Pokud je použití některého z artefaktů omezeno licencí, jsou v této části uvedeny podrobnosti.

5.4.1 Vývojové a pomocné nástroje

Vývoj probíhal na počítači s komerční licencí operačního systému Microsoft Windows 10 Professional a s použitím komerční licence Microsoft Office Professional 2016.

Při vývoji aplikace a při přípravě dokumentace byly použity komerční licence následujících nástrojů:

- Balsamiq Mockups 3 [84]
- Enterprise Architect 15.2 Corporate Edition [85]
- PDF-XChange Editor [86]
- Jasc Paint Shop Pro 7.01 [87]
- IntelliJ IDEA Ultimate 2020.3 [88]

Licenční ujednání SonarQube Community Edition [71] použité v lokálně provozovaném kontejneru Docker je k dispozici na adrese <http://www.gnu.org/licenses/lgpl-3.0.txt> a je uloženo na médiu ve složce `/docs/licence/SonarQube`. Licenční ujednání Dockeru je k dispozici na adrese <https://www.docker.com/legal/docker-terms-service> a je uloženo na médiu ve složce `/docs/licence/Docker`.

5.4.2 Vlastní aplikace

Aplikace je napsána v jazyce Java 11. K dispozici jsou dvě verze – Open JDK [89], kterou je možné použít bezplatně, a Oracle JDK [90], kdy je potřeba mít ke komerčnímu provozu zakoupenou placenou licenci. Licenční ujednání varianty Open JDK je dostupné na adrese <https://openjdk.java.net/legal/gplv2+ce.html> a je uloženo na médiu ve složce `/docs/licence/OpenJDK`.

Licenční ujednání použitého serveru Tomcat [1] je k dispozici na adrese <https://www.apache.org/licenses/LICENSE-2.0.html> a je uloženo na médiu ve složce `/docs/licence/Tomcat`. Na toto ujednání na stejné adrese se odkazuje licenční ujednání Frameworku Vaadin, jehož drobnou část používá Spring Framework pro práci se soubory typu Json (patrně se jedná o chybu sestavení knihoven Spring Framework, protože stejný kód je jejich součástí). Stejně licenční ujednání používá nástroj Maven [91]. Licenční ujednání je k dispozici na adrese <https://maven.apache.org/ref/3.0/license.html> a je uloženo na médiu ve složce `/docs/licence/Maven`. Shodně licenční ujednání používá Spring Framework [79]. Licenční ujednání je k dispozici na adrese <https://github.com/spring-projects/spring-framework/blob/master/src/docs/dist/license.txt> a je uloženo na médiu ve složce `/docs/licence/Spring`. Spring Boot a Spring Security používají rovněž stejná licenční ujednání. Jsou k dispozici na adresách <https://github.com/spring-projects/spring-boot/blob/master/LICENSE.txt> a <https://github.com/spring-projects/spring-security/blob/master/LICENSE.txt> a jsou uložena na médiu ve složkách `/docs/licence/SpringBoot` a `/docs/licence/SpringSecurity`.

Licenční ujednání pro použití databáze PostgreSQL [6] a nástroje pgAdmin [92] je k dispozici na adrese <https://www.pgadmin.org/licence/#postgresql> a je uloženo na médiu ve složce `/docs/licence/postgre`.

Licenční ujednání nástroje pro testování Mockito [93] je k dispozici na adrese <https://github.com/mockito/mockito/wiki/License> a je uloženo na médiu ve složce `/docs/licence/Mockito`. Licenční ujednání nástroje pro testování JUnit [94] je k dispozici na adrese <https://junit.org/junit4/license.html> a je uloženo na médiu ve složce `/docs/licence/JUnit`. Licenční ujednání nástroje JaCoCo pro testování pokrytí aplikace je k dispozici na adrese <https://www.jacoco.org/jacoco/trunk/doc/license.html> je uloženo na médiu ve složce `/docs/licence/JaCoCo`.

Licenční ujednání logovací komponenty LogBack [95] je k dispozici na adrese <http://logback.qos.ch/license.html> a je uloženo na médiu ve složce `/docs/licence/LogBack`.

5.4.3 Komponenty uživatelského rozhraní

Licenční ujednání frameworku Bootstrap je k dispozici na adrese <https://github.com/twbs/bootstrap/blob/v4.0.0/LICENSE> a je uloženo na médiu ve složce `/docs/licence/Bootstrap`.

Licenční ujednání jQuery se řídí standardní MIT licencí. Odkaz na licenci MIT je uveden na adrese <https://jquery.org/license/>, samotná licence je k dispozici na adrese <https://tldrlegal.com/license/mit-license>. Dokumenty jsou uloženy na médiu ve složce `/docs/licence/jquery`. Grafická šablona vychází ze šablony Spacelab [81] dostupné na adrese <https://bootswatch.com/spacelab/>. Licence šablony je k dispozici na adrese <https://github.com/thomaspark/bootswatch/blob/master/LICENSE> a je uložena ve složce `/docs/licence/SpaceLab`. Licenční ujednání notifikační komponenty toastr je k dispozici na adrese <https://github.com/CodeSeven/toastr/blob/master/LICENSE> a je uložena na médiu ve složce `/docs/licence/toastr`.

Ze sady Glyphicons [96] byly použity ty ikony, které jsou k dispozici v rámci frameworku Bootstrap 3. Bootstrap 3 dále používá volně dostupný výběr ikon ze sady Font Awesome [97] a volně dostupné fonty Google [98].

Licenční ujednání komponenty pro zobrazení kalendáře Datepicker je k dispozici na adrese <https://github.com/uxsolutions/bootstrap-datepicker/blob/master/LICENSE> a na médiu je uloženo ve složce `/docs/licence/Datepicker`.

Licenční ujednání Goole reCaptcha [99] je k dispozici na adrese <https://github.com/google/recaptcha/blob/master/LICENSE> a na médiu je uložené ve složce `/docs/licence/reCaptcha`.

5.4.4 Implementace nahrávání videa

Kód pro pořízení videonahrávky byl inspirován demonstrační implementací [60]. Kód této demonstrační implementace je publikován na adrese <https://github.com/addpipe/Media-Recorder-API-Demo> [100] a není doprovázen žádným licenčním ujednáním. V odkazovaném článku [55] jsou klíčové části kódu vysvětleny a v diskusi pod článkem autor kódu odpovídá na dotazy a podporuje reimplementaci kódu čtenáři článku. Pro vyloučení případných pochybností jsem požádal autora kódu o souhlas s jeho užitím a tento souhlas jsem získal. Je přiložen k této práci v příloze A.

Výběr vstupních a výstupních zařízení je implementován podle demonstračního kódu na adrese <https://webrtc.github.io/samples/src/content/devices/input-output/> [101], jehož licence typu BSD3 je k dispozici na adrese <https://github.com/webrtc/samples/blob/gh-pages/LICENSE.md> a je uložena na médiu ve složce `/docs/licence/WebRTC`.

5.4.5 Tipy a triky

Při psaní kódu jsem běžně využíval dostupnou dokumentaci pro html kód a javascript [102] a [103], v případě serverové části pak dokumenti Spring frameworku [104]. Četným zdrojem informací byl například také web baeldung.com [105] nebo stackoverflow.com [106] a řada dalších náhodných zdrojů. Tyto dílčí vstupy nejsou v tomto dokumentu citovány, v kódu jsou však uvedeny formou odkazu tak, jak je zachyceno na obrázku 5.22.

Google reCaptcha byla implementována podle dokumentace, která je k dispozici na stránkách společnosti Google [107].

```
// reference 6.4.2021 https://stackoverflow.com/questions/54189435/how-to-disable-context-menu-for-a-specific-element
document.getElementById("recordingControl$id")
    .addEventListener("contextmenu", function(e) {e.preventDefault();});

// reference 28.3.2021 https://stackoverflow.com/questions/43142703/get-a-reference-to-currently-active-datasource-in-spring-boot
@Autowired
private DataSource dataSource;
```

■ **Obrázek 5.22** Odkazy na zdroje v kódu aplikace

5.4.6 Nepoužité komponenty

Práce nevyužívá knihovnu `adapter.js` [65], ale její použití v budoucnosti je pravděpodobné. Knihovna `adapter.js` je k dispozici pod licencí typu BSD [67].



Kapitola 6

Závěr

Cílem práce bylo navrhnout a implementovat databázi a webovou aplikaci/rozhraní pro telerehabilitační systém s backendem implementovaným v Javě. Zadavatel dále požadoval, aby přes rozhraní bylo možné nahrát objemná data (například video) a následně vyčkat na výsledky vypočtené výpočetním serverem. V rámci práce jsem měl prozkoumat současné softwarové technologie pro danou úlohu, navrhnout vhodný databázový model pro danou problematiku, navrhnout webový server, backend a frontend webové aplikace. Navrženou databázi a aplikaci jsem měl implementovat a ověřit její funkčnost. Zadavatel také požadoval vytvořit testy ověřující funkčnost webové aplikace, zdokumentovat implementovanou databázi a webovou aplikaci a vytvořit manuál k obsluze webové aplikace.

Aplikace, která je věcným výstupem bakalářské práce, je v době dokončení bakalářské práce funkční a je připravena na předprodukční testování nebo pilotní provoz pro omezený okruh uživatelů – aplikační požadavky zadavatele jsem implementoval a aplikaci je možné použít. Nejsou vedeny žádné známé chyby a většinu drobných nedostatků, které jsou příležitostně nalezeny, je možné odstranit během několika minut.

Databázi a webovou aplikaci/rozhraní pro telerehabilitační systém jsem navrhl a implementoval.

Backend webového rozhraní jsem implementoval v Javě. Přes rozhraní je možné nahrát objemná data (například video) a je možné vyčkat na výsledky vypočtené výpočetním serverem. V rámci práce jsem prozkoumal současné softwarové technologie pro danou úlohu. Navrhl jsem vhodný databázový model pro danou problematiku. Navrhl jsem webový server, backend a frontend webové aplikace. Navrženou databázi i aplikaci jsem implementoval a ověřil jsem jejich funkčnost. Vytvořil jsem testy ověřující funkčnost webové aplikace. Implementovanou databázi a aplikaci jsem zdokumentoval a vytvořil jsem manuál k obsluze webové aplikace.

Vytčeného cíle práce jsem dosáhl a zadání bakalářské práce jsem splnil.

Literatura

- [1] THE APACHE SOFTWARE FOUNDATION. *Apache Tomcat* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://tomcat.apache.org/>
- [2] VMWARE, INC. OR ITS AFFILIATES. *Spring Boot* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://spring.io/projects/spring-boot>
- [3] ORACLE. *GlassFish* [software]. 2019. [přístup 4. 5. 2021]. Dostupné z <https://javaee.github.io/glassfish/>
- [4] Red Hat. *WildFly* [software]. 2021. [přístup 4. 5. 2021]. Dostupné z <https://www.wildfly.org/>
- [5] Red Hat. *Quarkus* [software]. 2021. [přístup 4. 5. 2021]. Dostupné z <https://quarkus.io/>
- [6] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.postgresql.org/>
- [7] ORACLE. *MySQL* [software]. 2021. [přístup 5. 5. 2021]. Dostupné z <https://www.mysql.com/>
- [8] ANGULAR TEAM AT GOOGLE. *Angular* [software]. 2021. Dostupné z <https://angular.io/>
- [9] ANGULAR TEAM AT GOOGLE. Building dynamic forms. In: *Angular* [online]. Google, 2021. [cit. 13. 4. 2021]. Dostupné z <https://angular.io/guide/dynamic-form>
- [10] ANGULAR TEAM AT GOOGLE. TypeScript configuration. *Angular* [online]. Google, 2021. [cit. 13. 4. 2021]. Dostupné z <https://angular.io/guide/typescript-configuration>
- [11] MICROSOFT. *TypeScript* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.typescriptlang.org/>
- [12] FARID, Danial. Split a big file to be upload using angular. In: *TopSites* [online] TheTopSites.net, 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.thetopsites.net/article/58238261.shtml>
- [13] ANSARI, Mohd Faisal. angular-file-uploader. In: *Build Amazing Things* [online] npm. Inc., 2020. [cit. 13. 4. 2021]. Dostupné z <https://www.npmjs.com/package/angular-file-uploader>

- [14] BUSH, Dave. Upload an In Memory Image as a File using Angular. In: *Dave Bush* [online] Dave Bush, 2018. [cit. 13. 4. 2021]. Dostupné z <https://davembush.medium.com/upload-an-in-memory-image-as-a-file-using-angular-7f1b4c1152bf>
- [15] SCHLOTE, Sebastian. *ngx-webcam* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.npmjs.com/package/ngx-webcam>
- [16] MOZILLA AND INDIVIDUAL CONTRIBUTORS. Media Capture and Streams API (Media Stream). In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/Media_Streams_API
- [17] KHAN, Muaz. *RecordRTC.js* [software]. 2019. [přístup 13. 4. 2021]. Dostupné z <https://recordrtc.org/>
- [18] DELGADO, Carlos. How to record a video with audio in the browser with JavaScript (WebRTC). In: *Our Code World* [online]. Carlos Delgado, 2018. [cit. 13. 4. 2021]. Dostupné z <https://ourcodeworld.com/articles/read/671/how-to-record-a-video-with-audio-in-the-browser-with-javascript-webrtc>
- [19] GUTTANDIN, Chris. *extendable-media-recorder* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.npmjs.com/package/extendable-media-recorder>
- [20] ORACLE AND/OR ITS AFFILIATES. What Is a JSP Page? In: *Opensource.com* [online]. Oracle and/or its affiliates, 2010. [cit. 13. 4. 2021]. Dostupné z <https://docs.oracle.com/javase/5/tutorial/doc/bnagy.html>
- [21] MUKHERJEE, Nimisha. How JavaScript became a serious programming language. In: *Opensource.com* [online]. Red Hat, Inc., 2020. [cit. 19. 4. 2021]. Dostupné z <https://opensource.com/article/20/10/history-javascript>
- [22] PETERKA, Jiří. Linková vrstva - II. In: *Archiv článků a přednášek Jiřího Peterky* [online]. Jiří Peterka, 2015. [cit. 12. 4. 2021]. Dostupné z <https://www.earchiv.cz/a92/a219c110.php3>
- [23] PETERKA, Jiří. Protokol TCP – III. In: *Archiv článků a přednášek Jiřího Peterky* [online]. Jiří Peterka, 2015. [cit. 12. 4. 2021]. Dostupné z <https://www.earchiv.cz/a93/a309c110.php3>
- [24] PETERKA, Jiří. Referenční model ISO/OSI – sedm vrstev. In: *Archiv článků a přednášek Jiřího Peterky* [online]. Jiří Peterka, 2015. [cit. 12. 4. 2021]. Dostupné z <https://www.earchiv.cz/a92/a213c110.php3>
- [25] FIEDLER, Glenn. Why can't I send UDP packets from a browser? In: *Gaffer On Games* [online]. Glenn Fiedler, 2017. [cit. 12. 4. 2021]. Dostupné z https://gafferongames.com/post/why_cant_i_send_udp_packets_from_a_browser/
- [26] ALZA.CZ A.S. Zyxel VMG3312-T20A. In: *Alza.cz* [online]. Alza.cz a.s. [cit. 12. 4. 2021]. Dostupné z <https://www.alza.cz/zyxel-vmg3312-t20a-d5484622.htm>
- [27] O2 CZECH REPUBLIC A.S. Supervýkonný modem O2 Smart Box. In: *O2.cz* [online]. O2 Czech Republic a.s., 2021. [cit. 12. 4. 2021]. Dostupné z <https://www.o2.cz/osobni/smart-box>
- [28] OOKLA, LLC. *SPEEDTEST* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://www.speedtest.net/>

- [29] PETERKA, Jiří. Protokol TCP – II. In: *Archiv článků a přednášek Jiřího Peterky* [online]. Jiří Peterka, 2015. [cit. 12. 4. 2021]. Dostupné z <https://www.earchiv.cz/a93/a307c110.php3>
- [30] MOZILLA AND INDIVIDUAL CONTRIBUTORS. HTTP request methods. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>
- [31] MOZILLA AND INDIVIDUAL CONTRIBUTORS. HTTP Messages. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages>
- [32] MOZILLA AND INDIVIDUAL CONTRIBUTORS. The WebSocket API (WebSockets). In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
- [33] INTERNET ENGINEERING TASK FORCE (IETF). The WebSocket Protocol. In: *IETF Tools* [online]. IETF, 2011. [cit. 12. 4. 2021]. Dostupné z <https://tools.ietf.org/html/rfc6455.html>
- [34] AUGUSTIN, Aymeric and contributors. Design. In: *websockets* [online]. Aymeric Augustin and contributors, 2019. [cit. 13. 4. 2021]. Dostupné z <https://websockets.readthedocs.io/en/stable/design.html>
- [35] MOZILLA AND INDIVIDUAL CONTRIBUTORS. FormData. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developer.mozilla.org/en-US/docs/Web/API/FormData>
- [36] MOZILLA AND INDIVIDUAL CONTRIBUTORS. WebSocket.bufferedAmount. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developer.mozilla.org/en-US/docs/Web/API/WebSocket/bufferedAmount>
- [37] BEZKODER.COM. Spring Boot File upload example with Multipart File. In: *{z}Koder* [online]. bezkoder.com, 2021. [cit. 13. 4. 2021]. Dostupné z <https://bezkoder.com/spring-boot-file-upload/>
- [38] AURA. *Hotspot Shield* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://www.hotspotshield.com/>
- [39] GEOIP.COM. *GEOIP.COM* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://geoip.com/>
- [40] THE WORLD WIDE WEB CONSORTIUM (W3C). HTML Media Capture. In: *W3C* [online]. The World Wide Web Consortium (W3C), 2021. [cit. 13. 4. 2021]. Dostupné z <https://w3c.github.io/html-media-capture/>
- [41] NAICU, Octavian. Correct Syntax for HTML Media Capture. In: *Deconstruct* [online]. Pipe Services S.R.L., 2016. [cit. 13. 4. 2021]. Dostupné z <https://blog.addpipe.com/correct-syntax-html-media-capture/>
- [42] NAICU, Octavian (Pipe Recording Platform). HTML Media Capture Examples. In: *AddPipe.com* [online]. Pipe Services S.R.L., 2020. [cit. 13. 4. 2021]. Dostupné z <https://addpipe.com/html-media-capture-demo/>
- [43] NAICU, Octavian. Why we chose WebRTC over Media Recorder API for HTML5 Video Recording. In: *Deconstruct* [online]. Pipe Services S.R.L., 2017. [cit. 13. 4. 2021]. Dostupné z <https://blog.addpipe.com/html5-video-recording-webrtc-mediarecorder-api/>

- [44] MOZILLA AND INDIVIDUAL CONTRIBUTORS. WebRTC API. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API
- [45] DUTTON, Sam. Get Started with WebRTC. In: *Tutorials* [online]. Google, 2012. [cit. 13. 4. 2021]. Dostupné z <https://www.html5rocks.com/en/tutorials/webrtc/basics/>
- [46] MOZILLA AND INDIVIDUAL CONTRIBUTORS. Media Capture and Streams API (Media Stream). In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/Media_Streams_API
- [47] MOZILLA AND INDIVIDUAL CONTRIBUTORS. Signaling and video calling. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Signaling_and_video_calling
- [48] BALKO, Sören. Why WebRTC-based webcam recorders are a terrible idea. In: *Clipchamp* [online]. Clipchamp, 2017. [cit. 13. 4. 2021]. Dostupné z <https://clipchamp.com/en/blog/webrtc-webcam-recorders-are-terrible/>
- [49] BRIGHTCOVE, INC. *Video JS* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://videojs.com/>
- [50] TRIEMSTRA, Thijs. *videojs-record* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://github.com/collab-project/videojs-record>
- [51] KHAN, Muaz, Audio+Video+Screen Recording using RecordRTC. In: *WebRTC Demos, Experiments, Libraries, Examples* [online]. Muaz Khan, 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.webrtc-experiment.com/RecordRTC/>
- [52] MOZILLA AND INDIVIDUAL CONTRIBUTORS. MediaStream Recording API. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/MediaStream_Recording_API
- [53] THE WORLD WIDE WEB CONSORTIUM (W3C). MediaStream Recording. In: *W3C* [online]. The World Wide Web Consortium (W3C), 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.w3.org/TR/mediastream-recording/>
- [54] MIDO22 at stackoverflow.com. HTML 5 video recording and storing a stream. In: *Stack Overflow* [online]. Stack Exchange Inc., 2021. [cit. 13. 4. 2021]. Dostupné z <https://stackoverflow.com/questions/18509385/html-5-video-recording-and-storing-a-stream>, <https://stackoverflow.com/a/34259326>
- [55] NEGROTA, Remus. HTML5's Media Recorder API in Action on Chrome and Firefox. In: *Deconstruct* [online]. Pipe Services S.R.L., 2015. [cit. 13. 4. 2021]. Dostupné z <https://blog.addpipe.com/mediarecorder-api/>
- [56] DUTTON, Sam. Record Audio and Video with MediaRecorder. In: *Google Developers* [online]. Google, 2020. [cit. 13. 4. 2021]. Dostupné z <https://developers.google.com/web/updates/2016/01/mediarecorder>
- [57] THE WEBRTC PROJECT AUTHORS, THE CHROMIUM AUTHORS. WebRTC samples. In: *GitHub Pages* [online]. GitHub, Inc., 2018. [cit. 13. 4. 2021]. Dostupné z <https://webrtc.github.io/samples/>
- [58] THE WEBRTC PROJECT AUTHORS, THE CHROMIUM AUTHORS. WebRTC samples MediaRecorder. In: *GitHub Pages* [online]. GitHub, Inc., 2018. [cit. 13. 4. 2021]. Dostupné z <https://webrtc.github.io/samples/src/content/getusermedia/record/>

- [59] QUICKBLOX. Media Stream Recorder by Quickblox. In: *javascript-media-recorder* [online]. GitHub, Inc., 2018. [cit. 13. 4. 2021]. Dostupné z <https://quickblox.github.io/javascript-media-recorder/sample/>
- [60] NAICU Octavian, Remus NEGROTA (Pipe Recording Platform). Media Recorder API Demo. In: *AddPipe.com* [online]. Pipe Services S.R.L., 2015. [cit. 13. 4. 2021]. Dostupné z <https://addpipe.com/media-recorder-api-demo/>
- [61] THE WORLD WIDE WEB CONSORTIUM (W3C). Roadmap of Web Applications on Mobile. In: *W3C* [online]. The World Wide Web Consortium (W3C), 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.w3.org/Mobile/roadmap/media.html>
- [62] THE WORLD WIDE WEB CONSORTIUM (W3C). W3C Process Document. In: *W3C* [online]. The World Wide Web Consortium (W3C), 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.w3.org/2004/02/Process-20040205/tr.html>
- [63] NAICU, Octavian. MediaStream Recorder API Now Available in Safari Technology Preview 73. In: *Deconstruct* [online]. Pipe Services S.R.L., 2015. [cit. 13. 4. 2021]. Dostupné z <https://blog.addpipe.com/safari-technology-preview-73-adds-limited-mediastream-recorder-api-support/>
- [64] PIPE SERVICES S.R.L. Simple, transparent pricing. In: *AddPipe.com* [online]. Pipe Services S.R.L., 2015. [cit. 13. 4. 2021]. Dostupné z <https://addpipe.com/pricing>
- [65] MOZILLA AND INDIVIDUAL CONTRIBUTORS. Improving compatibility using WebRTC adapter.js. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/adapter.js
- [66] THE WEBRTC PROJECT AUTHORS, THE ADAPTER.JS PROJECT AUTHORS. *adapter.js* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://github.com/webrtc/adapter/tree/master/release>
- [67] THE WEBRTC PROJECT AUTHORS, THE ADAPTER.JS PROJECT AUTHORS. LICENSE In: *GitHub Pages* [online]. GitHub, Inc., 2018. [cit. 13. 4. 2021]. Dostupné z <https://github.com/webrtc/adapter/blob/master/LICENSE.md>
- [68] VISICOM MEDIA INC. *ManyCam* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://manycam.com/>
- [69] LOGITECH. *Logitech Capture* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://www.logitech.com/en-hk/product/capture>
- [70] GITLAB. *GitLab* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://about.gitlab.com/>
- [71] SONARSOURCE S.A, SWITZERLAND. *SonarQube* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://www.sonarqube.org/>
- [72] VMWARE, INC. OR ITS AFFILIATES. *Spring Security* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://spring.io/projects/spring-security>
- [73] BOOTSTRAP TEAM. *Bootstrap* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://getbootstrap.com/>
- [74] OPENJS FOUNDATION AND JQUERY CONTRIBUTORS. *jQuery* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://jquery.com/>

- [75] FJÄLLEMARK, Hans, John PAPA & Tim FERRELL. *Toastr* [software]. 2014. [přístup 13. 4. 2021]. Dostupné z <https://codeseven.github.io/toastr/>
- [76] MOZILLA AND INDIVIDUAL CONTRIBUTORS. *async function*. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function
- [77] MOZILLA AND INDIVIDUAL CONTRIBUTORS. *Graceful asynchronous programming with Promises*. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Promises>
- [78] MOZILLA AND INDIVIDUAL CONTRIBUTORS. *Atoms*. In: *MDN Web Docs* [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Atoms
- [79] VMWARE, INC. OR ITS AFFILIATES. *Spring* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://spring.io>
- [80] PETRE, Stefan. *Datepicker for Bootstrap v1.6.4* [software]. 2012. [přístup 13. 4. 2021]. Dostupné z <https://github.com/eternicode/bootstrap-datepicker>
- [81] PARK, Thomas. *Spacelab* [software]. 2012. [přístup 16. 4. 2021]. Dostupné z <https://bootswatch.com/spacelab/>
- [82] KUSKIN, Artur. *Handling Passwords with Spring Boot and Spring Security*. In: *Reflecting = How + Why* [online]. Tom Hombergs, 2021. [cit. 13. 4. 2021]. Dostupné z <https://reflecting.io/spring-security-password-handling/>
- [83] DOCKER, INC. *Docker Desktop* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.docker.com/products/docker-desktop>
- [84] BALSAMIQ STUDIOS, LLC. *Balsamiq Mockups 3* [software]. 2017. [přístup 13. 4. 2021]. Nástupnický software se stejnými funkcemi, ale jinou podkladovou technologií dostupný z <https://balsamiq.com/wireframes/mockups3fordesktop/>
- [85] SPARX SYSTEMS PTY LTD. *Enterprise Architect 15.2 Corporate Edition* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.sparxsystems.com/>
- [86] TRACKER SOFTWARE PRODUCTS (Canada) Ltd. *PDF-XChange Editor 9.0* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://www.tracker-software.com/product/pdf-xchange-editor>
- [87] JASC SOFTWARE, INC. *Jasc Paint Shop Pro* [software]. 2000.
- [88] JETBRAINS S. R. O. *IntelliJ IDEA 2020.3 (Ultimate Edition)* [software]. [přístup 12. 4. 2021]. Dostupné z <https://www.jetbrains.com/idea/>
- [89] ORACLE CORPORATION AND/OR ITS AFFILIATES. *OpenJDK* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://openjdk.java.net/>
- [90] ORACLE. *Java SE Development Kit 11*. [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>
- [91] THE APACHE SOFTWARE FOUNDATION. *Maven* [software]. 2021. [přístup 12. 4. 2021]. Dostupné z <https://maven.apache.org/>
- [92] THE PGADMIN DEVELOPMENT TEAM. *pgAdmin* [software]. 2021. [přístup 13. 4. 2021]. Dostupné z <https://www.pgadmin.org/>

- [93] SZCZEPAN FABER AND FRIENDS. *Mockito* [software]. 2021.
- [94] THE JUNIT TEAM. *JUnit* [software]. 2021.
- [95] QUALITY OPEN SOFTWARE (QOS.CH). *LogBack* [software]. 2021.
- [96] GLYPHICONS. *Glyphicons* [software]. 2021. [přístup 16. 4. 2021]. Dostupné z <https://www.glyphicons.com/>
- [97] FONTICONS, INC. *Font Awesome* [software]. 2021. [přístup 16. 4. 2021]. Dostupné z <https://fonts.google.com/>
- [98] GOOGLE. *Google Fonts* [software]. 2021. [přístup 16. 4. 2021]. Dostupné z <https://fontawesome.com/>
- [99] GOOGLE. *Google reCaptcha v2* [software]. 2014. [přístup 12. 4. 2021]. Dostupné z <https://www.google.com/recaptcha/about/>
- [100] NAICU, Octavian, Remus NEGROTTA. *Media Recorder API Demo* [software]. GitHub, Inc., 2021. [cit. 20. 4. 2021]. Dostupné z <https://github.com/addpipe/Media-Recorder-API-Demo>
- [101] THE WEBRTC PROJECT AUTHORS. Select sources & outputs. In: *WebRTC samples* [software]. GitHub, Inc., 2021. [cit. 20. 4. 2021]. Dostupné z <https://webrtc.github.io/samples/src/content/devices/input-output/>
- [102] MOZILLA AND INDIVIDUAL CONTRIBUTORS. *Resources for developers, by developers*. [online]. Mozilla, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developer.mozilla.org/en-US/>
- [103] REFSNES DATA. *W3SCHOOLS* [online]. Refsnes Data, 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.w3schools.com/>
- [104] VMWARE, INC. OR ITS AFFILIATES. *Guides* [online]. VMware, Inc. or its affiliates, 2021. [cit. 13. 4. 2021]. Dostupné z <https://spring.io/guides>
- [105] TARNUM JAVA SRL. *Baeldung* [online] Tarnum Java SRL, 2021. [cit. 13. 4. 2021]. Dostupné z <https://www.baeldung.com/>.
- [106] STACK EXCHANGE INC. *Stack Overflow.*, Stack Exchange Inc., 2021. [cit. 13. 4. 2021]. Dostupné z <https://stackoverflow.com/>.
- [107] GOOGLE. Developer's Guide. In: *Google Developers* [online]. Google, 2021. [cit. 12. 4. 2021]. Dostupné z <https://developers.google.com/recaptcha/intro>

..... Kapitola A

Souhlas s užitím kódu

petrzja5@cvut.cz

From: Octavian Naicu <octavian@addpipe.com>
Sent: úterý 20. dubna 2021 17:20
To: Petrzilka, Jan
Subject: Re: Media Recorder API Demo licence conditions

Hi Jan,

Feel free to use the published code in your thesis.

I am glad it helped you a lot.

On Tue, Apr 20, 2021 at 11:32 AM Petrzilka, Jan <petrzja5@fit.cvut.cz> wrote:

Dear Mr Naicu,

I am a bachelor student of information science at Czech Technical University in Prague. I would like to thank you a lot for your article, demo and source code you made available to the public at the below listed addresses:

<https://blog.addpipe.com/mediarecorder-api/>

<https://addpipe.com/media-recorder-api-demo/>

<https://github.com/addpipe/Media-Recorder-API-Demo>

I am finalizing my bachelor thesis these days and I am running compliance checks on it. I found the above listed materials extremely useful, they helped me a lot in understanding how the Media Recorder API works and - because of this - a few of lines of the code in my thesis is inspired by the code you published and, of course, they look very similar. This is all stated in my thesis with references to the sources listed above. I assumed that you published the code so that everyone can look at it and understand how it and the API works together, but I realized today that there is no licence agreement published along with the code and that it is not clear that I am allowed to use the demo and the code you published that way.

Therefore, I am contacting you right now, admitting that I looked at the code you published and that it helped me a lot. I need to ask you for a clarification of the conditions and your permission to study the code you published and get inspired by it in my own bachelor thesis.

If you need to understand what my thesis is about, I attach its specification proposal to this e-mail. Even though it is in Czech language, I believe that you can get an idea what the software is about as the abstract and the screenshots are in English.

Many thanks.

Kind regards,

Jan Petrzilka

--

Best Regards,

--

Octavian Naicu,
Founder and CTO @ Pipe

Kapitola B

Uživatelská příručka

Uživatelská příručka popisuje webovou aplikaci TeleCam.cz z uživatelského hlediska. Uživatelé aplikace jsou v aplikaci registrováni s jednou z následujících rolí:

- Klient
- Terapeut
- Správce

Tyto role odpovídají jednotlivým typům reálných uživatelů – správci aplikace používají uživatelský přístup s rolí správce, terapeuti uživatelské přístupy s rolí terapeut a klienti uživatelské přístupy s rolí klient.

Uživatelé s rolí správce nastavují konfiguraci aplikace a mohou registrovat další uživatele v roli terapeut nebo klient. Uživatelé s rolí terapeut mají přiřazené klienty a mohou přistupovat k jejich úlohám. Mohou také zakládat nové klienty. Uživatelé s rolí klient mají přiřazení právě jednoho terapeuta, mohou vyplňovat úlohy a zobrazovat jejich vyhodnocení.

Uživatelé s rolí správce se mohou přihlásit jako některý z uživatelů s rolí terapeut nebo klient. Uživatelé s rolí terapeut se mohou přihlásit jako některý z jím přiřazených klientů. Tato funkčnost je užitečná, pokud terapeut potřebuje vidět aplikaci z pohledu klienta a například klientovi ukázat, jak se aplikace ovládá; obdobně to platí pro správce aplikace. Uživatelé nabývají stavů podle tabulky B.1.

■ Tabulka B.1 Stavů uživatelů

Stav	Význam
Aktivní	Uživatel se může přihlásit. Pokud se jedná o klienta, může odeslat úlohu.
Inaktivní	Uživatel se může přihlásit, ale nemůže odeslat úlohu.
Blokovaný	Uživatel se nemůže přihlásit a nemůže odeslat úlohu.

Základní funkčnost aplikace lze shrnout do následujících bodů:

- Správce definuje šablonu úlohy
- Klient provede úlohu tak, že v aplikaci vyplní šablonu
- Aplikace postoupí vyplněnou šablonu s přílohami další aplikaci k vyhodnocení a získá od ní výsledky
- Aplikace zobrazí klientovi výsledky vyhodnocení úlohy

- Terapeut má přístup k úlohám jím přiřazených klientů

Podporovanými prohlížeči jsou prohlížeče Chrome a Firefox. Na zařízení iPad 2018 s operačním systémem iOS byla aplikace testována také s prohlížečem Safari – pro správnou funkci tohoto prohlížeče je potřeba v nastavení prohlížeče zapnout experimentální funkčnost „Media recorder“.

Uživatelská příručka je členěna do částí podle stránek pro uživatele s jednotlivými rolemi.

B.1 Společné stránky

B.1.1 Úvodní stránka

Po spuštění aplikace je uživateli zobrazena úvodní stránka na obrázku B.1. Tato stránka je, jako všechny stránky aplikace, připravena v českém a anglickém jazyce. Přepínání mezi jazyky se provádí kliknutím na příslušné tlačítko v pravém horním rohu stránky. Při změně jazyka je aktuální stránka načtena znovu.

Česky

English

TeleCam

Vítejte v aplikaci TeleCam

Aplikace je přístupná pouze oprávněným uživatelům. Před pokračováním se, prosím, ujistěte, že jste oprávněným uživatelem.

 Pokračovat

© 2021

Jan Petržílka, jan.petrzilka.cz

Fakulta informačních technologií, České vysoké učení technické v Praze, fit.cvut.cz

Fakulta biomedicínského inženýrství, České vysoké učení technické v Praze, fbmi.cvut.cz

- **Obrázek B.1** Úvodní stránka aplikace

Stránka na obrázku B.1 je kopií webové stránky v prohlížeči. Grafická podoba dalších stránek z aplikace je v příručce tam, kde je to možné, upravena tak, aby byla zlepšena jejich čitelnost; často se také jedná jen o výsek většího celku webové stránky. Z úvodní stránky uživatel může pokračovat na přihlašovací stránku.

B.1.2 Přihlašovací stránka

Přihlašovací stránka na obrázku B.2 je uživateli zobrazena vždy, pokud přistupuje na stránku, která vyžaduje přihlášení uživatele, a není dosud do aplikace přihlášen. U stránek, kde přihlášení není požadováno, je tato skutečnost v příručce uvedena.

Přihlaste se

E-mail: jmeno.prijmeni@firma.cz

Heslo:

Trvalé přihlášení

■ **Obrázek B.2** Přihlašovací stránka

Na přihlašovací stránce uživatel vkládá svoje přihlašovací údaje – e-mailovou adresu a heslo. Aplikace jej může vyzvat k provedení Captcha testu. Po úspěšném přihlášení pokračuje na úvodní stránku podle své uživatelské role. Alternativně může přejít na stránku pro obnovení hesla.

Pokud uživatel využije možnosti trvalého přihlášení, zůstává v prohlížeči přihlášen nejdéle po dobu 14 dní. Přihlášení je ukončeno, pokud se uživatel v daném prohlížeči odhlásí nebo pokud si změni heslo.

B.1.3 Stránka pro obnovení hesla

Na stránce pro obnovení hesla na obrázku B.3 uživatel vyplní svoji e-mailovou adresu a provede Captcha test. Pokud uživatel existuje, pak aplikace na jeho e-mailovou adresu odešle link pro nastavení nového hesla. Uživatel z chování aplikace nepozná, jestli v aplikaci registrovaný uživatel s uvedenou e-mailovou adresou existuje.

Obnovení hesla

E-mail: jmeno.prijmeni@firma.cz

Nejsem robot reCAPTCHA Ochrana soukromí - Smluvní podmínky

■ **Obrázek B.3** Obnovení hesla

Na adresu uživatele je odeslán e-mail na obrázku B.4. Po kliknutí na link uživatel nastaví svoje nové heslo.

Jan Petržílka

From: web@telecam.cz
Sent: středa 31. března 2021 22:54
To: th@petrzilka.cz
Subject: Link na obnovení hesla / Password reset link

Dobrý den,

kliknutím na link níže můžete obnovit svoje heslo na webu telecam.cz.

<https://telecam.cz/setpassword?u=746840706574727a696c6b612e637a&t=37372b3937372b39612b2b2f765550767637314d37372b395251>

Přeji hezký den,

server <https://telecam.cz>

■ **Obrázek B.4** E-mail s linkem na obnovení hesla

B.1.4 Stránka pro nastavení hesla

Po založení nového uživatele nebo při obnovení hesla pošle aplikace na e-mailovou adresu uživatele link na nastavení hesla obdobně, jako na obrázku B.4. Po jeho otevření aplikace zobrazí stránku pro nastavení nového hesla na obrázku B.5. Aplikace vyžaduje, aby nové heslo obsahovalo alespoň jedno velké písmeno, jedno malé písmeno, jednu číslici nebo jeden speciální znak. Minimální délka hesla je 8 znaků. Aplikace uživateli při vkládání hesla poskytuje dynamickou nápovědu o tom, zda vkládané heslo odpovídá požadavkům.

Nové heslo

Heslo: 

Napište heslo znovu: 

■ **Obrázek B.5** Nastavení hesla

Po nastavení nového hesla aplikace uživateli zobrazí úvodní stránku podle jeho uživatelské role.

B.1.5 Stránka pro změnu hesla

Přihlášený uživatel může otevřít stránku pro změnu hesla na obrázku B.6 a na této stránce heslo změnit. Pro změnu hesla je potřeba vložit heslo původní a dvakrát heslo nové. Požadavky na nové heslo jsou uvedeny v odstavci B.1.4. Aplikace uživateli při vkládání hesla poskytuje dynamickou nápovědu o tom, zda vkládané heslo odpovídá požadavkům.

Změna hesla

Stávající heslo:

Nové heslo:

Napište nové heslo znovu:

Nastavit heslo

■ Obrázek B.6 Změna hesla

B.2 Stránky uživatele s rolí klient

Uživatel s rolí klient má k dispozici následující funkčnosti přístupné prostřednictvím jednoúrovňového menu:

- „Údaje“ – otevře stránku pro editaci údajů
- „Nová úloha“ – umožní provést novou úlohu
- „Úlohy“ – zobrazí provedené úlohy klienta
- „Změna hesla“
- „Odhlásit“

B.2.1 Stránka „Údaje“

Na stránce „Údaje“ aplikace klientovi zobrazí na stránce na obrázku B.7 jeho základní osobní údaje, přiřazeného terapeuta a další uložené osobní údaje podle aktuální šablony pro osobní údaje používané pro úlohy. Údaje podle šablony pro osobní údaje může uživatel změnit a uložit.

Editace údajů

E-mailová adresa (*):	<input type="text" value="cl@petrzilka.cz"/>
Titul před jménem:	<input type="text"/>
Jméno (*):	<input type="text" value="Jan"/>
Příjmení (*):	<input type="text" value="Klient"/>
Titul za jménem:	<input type="text"/>
Terapeut (*):	<input type="text" value="Jan Terapeut (th@petrzilka.cz)"/>
Datum narození (*):	<input type="text" value="28.03.2021"/>
Váha [kg] (*):	<input type="text" value="105"/>
Výška [cm] (*):	<input type="text" value="180"/>
<input type="button" value="✓ Uložit"/>	

■ **Obrázek B.7** Editace osobních údajů

B.2.2 Stránka „Nová úloha“

Aplikace nejprve klientovi nabídne na stránce na obrázku B.8 výběr ze seznamu aktivních šablon pro úlohy.

Nová úloha

Seznam úloh:	<input type="text" value="Cvičení - paže - plný rozsah (id 85)"/>
<input type="button" value="✓ Pokračovat"/>	

■ **Obrázek B.8** Výběr typu úlohy

Po výběru typu úlohy aplikace připraví na stránce na obrázku B.9 formulář pro vyplnění úlohy a zobrazí jej klientovi.

Nová úloha Cvičení - paže - plný rozsah

Osobní údaje

Datum narození (*) :	<input type="text" value="28.03.2021"/>	
Váha [kg] (*) :	<input type="text" value="30 - 250"/>	
Výška [cm] (*) :	<input type="text" value="100"/>	

Údaje úlohy

Výška židle (*) :	<input type="text" value="100 - 140"/>
Záznam cvičení (*) :	<input type="text" value="Přiložte soubor s videem (webm, avi, mp4, mov)"/>
	<input type="button" value="📷 Nahrát video"/> <input type="button" value="📁 Vybrat soubor"/>

Odeslat:	<input type="button" value="✉ Odeslat"/>
-----------------	--

■ Obrázek B.9 Formulář úlohy

Formulář může obsahovat pole vyžadující pořízení videonahrávky. V takovém případě aplikace otevře okno zachycené na obrázku B.10.

Klient formulář vyplní (povinná pole jsou označena hvězdičkou) a odešle na server. Úloha odeslaná na server má stav „Čeká na zpracování“.

Nahrávání videa

Pořidte nahrávku a okno zavřete.

Kamera:	Logitech Webcam C925e (046d:085b) ▼
Mikrofon:	Headset Microphone (Jabra Link 370) (0b0e:245e) ▼
Formát:	1280 x 1024 / 30 fps (video/webm;codecs=h264)
Velikost souboru:	11.019 MB, 11019287 B
Délka souboru:	59 s
Reproduktor:	Default - Headset Earphone (Jabra Link 370) (0b0e:245e) ▼
Ztlumit:	<input checked="" type="checkbox"/>





■ **Obrázek B.10** Pořízení videonahrávky, v obdélníku je obraz z kamery

B.2.3 Stránka „Úlohy“

Tato stránka na obrázku B.11 je pro uživatele s rolí klient výchozí stránkou po přihlášení do aplikace. Na stránce „Úlohy“ aplikace zobrazí přehled všech úloh přihlášeného klienta. Zobrazení podporuje stránkování a filtrování. Data je možné filtrovat podle kategorií uvedených v tabulce B.2.

Přehled úloh

Od data: 
Do data: 
Úloha:
Stav úlohy:

Terapeut:

ID	Datum Stav	Název úlohy	Klient E-mail klienta	Terapeut E-mail terapeuta	Operace
212	31.03.2021 Zpracovaná	Cvičení - paže - plný rozsah	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="☰ Otevřít"/>
211	31.03.2021 Čeká na zpracování	ALL	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="☰ Otevřít"/>

■ **Obrázek B.11** Přehled úloh klienta

■ **Tabulka B.2** Filtrování úloh

Kategorie	Význam
Od data	Nejnižší datum porízení úlohy
Do data	Nejvyšší datum porízení úlohy
Úloha	Typ úlohy
Stav úlohy	Stav úlohy
Terapeut	Terapeut přiřazený ke klientovi v době provedení úlohy

Klient může vypracovanou úlohu zobrazit. Pokud k úloze již existuje vyhodnocení, je úloha ve stavu „Zpracovaná“ a klientovi je toto vyhodnocení zobrazeno na obrázku B.12 spolu s provedenou úlohou. O dostupnosti vyhodnocení je klient informován e-mailem na obrázku B.13. Po kliknutí na link je klientovi zobrazena stránka na obrázku B.12 bez nutnosti přihlášení do aplikace.

Zobrazení úlohy Cvičení - paže - plný rozsah

Klient: Josef Klient (test03@telecam.cz)
Terapeut: František Terapeut, DIS. (test02@telecam.cz)
Datum zpracování: 31.03.2021

Osobní údaje

Datum narození (*): 29.02.2016 




Váha [kg] (*): 84

Výška [cm] (*): 180

Údaje úlohy

Výška židle (*): 120

Záznam cvičení (*): video.webm

 Přehrát  Otevřít  Stáhnout

Výsledky zpracování

Popis cvičení: Cvičil jste vesměs **správně**

Správná pozice:



rightposition.jpg

 Otevřít  Stáhnout

Zpracované video: 144.data

 Přehrát  Otevřít  Stáhnout

Operace:  Upravit obsah

■ **Obrázek B.12** Vyhodnocení úlohy klienta

Jan Petržílka

From: web@telecam.cz
Sent: neděle 28. března 2021 17:17
To: cl@petrzilka.cz
Subject: Link na výsledky úlohy / Task result link

Dobrý den,

kliknutím na link níže můžete zobrazit výsledky úlohy 205.

<https://telecam.cz/viewlinkedresponse/37362d57775262625831526b5275753243636c35452f4765613169726f674465577a65545254317673644962302f57765a4b7862784c303072786b734f656c4147773852774553617a7a6d66325877454d4d7a774f334c2b67>

Přeji hezký den,

server <https://telecam.cz>

■ **Obrázek B.13** Zpráva o vyhodnocení úlohy

Klient může zobrazenou úlohu využít jako základ pro novou úlohu téhož typu – původní přílohy jsou převzaty, klient může vyplnit nová metadata úlohy, použít původní přílohy nebo je změnit, a pro úlohu se použije šablona v podobě z okamžiku začátku vyplňování původní úlohy.

B.3 Stránky uživatele s rolí terapeut

Uživatel s rolí terapeut má k dispozici následující stránky přístupné prostřednictvím jednoúrovňového menu:

- „Úlohy“ – stránka pro zobrazení provedených úloh přiřazených klientů
- „Nová úloha“ – stránka pro zobrazení formuláře vložení nové úlohy
- „Klienti“ – stránka pro zobrazení přiřazených klientů
- „Nový klient“ – stránka pro založení nového klienta

B.3.1 Stránka „Úlohy“

Na stránce „Úlohy“ na obrázku B.14 aplikace zobrazí přehled všech úloh klientů, které má přihlášený terapeut přiřazen. Zobrazení podporuje stránkování a filtrování. Data je možné filtrovat podle kategorií uvedených v tabulce B.2 a podle osobních údajů klienta. Terapeut uvedený v přehledu je terapeutem, který byl klientovi přiřazen v době provádění úlohy.

Přehled úloh

Od data: **Do data:** **Úloha:** **Stav úlohy:**

Terapeut: **Klient:**

ID	Datum Stav	Název úlohy	Klient E-mail klienta	Terapeut E-mail terapeuta	Operace
212	31.03.2021 Zpracovaná	Cvičení - paže - plný rozsah	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="☰ Otevřít"/>
211	31.03.2021 Čeká na zpracování	ALL	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="☰ Otevřít"/>

Obrázek B.14 Přehled úloh klientů terapeuta

B.3.2 Stránka „Nová úloha“

Na stránce „Nová úloha“ může terapeut zobrazit formulář pro vložení nové úlohy tak, jak to může udělat klient podle odstavce B.2.2. Terapeut ale nemůže vyplněný formulář odeslat.

B.3.3 Stránka „Klienti“

Tato stránka na obrázku B.15 je pro uživatele s rolí terapeut výchozí stránkou po přihlášení do aplikace.

Seznam uživatelů / klientů

Od data: **Do data:** **Stav uživatele:** **Uživatel:**

ID	Vytvořen dne	Typ Stav	Jméno E-mail uživatele	Terapeut E-mail terapeuta	Operace s uživatelem
79	31.03.2021	Klient Aktivní	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="✎ Upravit"/> <input type="button" value="🔑 Obnovit heslo"/> <input type="button" value="👤 Přihlásit"/> <input type="button" value="✉ Nová úloha"/>

Obrázek B.15 Přehled klientů terapeuta

Na stránce „Klienti“ aplikace zobrazí klienty přihlášeného terapeuta. Stránka podporuje stránkování a filtrování. Data je možné filtrovat podle kategorií uvedených v tabulce B.3.

■ **Tabulka B.3** Filtrování klientů

Kategorie	Význam
Od data	Nejnižší datum založení klienta
Do data	Nejvyšší datum založení klienta
Stav uživatele	Stav uživatele
Uživatel	Filtrování podle osobních údajů uživatele

U jednotlivých klientů jsou k dispozici následující funkčnosti přístupné prostřednictvím jednoúrovňového menu:

- „Upravit“ – otevře stránku pro editaci údajů klienta
- „Obnovit heslo“ – odešle klientovi link na nastavení hesla
- „Přihlásit“ – přihlásí terapeuta jako daného klienta
- „Nová úloha“ – umožní terapeutovi vybrat některou z aktivních šablon a odeslat klientovi link na vyplnění úlohy podle této šablony

Níže uvádíme podrobnosti k těmto funkčnostem.

B.3.3.1 Tlačítko „Upravit“

Aplikace otevře stránku na obrázku B.16, na níž může terapeut editovat osobní údaje klienta, jeho stav a údaje podle šablony použité pro osobní údaje při provádění úloh, nemůže ale měnit e-mailovou adresu klienta.

Editace údajů

E-mailová adresa (*):	<input type="text" value="test03@telecam.cz"/>
Titul před jménem:	<input type="text"/>
Jméno (*):	<input type="text" value="Josef"/>
Příjmení (*):	<input type="text" value="Klient"/>
Titul za jménem:	<input type="text"/>
Stav uživatele (*):	<input type="text" value="Aktivní"/>
Terapeut (*):	<input type="text" value="František Terapeut, DiS. (test02@telecam.cz)"/>
Datum narození (*):	<input type="text" value="29.02.2016"/>
Váha [kg] (*):	<input type="text" value="100"/>
Výška [cm] (*):	<input type="text" value="180"/>
<input type="button" value="✓ Uložit"/>	

■ **Obrázek B.16** Editace údajů klienta

B.3.3.2 Tlačítko „Přihlásit“

Aplikace přihlásí uživatele s rolí terapeut jako vybraného klienta. Na stránkách je tato skutečnost indikována textem v pravém horním rohu stránky jako na obrázku B.17. Kliknutím na symbol [x] dojde k návratu do původní uživatelské role.



■ **Obrázek B.17** Přihlášení terapeuta jako klienta

B.3.3.3 Tlačítko „Nová úloha“

Aplikace otevře nové okno na obrázku B.18, kde terapeut vybere úlohu, kterou má klient provést. Po potvrzení výběru aplikace klientovi odešle link na obrázku B.19, pomocí něhož může klient vyplnit formulář úlohy bez přihlášení do aplikace.

Nová úloha pro klienta

Aplikace odešle klientovi link na úlohu vybranou níže.



■ **Obrázek B.18** Odeslání linku na úlohu

Jan Petržílka

From: web@telecam.cz
Sent: sobota 3. dubna 2021 11:41
To: newone@petrzilka.cz
Subject: Link na úlohu / Task link

Dobrý den,

kliknutím na link níže můžete otevřít úlohu "Cvičení - otáčení trupu".

<https://telecam.cz/createlinkedresponse/38302d64436b4346515658787a6834775a634d4c4a46446e50324363715955527667465836336a7a6e7274414e746b394757513466336a70697178375159587255732f546c573944574b69734c782f4c59684a4d704e496e51>

Přeji hezký den,

server <https://telecam.cz>

■ **Obrázek B.19** Link na provedení úlohy bez přihlášení

B.3.4 Stránka „Nový klient“

Tato funkčnost terapeutovi umožňuje založit nového klienta. Terapeut vyplní e-mailovou adresu klienta, jeho osobní údaje a stav a údaje podle aktuální šablony pro osobní údaje používané pro úlohy podobně, jako na obrázku B.16.

B.4 Stránky uživatele s rolí správce

Na rozdíl od uživatelů s rolemi klient nebo terapeut nabízí aplikace uživatelům s rolí správce víceúrovňové menu, jehož struktura a obsah je uveden níže.


- „Uživatelé“ – vstup do menu nižší úrovně
 - „Nový uživatel“ – stránka pro založení nového uživatele
 - „Přehled uživatelů“ – stránka pro zobrazení přiřazených klientů
 - „Změna terapeuta“ – stránka umožňující hromadnou změnu terapeuta u více klientů
- „Úlohy“ – vstup do menu nižší úrovně
 - „Nová úloha“ – stránka pro zobrazení formuláře vložení nové úlohy
 - „Přehled úloh“ – stránka pro zobrazení provedených úloh přiřazených klientů
- „Nastavení“ – vstup do menu nižší úrovně
 - „Konfigurace“ – stránka pro nastavení systémových parametrů aplikace
 - „Šablony úloh“ – přístup k nastavení šablon úloh
 - „Lokalizace“ – stránka pro nastavení jazykově závislých textů
 - „Log“ – zobrazení historie používání aplikace
 - „Test přenosu dat“ – stránka pro ověření rychlosti přenosu dat směrem k serveru

Výchozí stránka uživatelů s rolí administrátor nenabízí žádné konkrétní operace.

B.4.1 Stránka „Nový uživatel“

Uživatel s rolí správce může na této stránce na obrázku B.20 založit jiného uživatele s rolí správce, terapeut nebo klient. Správce vyplňuje e-mailovou adresu uživatele, jeho osobní údaje, roli a stav. V případě nového uživatele s rolí klient vyplňuje navíc terapeuta, jemuž je přiřazen, a osobní údaje podle aktuální šablony pro osobní údaje používané pro úlohy.

Nový uživatel

E-mailová adresa (*):	<input type="text" value="jmeno.prijmeni@firma.cz"/>
Titul před jménem:	<input type="text"/>
Jméno (*):	<input type="text"/>
Příjmení (*):	<input type="text"/>
Titul za jménem:	<input type="text"/>
Typ uživatele (*):	<input type="text" value="Klient"/>
Stav uživatele (*):	<input type="text" value="Aktivní"/>
Terapeut (*):	<input type="text" value="Jan Terapeut (th@petrzilka.cz)"/>
Datum narození (*):	<input type="text"/> 
Váha [kg] (*):	<input type="text" value="30 - 250"/>
Výška [cm] (*):	<input type="text" value="80 - 250"/>
<input type="button" value="✉ Vytvořit uživatele"/>	

■ **Obrázek B.20** Založení nového uživatele

Aplikace novému uživateli odešle link na obrázku B.21 na stránku, kde si nastaví svoje první heslo.

Jan Petržilka

From: web@telecam.cz
Sent: sobota 3. dubna 2021 11:37
To: newOne@petrzilka.cz
Subject: Link na nastavení prvního hesla / Link to create initial password

Dobrý den,

na webu telecam.cz byl registrován uživatelský účet s Vaší e-mailovou adresou. Pro nastavení hesla klikněte, prosím, na link níže.

<https://telecam.cz/setpassword?u=6e65774f6e6540706574727a696c6b612e637a&t=494f2b2f7655386b582b2b2f7656636a48444d>

Děkuji a přeji hezký den,

server <https://telecam.cz>

■ **Obrázek B.21** Link na nastavení prvního hesla

B.4.2 Stránka „Přehled uživatelů“

Na této stránce na obrázku B.22 aplikace zobrazí registrované uživatele. Stránka podporuje stránkování a filtrování. Data je možné filtrovat podle kategorií uvedených v tabulce B.4.

Seznam uživatelů / klientů

Od data: **Do data:** **Typ uživatele:** **Stav uživatele:**

Terapeut: **Uživatel:**

ID	Vytvořen dne	Typ Stav	Jméno E-mail uživatele	Terapeut E-mail terapeuta	Operace s uživatelem
80	03.04.2021	Klient Aktivní	Jan NewOne newone@petrzilka.cz	Jan Terapeut th@petrzilka.cz	<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>
79	31.03.2021	Klient Aktivní	Josef Klient test03@telecam.cz	František Terapeut, DIS. test02@telecam.cz	<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>
78	31.03.2021	Terapeut Aktivní	František Terapeut, DIS. test02@telecam.cz		<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>
76	28.03.2021	Klient Aktivní	Jan Klient cl@petrzilka.cz	Jan Terapeut th@petrzilka.cz	<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>
75	28.03.2021	Terapeut Aktivní	Jan Terapeut th@petrzilka.cz		<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>
74	24.03.2021	Správce Aktivní	Jan Hejda jan.hejda@telecam.cz		<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>
36	05.03.2021	Správce Aktivní	Správce Systému admin@telecam.cz		<input type="button" value="Upravit"/> <input type="button" value="Obnovit heslo"/> <input type="button" value="Přihlásit"/> <input type="button" value="Smazat"/>

■ **Obrázek B.22** Přehled uživatelů

■ **Tabulka B.4** Filtrování uživatelů

Kategorie	Význam
Od data	Nejnižší datum založení klienta
Do data	Nejvyšší datum založení klienta
Typ uživatele	Role uživatele
Stav uživatele	Stav uživatele
Uživatel	Filtrování podle osobních údajů uživatele

U jednotlivých klientů jsou k dispozici následující funkčnosti přístupné prostřednictvím jednoúrovňového menu:

- „Upravit“ – otevře stránku pro editaci údajů klienta
- „Obnovit heslo“ – odešle klientovi link na nastavení hesla
- „Přihlásit“ – přihlásí terapeuta jako daného klienta
- „Smazat“ – odstraní vybraného uživatele

Níže uvádíme podrobnosti k těmto funkčnostem.

B.4.2.1 Tlačítko „Upravit“

Aplikace otevře stránku, na které zobrazí údaje uživatele. Funkčnost je podobná jako funkčnost popsaná v kapitole B.2.1; v tomto případě ale správce může editovat všechny údaje klientů a terapeutů včetně e-mailové adresy. Typ uživatele změnit nelze.

B.4.2.2 Tlačítko „Přihlásit“

Funkčnost je shodná s funkčností popsanou v kapitole B.3.3.2.

B.4.2.3 Tlačítko „Smazat“

Po kliknutí na tlačítko aplikace odstraní vybraného uživatele. V případě, že se jedná o uživatele s rolí terapeut, je uživatel odstraněn pouze v případě, kdy nemá žádné přiřazené klienty ani nevystupuje jako terapeut v žádné úloze. V případě, že se jedná o uživatele s rolí klient, dojde s odstraněním uživatele také k odstranění všech jeho úloh.

B.4.3 Stránka „Změna terapeuta“

Stránka na obrázku B.23 nabízí funkčnost hromadné změny terapeuta u více klientů. Správce vybere původního a nového terapeuta a kliknutím na tlačítko „Změnit“ provede změnu terapeuta u všech klientů původního terapeuta.

Změna terapeuta

Stávající terapeut:

Nový terapeut:

■ Obrázek B.23 Změna terapeuta

B.4.4 Stránka „Nová úloha“

Na stránce „Nová úloha“ může správce zobrazit formulář pro vložení nové úlohy tak, jak to může udělat klient podle odstavce B.2.2. Správce ale nemůže vyplněný formulář odeslat.

B.4.5 Stránka „Přehled úloh“

Na stránce „Úlohy“ na obrázku B.24 aplikace zobrazí přehled všech úloh. Zobrazení podporuje stránkování a filtrování. Data je možné filtrovat podle kategorií uvedených v tabulce B.2 a podle osobních údajů klienta. Úlohu je možné také odstranit.

Přehled úloh

Od data:

Do data:

Úloha:

Stav úlohy:

Terapeut:

Klient:

ID	Datum Stav	Název úlohy	Klient E-mail klienta	Terapeut E-mail terapeuta	Operace
231	02.04.2021 Čeká na zpracování	SpecialForCopy	Jan Klient cl@petrzilka.cz	Jan Terapeut th@petrzilka.cz	<input type="button" value="☰ Otevřít"/> <input type="button" value="✖ Smazat"/>
212	31.03.2021 Zpracovaná	Cvičení - paže - plný rozsah	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="☰ Otevřít"/> <input type="button" value="✖ Smazat"/>
211	31.03.2021 Čeká na zpracování	ALL	Josef Klient test03@telecam.cz	František Terapeut, DiS. test02@telecam.cz	<input type="button" value="☰ Otevřít"/> <input type="button" value="✖ Smazat"/>
210	28.03.2021 Čeká na zpracování	ALL	Jan Klient cl@petrzilka.cz	Jan Terapeut th@petrzilka.cz	<input type="button" value="☰ Otevřít"/> <input type="button" value="✖ Smazat"/>

« 1 »

10 řádek na stránku

■ Obrázek B.24 Přehled úloh z pohledu správce

B.4.6 Stránka „Konfigurace“

Na stránce „Konfigurace“ na obrázcích B.25, B.26 a B.27 je uvedena aktuálně používaná instance databáze a systémová nastavení uvedená v tabulce B.5.

■ **Tabulka B.5** Systémová nastavení

Nastavení	Popis
Vyčištění cache	Html stránky a jejich části jsou pro zajištění rychlého běhu aplikace cachovány – tímto tlačítkem je možné uložené informace odstranit. Funkčnost je vhodná zejména pro situace, kdy ke stejné instanci databáze přistupuje více instancí aplikace, protože při změně nastavení stránek v aplikaci dojde ke smazání cache ve stejné instanci aplikace a nikoliv v ostatních aplikačních instancích.
Lokalizace	Přepínač umožňuje přepnout zobrazení aplikace bez použití lokalizace jako na obrázku B.28. V tomto stavu je většina textů uživatelského rozhraní zobrazena bez lokalizačních textů, takže je možné najít konfigurační položku pro zobrazené texty.
Počet chybných přihlášení za 60 minut pro Captcha test	Počet neúspěšných pokusů o přihlášení během posledních 60 minut, po jejichž překročení je při přihlašování uživatele vyžadováno podstoupit Captcha test.
Počet chybných přihlášení v seanci pro Captcha test	Počet neúspěšných pokusů o přihlášení během jedné seance, po jejichž překročení je při přihlašování uživatele vyžadováno podstoupit Captcha test.
Výstupní složka	Složka, do které jsou ukládány obsahy vyplněných formulářů a jejich přílohy, a odkud jsou importovány výsledky zpracování úloh.
Heslo pro Captcha test pro přihlášení	Hodnota „secret“ použitá pro Google reCaptcha v2 pro přihlášení uživatele.
Klíč pro Captcha test pro přihlášení	Hodnota „key“ použitá pro Google reCaptcha v2 pro přihlášení uživatele.
Heslo pro Captcha test pro obnovení hesla	Hodnota „secret“ použitá pro Google reCaptcha v2 pro obnovení hesla.
Klíč pro Captcha test pro obnovení hesla	Hodnota „key“ použitá pro Google reCaptcha v2 pro obnovení hesla.
Nastavení e-mailového serveru	Nastavení konfigurace e-mailového serveru. Údaje je možné před uložením vyzkoušet. Aplikace údaje, které nelze použít k odeslání e-mailu, neuloží.

Nastavení systému

Data source

Data source: jdbc:postgresql://telecam.cz:5432/telecam

Vyčištění cache

Vyčištění cache:

 Smazat cache

Lokalizace

Lokalizace:



 Uložit nastavení lokalizace

Počet chybných přihlášení za 60 minut pro Captcha test

Limit:

10

 Nastavit limit za 60 minut

Počet chybných přihlášení v seanci pro Captcha test

Limit:

3

 Nastavit limit na seanci

Výstupní složka

Složka:

/remote

 Nastavit složku

Heslo pro Captcha test pro přihlášení

Heslo:

↓ Uložit heslo

Klíč pro Captcha test pro přihlášení

Klíč:

↓ Uložit klíč

Heslo pro Captcha test pro obnovení hesla

Heslo:

↓ Uložit heslo

Klíč pro Captcha test pro obnovení hesla

Klíč:

↓ Uložit klíč

■ Obrázek B.26 Konfigurace – část 2

Nastavení e-mailového serveru

Start TLS:	<input type="text" value="true"/>
Port:	<input type="text" value="587"/>
Host:	<input type="text" value="bulk.smtp.cz"/>
Auth:	<input type="text" value="true"/>
Odesílatel:	<input type="text" value="web@telecam.cz"/>
Heslo:	<input type="password"/>

■ **Obrázek B.27** Konfigurace – část 3

admin@telecam.cz (id 36)

config -> `${adminPage}.CZ`

config -> `${dataSourceHeading}.CZ`

config -> `${dataSource}.CZ:` jdbc:postgresql://telecam.cz:5432/telecam

config -> `${cacheCleanup}.CZ`

config -> `${cacheCleanup}.CZ:`

config -> `${localization}.CZ`

config -> `${localizationLabel}.CZ:`

■ **Obrázek B.28** Aplikace s vypnutou lokalizací

B.4.7 Stránka „Šablony úloh“

Na stránce na obrázku B.29 jsou zobrazeny definované šablony úloh. Stránka podporuje stránkování a filtrování. Data je možné filtrovat podle kategorií uvedených v tabulce B.6. Na stránce jsou také pole pro přidání nové šablony.

Seznam šablon pro úlohy

Od data: **Do data:** **Textové hledání:**

ID	Datum	Aktivní	Název Identifikátor	Pořadí	Klientská data	Operace se šablonou
73	10.03.2021	<input checked="" type="checkbox"/>	Standardní osobní data standardClientData	71	<input checked="" type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✖ Smazat"/>
83	27.03.2021	<input type="checkbox"/>	IresultData IresultData	83	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✖ Smazat"/>
87	31.03.2021	<input type="checkbox"/>	Výsledek cvičení IexerciseResult	84	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✖ Smazat"/>
85	31.03.2021	<input checked="" type="checkbox"/>	Cvičení - paže - plný rozsah exerciseArmFullRange	86	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✖ Smazat"/>
86	31.03.2021	<input checked="" type="checkbox"/>	Cvičení - otáčení trupu exerciseTrunkRotation	87	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✖ Smazat"/>

▾

Nová šablona

Identifikátor:

CZ

Název šablony:

EN

■ **Obrázek B.29** Šablony úloh

■ **Tabulka B.6** Filtrování šablon

Kategorie	Význam
Od data	Nejnižší datum vytvoření šablony
Do data	Nejvyšší datum vytvoření šablony
Textové hledání	Textové hledání v identifikátoru šablony, v jejím názvu a v identifikátorech, názvech komponent a v textech komponent k zobrazení

Každá šablona má, stejně jako každá komponenta šablony, identifikátor a český a anglický název. Identifikátor a alespoň jeden z názvů jsou povinná pole. Identifikátor je určený k použití systémem pro zpracování úloh – hodnoty interních klíčů webové aplikace pro něj nemusí být vhodné a právě pro tento případ jsou určeny tyto identifikátory.

Šablony se v aplikaci používají v následujících případech:

- Při editaci údajů uživatele s rolí klient k vyplnění osobních dat klienta, která se používají při provedení úlohy.
- Jako formulář pro provedení úlohy.
- Jako formulář pro zobrazení výsledku zpracování úlohy.

Jednotlivé sloupce v přehledu mají význam popsáný v tabulce B.7:

■ **Tabulka B.7** Atributy šablon

Sloupec	Význam
ID	Interní aplikační klíč
Datum	Datum vytvoření šablony
Aktivní	Aktivní šablony, které nejsou šablonami pro klientská data, jsou nabízeny klientům k vypracování úlohy
Název	Název šablony v aktuálně zvoleném jazyce; pokud není název šablony ve zvoleném jazyce k dispozici, použije se název ve druhém jazyce s uvedením jazyka názvu
Identifikátor	Externí klíč šablony
Pořadí	Pořadí šablony při zobrazení v nabídce pro provedení úlohy
Klientská data	Označení šablony jako šablony používané pro osobní data klientů při provedení úlohy; právě jedna šablona musí být současně aktivní a označená jako klientská data
Operace – šipka nahoru	Zvýšení pořadí šablony, v případě první šablony její přesun na nejnižší pozici; v případě aplikace filtru probíhá posun v rámci výběru podle filtru
Operace – šipka dolů	Snížení pořadí šablony, v případě poslední šablony její přesun na nejvyšší pozici; v případě aplikace filtru probíhá posun v rámci výběru podle filtru
Operace – Upravit	Otevírá stránku pro úpravu obsahu šablony
Operace – Smazat	Odstraní šablonu

V dolní části stránky jsou pole pro založení nové šablony.

B.4.7.1 Editace šablony

Po kliknutí na tlačítko „Upravit“ na seznamu šablon otevře aplikace stránku na obrázku B.30 s obsahem šablony a umožní její úpravu. Stránka umožňuje šablonu přejmenovat, přidat do ní novou komponentu, měnit pořadí stávajících komponent, obsah komponent upravovat nebo komponenty ze šablony odstraňovat.

Význam identifikátorů a názvů komponent je shodný s významem identifikátorů a názvů šablon. V tabulce B.8 níže uvádíme popis sloupců, jejichž význam není jinak zřejmý:

■ **Tabulka B.8** Atributy komponent

Sloupec	Význam
Povinné	Takto označené části šablony jsou při provádění úlohy povinné a bez jejich vyplnění není možné úlohu odeslat
Předvyplněno	Takto označené části šablony jsou při použití šablony k provedení úlohy předvyplněny osobními daty klienta nebo daty z minulého provedení stejné úlohy týměž klientem

Aplikace pracuje s typy komponent uvedenými v tabulce B.9:

■ **Tabulka B.9** Typy komponent šablony

č.	Typ komponenty	Popis
1	Statický text	Statický text zobrazený ve formuláři, řídicí html pole jsou zobrazena jako text
10	Html text	Statický text zobrazený ve formuláři, řídicí html pole jsou použita jako řídicí pole
2	Číselné pole	Číselný vstup s nepovinným určením mezí od – do
3	Textové pole	Textový vstup
4	Datum	Vstup data, v případě české verze se použije formát dd.mm.yyyy, v případě anglické verze formát mm/dd/yyyy
5	Zaškrtačovací pole	Zaškrtačovací pole, pokud je toto pole označeno jako povinné, musí být zaškrtnuto
6	Videonahrávka	Pořízení videonahrávky nebo vložení souboru s videonahrávkou s příponou webm, avi, mp4 nebo mov
7	Soubor typu video	Vložení souboru s videonahrávkou s příponou avi nebo mp4
9	Soubor typu fotografie	Vložení souboru s příponou jpg nebo jpeg
11	Soubor typu obrázek	Vložení souboru s příponou jpg, jpeg, png nebo gif
8	Soubor	Vložení souboru s příponou jpg, jpeg, png, txt, pdf, doc, docx, nebo zip

Šablona úlohy 73

Identifikátor:

CZ

Název šablony:

EN

Seznam komponent

ID	Typ komponenty	Název Identifikátor	Pořadí	Povinné	Předvyplněno	Operace s komponentou
117	Datum	Datum narození dateOfBirth	117	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✕ Smazat"/>
118	Číselné pole	Váha [kg] weight	118	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✕ Smazat"/>
119	Číselné pole	Výška [cm] height	119	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎ Upravit"/> <input type="button" value="✕ Smazat"/>

Nová komponenta

Identifikátor:

Typ komponenty:

CZ

Název komponenty:

EN

■ Obrázek B.30 Šablona úlohy

B.4.7.2 Editace komponenty

Po kliknutí na tlačítko „Upravit“ na obsahu šablony aplikace otevře stránku na obrázku B.31 s určenou komponentou a umožní její úpravu. Je možné komponentu přejmenovat, změnit její typ a případně další hodnoty jako statický text nebo rozsah číselné vstupní hodnoty.

Komponenta šablony 118











Identifikátor:	<input type="text" value="weight"/>
Typ komponenty:	<input type="text" value="Číselné pole"/>
Hodnota od:	<input type="text" value="30"/>
Hodnota do:	<input type="text" value="250"/>
CZ	
Název komponenty:	<input type="text" value="Váha [kg]"/>
EN	
Název komponenty:	<input type="text" value="Weight [kg]"/>

■ Obrázek B.31 Editace komponenty šablony

B.4.8 Stránka „Lokalizace“

Aplikace je připravena jako vícejazyčná a konfigurovaná pro použití česky a anglicky. Texty v jednotlivých jazycích je možné nastavit právě prostřednictvím této funkčnosti. Aplikace nejprve otevře seznam všech html stránek na obrázku B.32, na něž je uživatelské rozhraní rozděleno – jedná se jak o šablony celých stránek, tak o šablony dílčích komponent nebo ovládacích prvků. Na stránce je také možné do aplikace přidat novou html stránku.

Seznam stránek

ID	Název stránky	Operace se stránkou
73	!resultData	 Editovat
72	admin	 Editovat
11	changePassword	 Editovat
64	changeTherapeutist	 Editovat
37	component	 Editovat
35	componentElement	 Editovat
60	usersResetPassword	 Editovat
57	userState	 Editovat
56	userType	 Editovat
49	viewResponse	 Editovat

Nová stránka

Název nové stránky:

 Vytvořit

■ **Obrázek B.32** Html stránky (seznam zkrácen)

B.4.8.1 Editace prvků na html stránce

Po kliknutí na vybranou html stránku aplikace otevře další stránku na obrázku B.33, kde je zobrazen jednak zdrojový text html stránky, jednak seznam všech prvků a nastavených hodnot v jednotlivých jazycích. Hodnoty prvků je možné měnit, samotné prvky pak přidávat nebo odstraňovat; stránku je možné také přejmenovat nebo odstranit.

U každého prvku je uveden text zástupného řetězce použitého na stránce, text k zobrazení v češtině nebo angličtině, a informace o tom, zda se jedná o systémový prvek – pro takto označené prvky (například se jedná o aplikační menu) aplikace zobrazuje lokalizovaný text i v případě vypnuté lokalizace aplikace, aby aplikaci bylo možné dále používat.

Lokalizace stránky

Název stránky:

Seznam prvků

Šablona

Vzor:

```
<label for="systemElement${elementId}" class="col-md-2 col-xs-12 control-label">${systemElement}:
</label>
<div class="col-md-10 col-xs-12">
  <span class="form-control"><input id="systemElement${elementId}" type="checkbox" ${checked} />
</span>
```

CZ:

EN:

Systémový prvek:

Nový prvek

Název nového prvku:

■ Obrázek B.33 Editace html stránky

B.4.9 Stránka „Log“

Aplikace vede log o provedených operacích. Tento log je ukládán jednak do souboru na webovém serveru, jednak do aplikační databáze. Počet záznamů je omezen na 100.000 položek. Na stránce na obrázku B.34 aplikace zobrazí uložené záznamy seřazené od nejnovějších. Stránka podporuje stránkování a filtrování.

Log

Od data:

Do data:

ID	Čas (GMT)	Třída	Úroveň	Text	Podrobnosti	Uživatel
47864	2021-04-03T17:25:48.167709Z	...e.admin.ShowLog (class cz.telecam.fe.admin.ShowLog)	info	GET	/admin/showlog	admin@telecam.cz (id 36)
47863	2021-04-03T17:25:38.726414Z	...fe.admin.Admin (class cz.telecam.fe.admin.Admin)	info	GET	/admin	admin@telecam.cz (id 36)
47862	2021-04-03T17:25:37.630244Z	...elecram.fe.Login (class cz.telecam.fe.Login)	info	system action	...telecam.cz from 94.112.194.164 (password login admin@telecam.cz from 94.112.194.164)	admin@telecam.cz (id 36)
47861	2021-04-03T17:25:37.252543Z	...elecram.fe.Login (class cz.telecam.fe.Login)	info	POST	/login	
47860	2021-04-03T17:24:54.749614Z	...elecram.fe.Login (class cz.telecam.fe.Login)	info	GET	/login	
47859	2021-04-03T17:08:59.321439Z	...cam.fe.Homepage (class cz.telecam.fe.Homepage)	info	GET	/ from 34.86.35.31	
47858	2021-04-03T16:32:32.460871Z	...m.fe.admin.Page (class cz.telecam.fe.admin.Page)	info	GET	/admin/page/12	admin@telecam.cz (id 36)
47857	2021-04-03T16:32:32.463501Z	...m.fe.admin.Page (class cz.telecam.fe.admin.Page)	info	GET	/admin/page/12	admin@telecam.cz (id 36)
47856	2021-04-03T16:32:24.161838Z	...fe.admin.Pages (class cz.telecam.fe.admin.Pages)	info	GET	/admin/pages	admin@telecam.cz (id 36)
47855	2021-04-03T16:32:11.234530Z	...m.fe.admin.Page (class cz.telecam.fe.admin.Page)	info	GET	/admin/page/66	admin@telecam.cz (id 36)

« 1 2 3 4784 4785 »

10 řádek na stránku

■ Obrázek B.34 Log

Data je možné filtrovat podle kategorií uvedených v tabulce B.10.

■ Tabulka B.10 Filtrování logů

Kategorie	Význam
Od data	Nejnižší datum záznamu v časové zóně GMT
Do data	Nejvyšší datum záznamu v časové zóně GMT

B.4.10 Stránka „Test přenosu dat“

Přenos dat z prohlížeče na webový server je jednou z klíčových funkcí aplikace. Součástí aplikace je proto také stránka na obrázku B.35, která umožňuje ověřit rychlost přenosu dat. Přenos je možné spustit s různým simulovaným zpožděním, které je implementováno jako čekání na straně serveru před odesláním odpovědi. Význam jednotlivých polí je popsán v tabulce B.11.

■ **Tabulka B.11** Atributy testu přenosu dat

Pole	Význam
Ping bez zátěže [ms]	Čas odezvy serveru měřený v prohlížeči před začátkem přenosu dat
Velikost dat [MB]	Aktuální velikost datového balíčku
Doba přenosu a zpracování dat [ms]	Vypočítaná doba, kterou trvá přenos dat oběma směry a jejich zpracování na serveru
Rychlost podle prohlížeče [Mb/s]	Tok dat měřený prohlížečem pro jeden datový balíček (jedná se o přenos datového obsahu, nikoliv přenos dat po lince)
Rychlost na serveru [Mb/s]	Tok dat měřený na serveru pro všechny probíhající seance (jedná se o přenos datového obsahu, nikoliv přenos dat po lince)
Celkem přeneseno [MB]	Celkový objem přenesených dat od začátku přenosu
Přeneseno souborů	Celkový počet přenesených datových balíčků od začátku přenosu
Ztraceno souborů	Celkový počet ztracených datových balíčků od začátku přenosu
Počet přenášených souborů	Aktuální počet datových balíčků, které jsou odesílány nebo byly odeslány a jejich přenos dosud nebyl potvrzen
Minut	Počet celých minut od začátku přenosu
Vteřin	Počet vteřin od poslední celé minuty od začátku přenosu
Průměrná rychlost [Mb/s]	Průměrná rychlost přenosu vypočítaná jako podíl přenesených dat a uplynulého času (jedná se o přenos datového obsahu, nikoliv přenos dat po lince)

Test přenosu dat

▶ Start se simulovaným zpožděním 0 ms

▶ Start se simulovaným zpožděním 250 ms

▶ Start se simulovaným zpožděním 500 ms

▶ Start se simulovaným zpožděním 750 ms

▶ Start se simulovaným zpožděním 1000 ms

■ Stop

Ping bez zátěže [ms]: 44
Velikost dat [MB]: 0.762
Doba přenosu a zpracování dat [ms]: 228
Rychlost podle prohlížeče [Mb/s]: 108.837
Rychlost na serveru [Mb/s] 32.834
Celkem přeneseno [MB]: 335.98
Přeneseno souborů: 385
Ztraceno souborů: 0
Počet přenášených souborů: 0
Minut: 1
Vteřin: 0
Průměrná rychlost [Mb/s]: 44.592

Ping bez zátěže [ms]: 551
Velikost dat [MB]: 0.786
Doba přenosu a zpracování dat [ms]: 244
Rychlost podle prohlížeče [Mb/s]: 23.302
Rychlost na serveru [Mb/s] 38.273
Celkem přeneseno [MB]: 315.78
Přeneseno souborů: 408
Ztraceno souborů: 0
Počet přenášených souborů: 0
Minut: 1
Vteřin: 0
Průměrná rychlost [Mb/s]: 41.464

Ping bez zátěže [ms]: 1044
Velikost dat [MB]: 0.705
Doba přenosu a zpracování dat [ms]: 148
Rychlost podle prohlížeče [Mb/s]: 32.391
Rychlost na serveru [Mb/s] 45.154
Celkem přeneseno [MB]: 299.52
Přeneseno souborů: 427
Ztraceno souborů: 0
Počet přenášených souborů: 0
Minut: 1
Vteřin: 0
Průměrná rychlost [Mb/s]: 38.991

■ **Obrázek B.35** Test přenosu dat se zpožděním zleva 0, 500 a 1000 ms na lince s rychlostí 50 Mb/s

Instalační příručka

Instalační příručka obsahuje popis instalace a administrace aplikace a popis souborů o provedených úlohách a jejich vyhodnocení, které jsou vyměňovány s návaznou aplikací.

Telecam.cz je webová aplikaci založená na Spring Boot frameworku používající webový server zapouzdřený v distribuovaném archivu a databázi PostgreSQL.

Vlastní aplikace je distribuována jako `jar` soubor `telecam.jar`, který lze spustit v Java Runtime Environmentu podporujícím Javu 11. Spuštěná aplikace vytvoří http webový server dostupný na portu 8089. Tento webový server je zpřístupněn pro okolní svět pomocí reverzní proxy Nginx, která službu vystaví na standardním portu 443 pro https protokol.

Součástí zdrojových souborů aplikace jsou inicializační databázové skripty.

Aplikace byla vyvinuta s použitím Spring Boot frameworku verze 2.4.1 a databáze PostgreSQL verze 10.16. Nasazení bylo ověřeno na Ubuntu Serveru 18.04 LTS. Příklady příkazů uvedené níže jsou platné pro tuto konfiguraci.

C.1 Instalace aplikace

Celý postup nasazení aplikace se skládá z těchto kroků:

- Instalace databáze PostgreSQL.
- Vytvoření databázových tabulek a jejich naplnění konfiguračními daty.
- Instalace Java Runtime Environmentu.
- Spuštění aplikace s nastavením přístupových údajů k databázi.
- Instalace a konfigurace reverzní proxy Nginx..

Tyto kroky jsou detailněji popsány v následujících kapitolách.

C.1.1 Instalace databáze

Nejprve je nutné na vhodný server nainstalovat databázi PostgreSQL ve verzi 10.16 z webových stránek <https://www.postgresql.org/>.

Pro správu databázového serveru doporučujeme použít nástroj pgAdmin dostupný na webových stránkách <https://www.pgadmin.org/>. Tento nástroj je také součástí instalace databáze PostgreSQL.

V databázovém serveru je nutné vytvořit uživatele, který bude k databázi přistupovat, a databázi pro umístění aplikačních tabulek. Doporučeným názvem uživatele je telecam a doporučeným názvem databáze je telecam.

Pomocí nástroje pro správu databáze je potřeba provést následující skripty z datového média, které je součástí této práce:

- Skript „create.sql“ který vytvoří datové tabulky.
- Skript „insert.sql“ který tabulky naplní konfiguračními daty.

Tímto je databáze připravena k připojení vlastní aplikace.

C.1.2 Instalace Java Runtime Environmentu

Na vhodný server je potřeba nainstalovat Java Runtime Environment podporující Javu 11. Prostředí je možné nainstalovat například ze stránky <https://jdk.java.net/java-se-ri/11> nebo alternativně ze stránky <https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html> (ve druhém případě je nutné zakoupit licenci).

Java Runtime Environment pro demonstrační instalaci byl nainstalován pomocí následujících příkazů:

```
sudo apt-get update
sudo apt install openjdk-11-jdk
```

C.1.3 Spuštění aplikace

Spuštění lze provést z příkazového řádku zadáním příkazu `java -jar tisk-portal.jar` s níže uvedenými parametry popisujícími přístup k databázi:

```
java -Dspring.datasource.url=<URL>
-Dspring.datasource.driverClassName=org.postgresql.Driver
-Dspring.datasource.username=<USER> -Dspring.datasource.password=<PASS>
-jar tisk-portal.jar
```

kde

- <URL> je adresa databáze,
- <USER> je identifikace uživatele databáze a
- <PASS> je přístupové heslo uživatele databáze.

Přístupové údaje je možné nastavit v souboru `src/main/resources/application.properties` před kompilací aplikace.

Aplikace je po spuštění dostupná na adrese `http://localhost:8089`. Výchozí přístupové údaje pro správce aplikace jsou uvedeny v tabulce C.1. Pokud v aplikaci není veden uživatel s uvedeným uživatelským jménem, aplikace jej při restartu založí s uvedeným heslem.

- **Tabulka C.1** Výchozí přístupové údaje

Uživatelské jméno	Heslo
admin@telecam.cz	cc6hSQu7ff

C.1.4 Reverzní proxy

Pro zpřístupnění aplikace pro okolní svět pomocí protokolu https na standardním portu 443 je použita reverzní proxy Nginx. Reverzní proxy Nginx je dostupná v rámci obvyklých instalačních balíčků pro Ubuntu Server 18.04 LTS pomocí příkazu:

```
sudo apt-get install nginx
```

Následně je potřeba povolit přístup k aplikaci Nginx přes firewall:

```
sudo ufw allow 'Nginx HTTPS'
```

Konfigurace reverzní proxy je definována v souboru `/etc/nginx/sites-available/reverse-proxy.conf`, který je potřeba symbolicky linkovat na soubor `/etc/nginx/sites-enabled/reverse-proxy.conf`.

Obsah konfiguračního souboru uvedeného výše je potřeba nastavit podle obrázku C.1 (služba bude podle uvedené konfigurace dostupná na adrese telecam.cz). Konfigurace zahrnuje i nastavení maximální velikosti přenášeného souboru a přeposlání IP adresy klientské stanice aplikaci.

```
server {
    server_name telecam.cz;
    client_max_body_size 10m;
    sendfile on;
    location / {
        proxy_pass http://127.0.0.1:8089/;
        client_max_body_size 10m;
        client_body_buffer_size 1024k;
        proxy_buffer_size 1024k;
        proxy_buffers 6 500k;
        proxy_busy_buffers_size 1024k;
        proxy_temp_file_write_size 1024k;
        proxy_set_header X-Real-IP $proxy_add_x_forwarded_for;
    }
    listen [::]:80;
    listen 80;
}
server {
    server_name www.telecam.cz;
    return 301 https://telecam.cz;
    listen [::]:80;
    listen 80;
}
```

■ Obrázek C.1 Konfigurace reverzní proxy

Posledním krokem je získání a instalace TLS/SSL certifikátů pro zajištění protokolu https. To lze provést s využitím aplikace Certbot. Instalace aplikace Certbot probíhá s využitím packet manageru:

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install python-certbot-nginx
```

Po instalaci Certbotu je potřeba jej spustit a dále postupovat pomocí interaktivního průvodce:

```
certbot --nginx
```

Aplikace je nyní dostupná na adrese <https://telecam.cz>.

C.1.5 Logování

Aplikace pro logování používá framework LogBack, který je součástí SpringBoot frameworku. Konfigurace výstupních souborů je k dispozici v souboru `src/main/resources/logback-spring.xml`. Při výchozím nastavení jsou logy vytvářeny do složky `logs` vytvořené ve složce, ve které je spuštěn soubor `.jar`. Počet záznamů v databázi je omezen na 100.000 položek; aplikace odstraní přebytečné záznamy každých 8 hodin.

C.1.6 Sestavení aplikace

Aplikaci je možné sestavit pomocí nástroje Maven následujícím příkazem:

```
mvn clean package -Dmaven.test.skip=true
```

Výstupem sestavení je soubor `.jar`. Pokud je vhodné při sestavení provedení všech testů, je možné vynechat poslední část příkazu – v tom případě je ale potřeba připravit testovou databázi s přístupovými údaji uvedenými v souboru `src/test/resources/application.properties` pro potřeby integračních testů. Tato testová databáze musí obsahovat běžnou konfiguraci aplikace; v průběhu integračních testů je odeslána e-mailová zpráva na adresu `testingX@telecam.cz`, kde X je celé číslo. Proti jedné testové databázi může současně běžet nejvýše jedna instance testů, jinak může dojít k chybě. Všechny požadované závislosti jsou uvedeny v konfiguračním souboru `pom.xml`.

C.2 Administrace aplikace

Pro správný běh aplikace je potřeba nastavit přístupové údaje ke službě reCaptcha, k e-mailovému serveru a zajistit přenos souborů do aplikace zpracovávající úlohy a zpět.

C.2.1 Přístupové údaje ke službě reCaptcha

Aplikace používá službu reCaptcha v2 k posílení bezpečnosti při přihlašování uživatelů a k obnově hesla uživatelů. Ve výchozím nastavení jsou používány přístupové údaje autora aplikace, vlastní přístupové údaje provozovatele aplikace je možné nastavit v uživatelském rozhraní aplikace po přihlášení správce aplikace.

C.2.2 Přístupové údaje k e-mailovému serveru

Aplikace používá e-mailový server k odesílání zpráv uživatelům. Přístupové údaje k e-mailovému serveru je možné nastavit v uživatelském rozhraní aplikace po přihlášení správce aplikace. Výchozím nastavením je použití e-mailového serveru společnosti Active24 a účtů autora aplikace, aplikace byla otestována také na použití školního e-mailového serveru na doméně `cvut.cz`. Vzorové nastavení pro školní server je uvedeno v tabulce C.2.

■ **Tabulka C.2** Připojení školního e-mailového serveru

Pole	Hodnota
Start TLS	true
Port	587
Host	smtp.office365.com
Odesílatel	username@cvut.cz
Heslo	heslo

C.2.3 Přenos souborů

Aplikace ukládá informace o provedených úlohách do souborové složky sdílené s návaznou aplikací, která data vyhodnocuje. Aplikace byla otestována na Wedos VPS serveru s daty sdílenými přes Wedos disk. Přístup aplikace na Wedos disk byl zajištěn pomocí protokolu CIFS; sdílení souborů s návaznou aplikací pak pomocí protokolu rsync.

V této kapitole jsou dále popsány výstupní s vstupní soubory. Každá úloha má soubory uložené ve vlastní složce, jejíž název je odvozen od aktuálního systémového času serveru v milisekundách. Výstupní soubory jsou uvedeny v tabulce C.3 a obsahují data o provedené úloze, vstupní soubory jsou uvedeny v tabulce C.4 a obsahují data o vyhodnocení úlohy. Klíčové soubory `clientData.json` a `taskData.json` mají schéma definované na obrázcích C.2 a C.3. Příklady souborů jsou na obrázcích C.4 a C.5. Typy komponent jsou číslovány podle tabulky C.5.

Aplikace prochází složky dosud nezpracovaných úloh každých 20 vteřin. Při jednom průchodu načte nejvýše jeden soubor o zpracované úloze. Toto omezení je implementováno proto, aby aplikace neposílala příliš mnoho e-mailových zpráv najednou.

■ **Tabulka C.3** Výstupní soubory

Soubor	Obsah
<code>clientData.json</code>	json soubor s údaji úlohy podle šablony používané pro osobní údaje
<code>clientTemplate.json</code>	json soubor s definicí šablony používané pro osobní údaje
<code>taskData.json</code>	json soubor s údaji úlohy podle šablony úlohy
<code>taskTemplate.json</code>	json soubor s definicí šablony úlohy
<code>response.ready</code>	soubor vytvořený ve složce jako poslední přejmenováním dočasného souboru; soubor indikuje připravenost ostatních souborů ve složce ke zpracování, jeho obsahem je json soubor s primárním klíčem úlohy, například <code>{"responseId":232}</code>
<code>rawData.json</code>	json soubor s daty v podobě, v jaké je server přijal od prohlížeče uživatele
<code>xxx.data</code>	Příloha úlohy, kde xxx je primární klíč komponenty šablony úlohy
<code>xxx.originalResponse</code>	json soubor s informací o úloze, ze které byla aktuální úloha odvozena; xxx je primární klíč původní úlohy a uvnitř souboru je také název složky, kde je původní úloha uložena, například <code>{"originalResponseId": 231, "originalFolder": "1617371231365"}</code>
<code>xxx.newResponse</code>	json soubor s informací o úloze, která byla z aktuální úlohy odvozena; xxx je primární klíč nové úlohy a uvnitř souboru je také název složky, kde je nová úloha uložena, například <code>{"newResponseId": 233, "newFolder": "1617371342627"}</code>

■ **Tabulka C.4** Vstupní soubory

Soubor	Obsah
xxx.data	soubor přílohy
resultData.json	json soubor s údaji o výsledku zpracování
processed	soubor svou přítomností indikující dostupnost ostatních souborů se zpracováním výsledků

Soubor `resultData.json` má obdobné schéma jako soubor `taskData.json`. Hlavním rozdílem je to, že uvedený identifikátor šablony identifikuje šablonu, která je použita pro zobrazení výsledků zpracování. Při zobrazení výsledků musí u jednotlivých komponent souhlasit jak typ komponenty, tak identifikátor komponenty. Současně je identifikátor šablony použit jako identifikátor html stránky použité pro lokalizaci textů a tak je zajištěna vícejazyčnost zobrazených výsledků. Příklad souboru je na obrázku C.6.

Soubory popisující schémata a příklady souborů jsou k dispozici na datovém médiu, které je součástí této práce.

■ **Tabulka C.5** Typy komponent šablony

č.	Typ komponenty	Popis
1	Statický text	Statický text zobrazený ve formuláři, řídicí html pole jsou zobrazena jako text
10	Html text	Statický text zobrazený ve formuláři, řídicí html pole jsou použita jako řídicí pole
2	Číselné pole	Číselný vstup s nepovinným určením mezí od – do
3	Textové pole	Textový vstup
4	Datum	Vstup data, v případě české verze se použije formát dd.mm.yyyy, v případě anglické verze formát mm/dd/yyyy
5	Zaškrťovací pole	Zaškrťovací pole, pokud je toto pole označeno jako povinné, musí být zaškrtnuto
6	Videonahrávka	Pořízení videonahrávky nebo vložení souboru s videonahrávkou s příponou webm, avi, mp4 nebo mov
7	Soubor typu video	Vložení souboru s videonahrávkou s příponou avi nebo mp4
8	Soubor	Vložení souboru s příponou jpg, jpeg, png, txt, pdf, doc, docx, nebo zip
9	Soubor typu fotografie	Vložení souboru s příponou jpg nebo jpeg
11	Soubor typu obrázek	Vložení souboru s příponou jpg, jpeg, png nebo gif

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Client Data",
  "description": "Data of the client data part of the task",
  "type": "object",
  "definitions": {
    "numericField": {
      "type": "object",
      "properties": {
        "identifier": {
          "type": "string",
          "description": "Identifier of the template component"
        }
      }
    }
  }
}
```

```
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 2
    },
    "value": {
      "type": "string",
      "description": "Value for the component captured by
the client"
    },
  }
},
"textField": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 3
    },
    "value": {
      "type": "string",
      "description": "Value for the component captured by
the client"
    },
  },
}
},
"dateField": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 4
    },
    "value": {
      "type": "string",
      "description": "Value for the component captured by
the client - date in format dd.mm.yyyy"
    },
  },
}
},
"checkBox": {
```

```

    "type": "object",
    "properties": {
      "identifier": {
        "type": "string",
        "description": "Identifier of the template component"
      },
      "type": {
        "type": "integer",
        "description": "Type of the template component",
        "const": 5
      },
      "value": {
        "type": "string",
        "description": "Value for the component captured by
          the client - checked or empty",
        "enum": ["checked", ""]
      },
    }
  },
  "componentValue": { "anyOf": [
    { "$ref": "#/definitions/textField" },
    { "$ref": "#/definitions/numericField" },
    { "$ref": "#/definitions/dateField" },
    { "$ref": "#/definitions/checkbox" }
  ] },
},
"properties": {
  "templateIdentifier": {
    "description": "Identifier of the template used for the client
      data part of the task",
    "type": "string"
  },
  "userId": {
    "description": "Primary key of the user",
    "type": "integer"
  },
  "username": {
    "description": "E-mail address of the user",
    "type": "string"
  },
  "name": {
    "description": "Full name of the user",
    "type": "string"
  },
  "fields": {
    "description": "List of values captured by the client",
    "type": "array",
    "items": { "$ref": "#/definitions/componentValue" }
  },
},
},

```

```
"required": ["templateIdentifier", "userId", "username", "name", "fields"]
}
```

■ **Obrázek C.2** Schéma souboru clientData.json

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Task Data",
  "description": "Data of the task specific part of the task",
  "type": "object",
  "definitions": {
    "numericField": {
      "type": "object",
      "properties": {
        "identifier": {
          "type": "string",
          "description": "Identifier of the template component"
        },
        "type": {
          "type": "integer",
          "description": "Type of the template component",
          "const": 2
        },
        "value": {
          "type": "string",
          "description": "Value for the component captured by the
            client"
        }
      },
      "required": ["identifier", "type", "value"]
    },
    "textField": {
      "type": "object",
      "properties": {
        "identifier": {
          "type": "string",
          "description": "Identifier of the template component"
        },
        "type": {
          "type": "integer",
          "description": "Type of the template component",
          "const": 3
        },
        "value": {
          "type": "string",
          "description": "Value for the component captured by the
            client"
        }
      },
      "required": ["identifier", "type", "value"]
    }
  }
}
```

```

"dateField": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 4
    },
    "value": {
      "type": "string",
      "description": "Value for the component captured by the
        client - date in format dd.mm.yyyy"
    },
  },
  "required": ["identifier", "type", "value"]
},
"checkbox": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 5
    },
    "value": {
      "type": "string",
      "description": "Value for the component captured by the
        client - checked or empty",
      "enum": ["checked", ""]
    },
  },
  "required": ["identifier", "type", "value"]
},
"video": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 6
    }
  }
}

```



```
    },
    "filename": {
      "type": "string",
      "description": "Name of the file as created by the front
        -end (when create by camera) or name of the file as
        attached by the user"
    },
    "file": {
      "type": "string",
      "description": "Name of the file stored in the shared
        folder"
    },
    "settings": { "anyOf": [
      { "$ref": "#/definitions/settingsForCamera" },
      { "$ref": "#/definitions/settingsForSelectedFile" },
      { "$ref": "#/definitions/settingsEmpty" }
    ]
  }
},
"required": ["identifier", "type", "filename", "file", "settings"]
},
"settingsForCamera": {
  "description": "Recorded file",
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "const": "camera"
    },
    "selectedMimeType": {
      "type": "string",
      "description": "Video format selected for recording such
        as 'video/webm;codecs=h264'"
    },
    "recordedMimeType": {
      "type": "string",
      "description": "Video format recorded such as
        'video/x-matroska;codecs=avc1,opus'"
    },
    "videoSettings": {
      "type": "object",
      "description": "Details of the video settings,
        resolution included"
    },
    "audioSettings": {
      "type": "object",
      "description": "Details of the audio settings,
        resolution included"
    }
  }
},
"required": ["type", "selectedMimeType", "recordedMimeType",
  "videoSettings", "audioSettings"]
```

```

},
"settingsForSelectedFile": {
  "description": "Selected file",
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "const": "file"
    }
  },
  "required": ["type"]
},
"settingsEmpty": {
  "description": "No file selected or recorded",
  "type": "object",
  "additionalProperties": false,
  "properties": {}
},
"videoAttachment": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",
      "const": 7
    },
    "filename": {
      "type": "string",
      "description": "Name of the file as attached by the user"
    },
    "file": {
      "type": "string",
      "description": "Name of the file stored in the shared
        folder"
    }
  },
  "required": ["identifier", "type", "filename", "file"]
},
"fileAttachment": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string",
      "description": "Identifier of the template component"
    },
    "type": {
      "type": "integer",
      "description": "Type of the template component",

```

```
        "const": 8
      },
      "filename": {
        "type": "string",
        "description": "Name of the file as attached by the user"
      },
      "file": {
        "type": "string",
        "description": "Name of the file stored in the shared
          folder"
      },
    },
    "required": ["identifier", "type", "filename", "file"]
  },
  "photoAttachment": {
    "type": "object",
    "properties": {
      "identifier": {
        "type": "string",
        "description": "Identifier of the template component"
      },
      "type": {
        "type": "integer",
        "description": "Type of the template component",
        "const": 9
      },
      "filename": {
        "type": "string",
        "description": "Name of the file as attached by the user"
      },
      "file": {
        "type": "string",
        "description": "Name of the file stored in the shared
          folder"
      },
    },
    "required": ["identifier", "type", "filename", "file"]
  },
  "pictureAttachment": {
    "type": "object",
    "properties": {
      "identifier": {
        "type": "string",
        "description": "Identifier of the template component"
      },
      "type": {
        "type": "integer",
        "description": "Type of the template component",
        "const": 11
      },
      "filename": {
        "type": "string",
```

```

        "description": "Name of the file as attached by the user"
    },
    "file": {
        "type": "string",
        "description": "Name of the file stored in the shared
        folder"
    },
    },
    "required": ["identifier", "type", "filename", "file"]
},
"componentValue": { "anyOf": [
    { "$ref": "#/definitions/textField" },
    { "$ref": "#/definitions/numericField" },
    { "$ref": "#/definitions/dateField" },
    { "$ref": "#/definitions/checkBox" },
    { "$ref": "#/definitions/video" },
    { "$ref": "#/definitions/videoAttachment" },
    { "$ref": "#/definitions/fileAttachment" },
    { "$ref": "#/definitions/photoAttachment" },
    { "$ref": "#/definitions/pictureAttachment" }
]},
},
"properties": {
    "templateIdentifier": {
        "description": "Identifier of the template used for the client data
        part of the task",
        "type": "string"
    },
    "date": {
        "description": "Date of task execution in format dd.MM.yyyy",
        "type": "string"
    },
    "fields": {
        "description": "List of values captured by the client",
        "type": "array",
        "items": { "$ref": "#/definitions/componentValue" }
    },
    },
    "required": ["templateIdentifier", "date", "fields"]
}

```

■ Obrázek C.3 Schéma souboru taskData.json

```
{
  "templateIdentifier": "standardClientData",
  "name": "Josef Klient",
  "fields": [
    {
      "identifier": "dateOfBirth",
      "type": 4,
      "value": "29.02.2016"
    },
    {
      "identifier": "weight",
      "type": 2,
      "value": "84"
    },
    {
      "identifier": "height",
      "type": 2,
      "value": "180"
    }
  ],
  "userId": 79,
  "username": "test03@telecam.cz"
}
```

■ **Obrázek C.4** Příklad souboru clientData.json

```

{
  "templateIdentifier": "exerciseArmFullRange",
  "date": "31.03.2021",
  "fields": [
    {
      "identifier": "chairHeight",
      "type": 2,
      "value": "120"
    },
    {
      "settings": {
        "videoSettings": {
          "exposureCompensation": 0,
          "exposureTime": 312.5,
          "facingMode": "user",
          "groupId": "1c7205de1cda479093968380364acd54c4ef6c59d79b1e6ebbde8e4b0226164f",
          "aspectRatio": 1.25,
          "deviceId": "7822f54806844c62f785f6da93eb5252802585acae316ece54222a276ce0fa9f",
          "colorTemperature": 4600,
          "saturation": 64,
          "frameRate": 30,
          "brightness": 0,
          "contrast": 50,
          "width": 1280,
          "resizeMode": "crop-and-scale",
          "sharpness": 50,
          "whiteBalanceMode": "continuous",
          "exposureMode": "continuous",
          "height": 1024
        },
        "selectedMimeType": "video/webm; codecs=h264",
        "type": "camera",
        "audioSettings": {
          "channelCount": 1,
          "echoCancellation": true,
          "autoGainControl": true,
          "noiseSuppression": true,
          "groupId": "b50b166081420844f5109a8b148793b74e4f5cdd44d87b7a965a4a624ad85a8b",
          "latency": 0.01,
          "sampleSize": 16,
          "deviceId": "default",
          "sampleRate": 48000
        },
        "recordedMimeType": "video/x-matroska; codecs=avc1,opus"
      },
      "identifier": "record",
      "filename": "video.webm",
      "file": "144.data",
      "type": 6
    }
  ]
}

```

■ Obrázek C.5 Příklad souboru taskData.json

```
{
  "templateIdentifier": "!exerciseResult",
  "date": "31.03.2021",
  "fields": [
    {
      "identifier": "description",
      "type": 10,
      "value": "Cvičil jste vesměs <b>správně</b>"
    },
    {
      "identifier": "rightPosition",
      "filename": "rightposition.jpg",
      "file": "146.data",
      "type": 8
    },
    {
      "identifier": "processedVideo",
      "filename": "processedvideo.webm",
      "file": "145.data",
      "type": 6
    }
  ]
}
```

■ **Obrázek C.6** Příklad souboru resultData.json

Návrh specifikace

V průběhu analýzy požadavků vznikl návrh specifikace aplikace jako výstup diskuse se zadavatelem. Hlavním cílem dokumentu bylo požadavky zadavatele shrnout a vzájemně se ujistit o jejich správném porozumění, definovat hlavní rysy aplikace a určit očekávané výstupy práce.

Text návrhu specifikace byl přípravou aplikace překonán a s výjimkou jazykových změn a formátování do nové šablony nebyl aktualizován. Přestože výsledná aplikace z návrhu specifikace vychází a v hlavních rysech jej respektuje, některé dílčí funkčnosti jsou implementovány odlišně. Významné rozdíly jsou uvedeny v tabulce D.1.

■ **Tabulka D.1** Změny implementace oproti návrhu specifikace

č.	Popis
1	Klient může opětovně odeslat přílohy úlohy jako součást nové úlohy bez jejich stažení.
2	Upozornění návazné aplikace na pořízenou úlohu není implementováno.
3	Přílohy dat o zpracování musí být uloženy ve sdílené složce a nemohou být na jiném webovém serveru. Tím ovšem není dotčena možnost uvést odkaz na zdroj na jiném webovém serveru do html textu odpovědi – v tom případě bude například zobrazen obrázek z jiného webového serveru.
4	Terapeut nemůže měnit data ze zpracování úlohy.
5	Správce a terapeut mohou měnit údaje úlohy, i když ještě není zpracovaná. Tím ovšem vznikne nová úloha ke zpracování.
6	Terapeut může změnit údaje klienta aniž by se jako tento klient přihlásil.
7	Správce může změnit údaje uživatele aniž by se jako tento uživatel přihlásil.
8	Terapeut a správce při změně stavu uživatele nebo klienta musí otevřít jeho údaje a nemohou změnu provést přímo z přehledu uživatelů nebo klientů.
9	Správce nemůže klientovi odeslat link na úlohu aniž by se předtím přihlásil jako terapeut.
10	Nad rámec požadavků je implementován aplikační log přístupný z aplikace.
11	Nad rámec minimálních požadavků jsou jako vícejazyčné provedeny všechny stránky aplikace.
12	Není implementovaná speciální úloha k ověření parametrů a formátu nahrávání videa a její předávání do návazné aplikace formou http requestu.
13	Pokud jsou k terapeutovi navázány úlohy klientů, není možné je smazat.
14	Logický a fyzický datový model byl změněn.
15	Odhlášení uživatele ukončí trvalé přihlášení pouze v aktuálním prohlížeči. K odhlášení v ostatních prohlížečích postačí změna hesla uživatele.
16	v čase mezi specifikací a implementací došlo k umožnění nahrávání videa ve formátu mp4 na zařízeních s operačním systémem iOS pomocí prohlížeče; tato funkčnost byla v aplikaci implementována.
17	Pokud má zařízení k dispozici více kamer, mikrofonů nebo reproduktorů, aplikace umožňuje jejich výběr pro pořízení a přehrání videonahrávky.
18	Číselná pole ve formuláři mohou mít rozsah od – do.
19	Rozsah dostupných komponent pro šablony úloh byl rozšířen o komponentu pro fotografie a o komponentu pro obrázky.

D.1 Obsah dokumentu

Tento dokument je vytvořen v rámci předmětu BIK-BPR (Bakalářský projekt). Dokument shrnuje obsah pravidelných diskusí autora práce s odborným konzultantem Ing. Janem Hejdou, Ph.D., a vymezuje obsah práce plánované pro následující předmět BIK-BAP (Bakalářská práce).

Práce vytvořená v rámci předmětu BIK-BAP se může lišit v implementačních detailech. Jiné změny podléhají schválení zadavatele a řešitele.

D.1.1 Seznam verzí

Seznam verzí je uveden v tabulce D.2.

D.1.2 Otevřené body

Otevřené body jsou uvedeny v tabulce D.3.

■ **Tabulka D.2** Seznam verzí

Verze	Datum	Poznámky
0.1	5. 12. 2020	První verze zpracovaná z průběžných poznámek
0.2	7. 12. 2020	Doplnění druhého otevřeného bodu
0.3	8. 12. 2020	Připíchnutí obrázků a tabulek na místo odkazu a zapracování dvou otevřených bodů.
0.4	29. 12. 2020	Zpracování připomínek J. Hejdy z 15. 12. 2020.

■ **Tabulka D.3** Otevřené body

Bod	Název	Popis
	Žádné zbylé otevřené body	

D.2 Zadání práce

Zadání práce tak, jak bylo formulováno:

„Cílem bakalářské práce je navrhnout a implementovat databázi a webovou aplikaci/rozhraní pro telerehabilitační systém. Backend webového rozhraní bude implementován v Javě. Přes rozhraní bude možné nahrát objemná data (například video) a bude možnost vyčkat na výsledky vypočtené výpočetním serverem. V rámci práce prozkoumejte současné softwarové technologie pro danou úlohu. Navrhněte vhodný databázový model pro danou problematiku. Navrhněte webový server, backend a frontend webové aplikace. Navrženou databázi i aplikaci implementujte a ověřte její funkčnost. Vytvořte testy ověřující funkčnost webové aplikace. Zdokumentujte implementovanou databázi a webovou aplikaci a vytvořte manuál k obsluze webové aplikace.“

D.3 Předmět práce (manažerské shrnutí)

Předmětem práce je návrh a dodávka webové aplikace s dokumentací podle zadání práce v upřesněném rozsahu uvedeném v tomto dokumentu.

Aplikace je implementovaná v Javě 11 s využitím frameworku Spring Boot. Front end aplikace je vytvořen v HTML 5 s využitím Javascriptu a frameworku Bootstrap.

Aplikace umožňuje přihlášení uživatelů následujících typů:

- Správce
- Terapeut
- Klient

Aplikace umožňuje definovat úlohy a klient tyto úlohy provádí. Odevzdaná úloha je uložena na serveru do nastavené složky, kde je zpracována jinou aplikací. Výsledky tohoto zpracování jsou touto aplikací přijaty a zobrazeny klientovi.

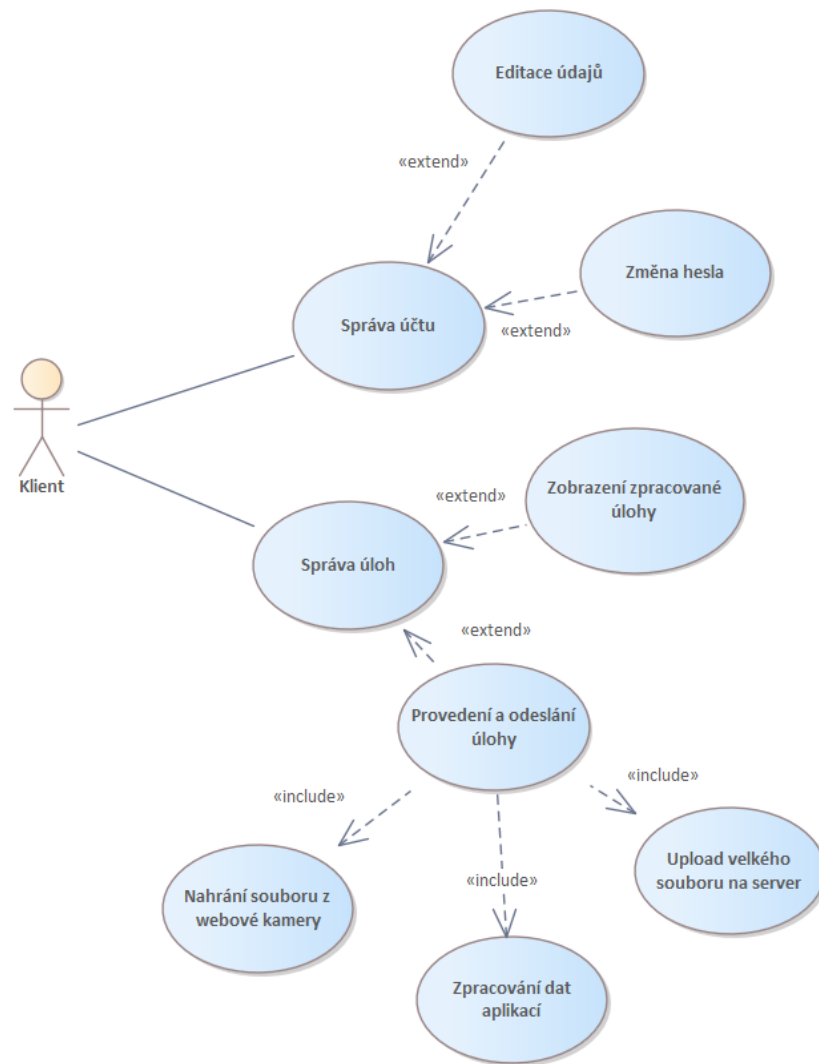
Pracovní název aplikace je **TeleCam**.

D.4 Funkční požadavky

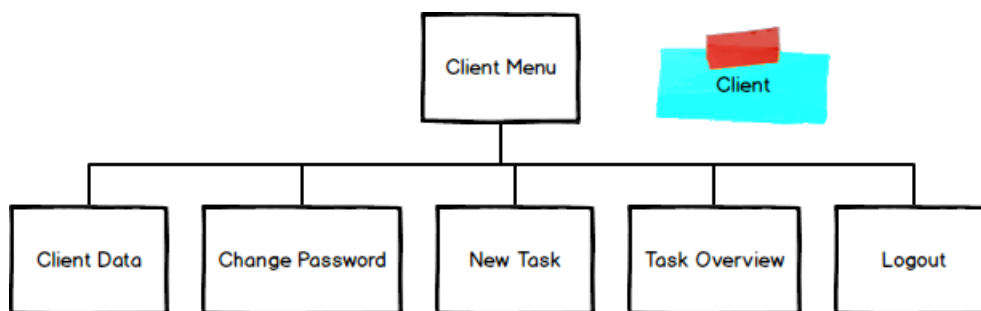
D.4.1 Případy užití

D.4.1.1 Případy užití klienta

Případy užití klienta jsou uvedeny na obrázku D.1. Případům užití odpovídá struktura aplikačního menu na obrázku D.2.



■ Obrázek D.1 Případy užití klienta



■ Obrázek D.2 Aplikační menu klienta

D.4.1.1.1 Editace údajů

Na obrazovce na obrázku D.3 může klient, jeho terapeut nebo správce editovat údaje klienta. Tyto údaje jsou předvyplněny na začátku plnění úlohy. Některé údaje, jako například jméno, nemůže klient sám editovat.

■ Obrázek D.3 Editace údajů klienta

D.4.1.1.2 Změna hesla

Na obrazovce na obrázku D.4 může uživatel změnit svoje heslo. Při vkládání nového hesla bude aplikace indikovat sílu zvoleného hesla a nepovolí použití slabého hesla.

D.4.1.1.3 Provedení a odeslání úlohy

Uživatel, který je aktivní a může vykonávat úlohy, vybere typ úlohy a vykoná úlohu podle postupu uvedeného níže.

Na obrazovce na obrázku D.5 vybere úlohu k plnění. Ve výběru jsou všechny aktuálně platné typy úloh.

■ Obrázek D.4 Změna hesla

■ Obrázek D.5 Výběr typu úlohy

Na následující obrazovce na obrázku D.6 plní úlohu. Osobní údaje jsou předvyplněny podle osobních údajů klienta a metadata o úloze mohou být předvyplněna podle posledního plnění

úlohy stejného typu týmž klientem.

Jednotlivé součásti úlohy jsou povinné nebo nepovinné. Aplikace validuje pouze přítomnost datového obsahu u povinných položek.

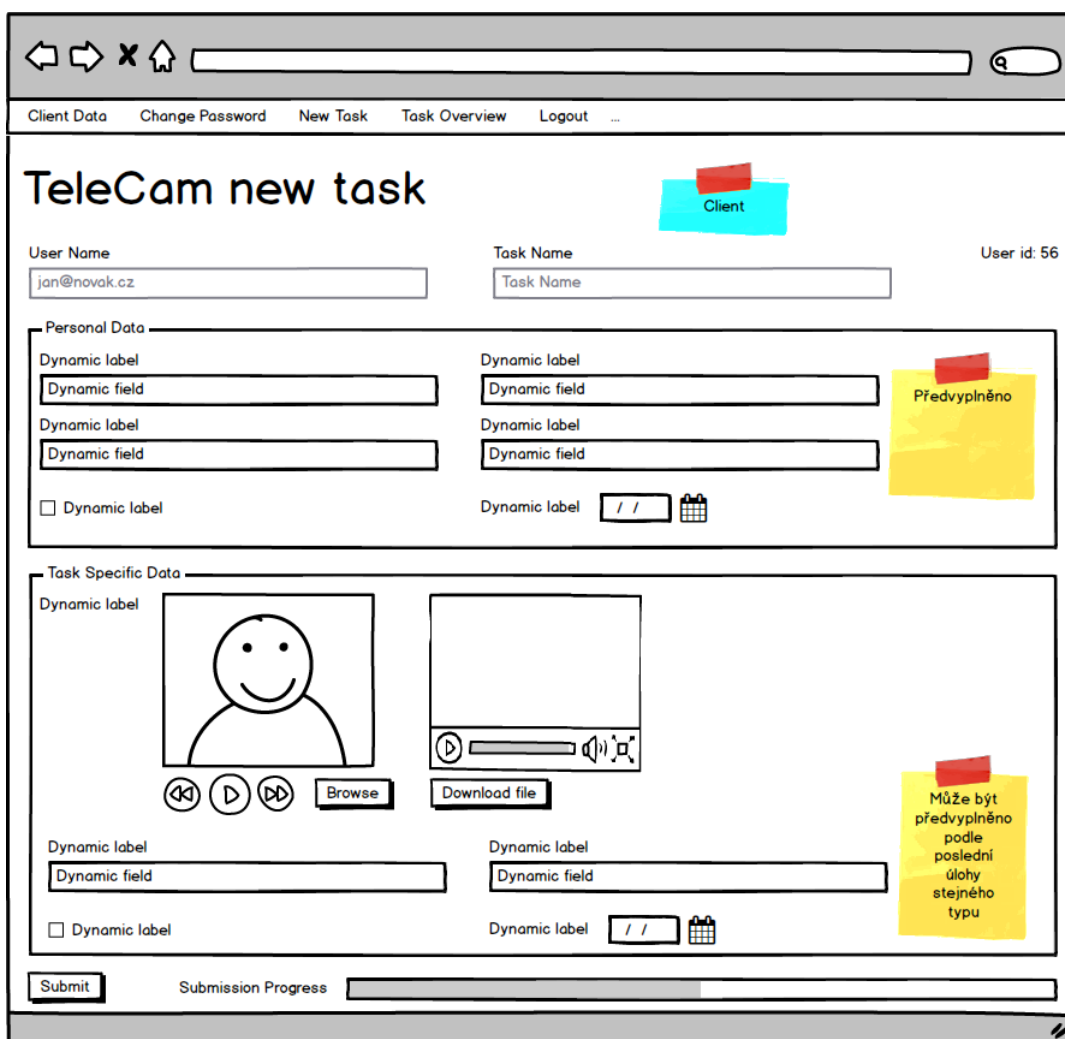
Součástí úlohy může být pořízení videonahrávky. Uživatel soubor vytvoří webkamerou nebo nahraje z filesystému. Při nahrání souboru z webkamery jsou využita API prohlížeče k vytvoření souboru k nahrání na server a klient má k dispozici tlačítka pro ovládání pořízení a přehrání nahrávky. Klient si soubor před odesláním může přehrát a eventuálně smazat a nahrát znovu.

Uživatel si soubor prohlédne a spolu s metadaty jej může odeslat na server. Odeslaný soubor a metadata nemůže měnit, soubor ale může stáhnout a znovu odeslat jako součást jiné úlohy.

Při odeslání úlohy jsou aktualizována osobní data klienta a soubory a metadata úlohy jsou odeslána na server k dalšímu zpracování.

Každá úloha bude mít vlastní složku se soubory ke zpracování a s výsledky zpracování.

Aplikace upozorní návaznou aplikaci na soubory čekající na zpracování.



■ Obrázek D.6 Plnění úlohy

Výstupem automatického zpracování úlohy je soubor json, který může obsahovat následující objekty:

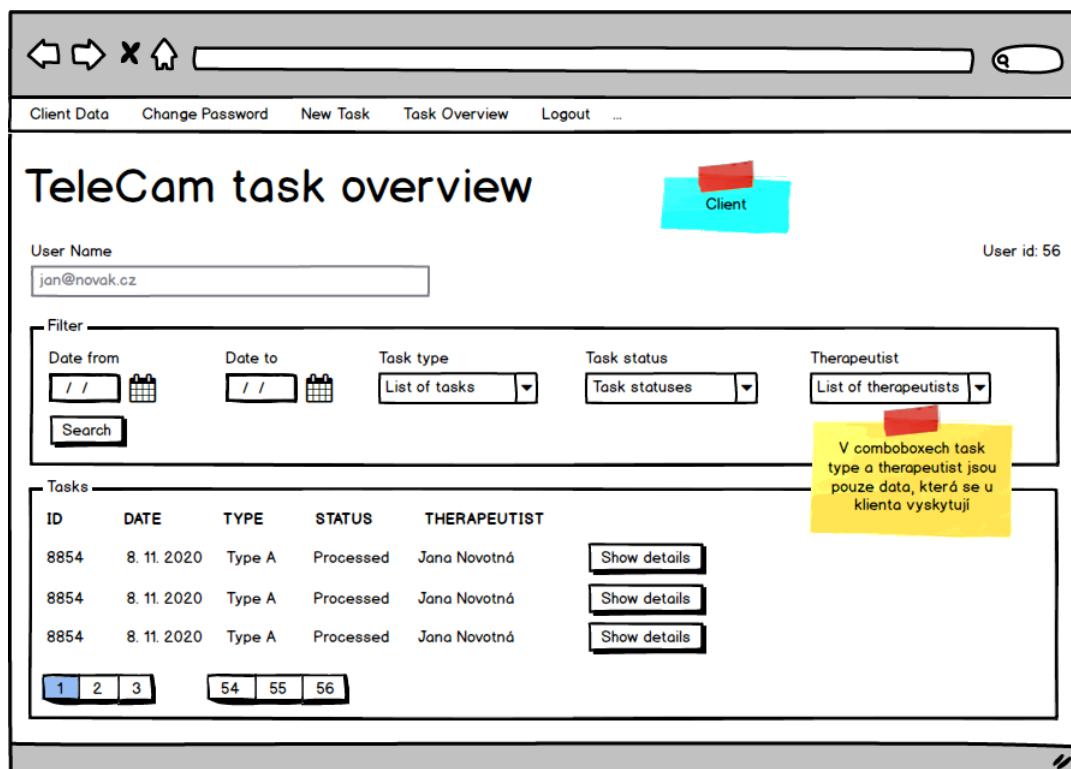
- Zobrazení textového pole

- Odkaz na obrázek
- Odkaz na video
- Obecně jakýkoliv HTML text ke vložení do stránky

Soubory mohou být uloženy na jiné URL adrese nebo ve složce náležející zpracovávané úloze.

Příklad:

```
{
  "name":
  {
    "type": "text",
    "value": "Správné cvičení"
  },
  "overall":
  {
    "type": "image",
    "value": "overall10001.png"
  }
}
```



■ Obrázek D.7 Přehled úloh

D.4.1.1.4 Zobrazení úlohy

Klient si může zobrazit přehled úloh pomocí obrazovky na obrázku D.7. Obrazovka umožňuje filtrování a stránkování.

Klient si také může nechat zobrazit detail provedené úlohy. Detail se zobrazí na obrazovce na obrázku D.8. Pokud je úloha již zpracovaná, jsou zobrazeny také informace ze zpracování úlohy.

Client Data Change Password New Task Task Overview Logout ...

TeleCam task detail

Client

Client Name: jan@novak.cz Task Name: Task Name User id: 56 Task id: 8855

Personal Data

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label

Dynamic label: // [Calendar icon]

Task Specific Data

Dynamic label: [Video player]

Download file

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label

Dynamic label: // [Calendar icon]

Task Response Data

Dynamic label: Dynamic field

Dynamic label: Dynamic field

■ Obrázek D.8 Detail úlohy

D.4.1.1.5 Zpracování dat aplikací

Aplikace na serveru uloží data úlohy do vyhrazené složky, kde jsou tato data zpracována návaznou aplikací. Tato návazná aplikace umístí do stejné složky výstupní soubory, které jsou aplikací klientovi zobrazeny při zobrazení zpracované úlohy.

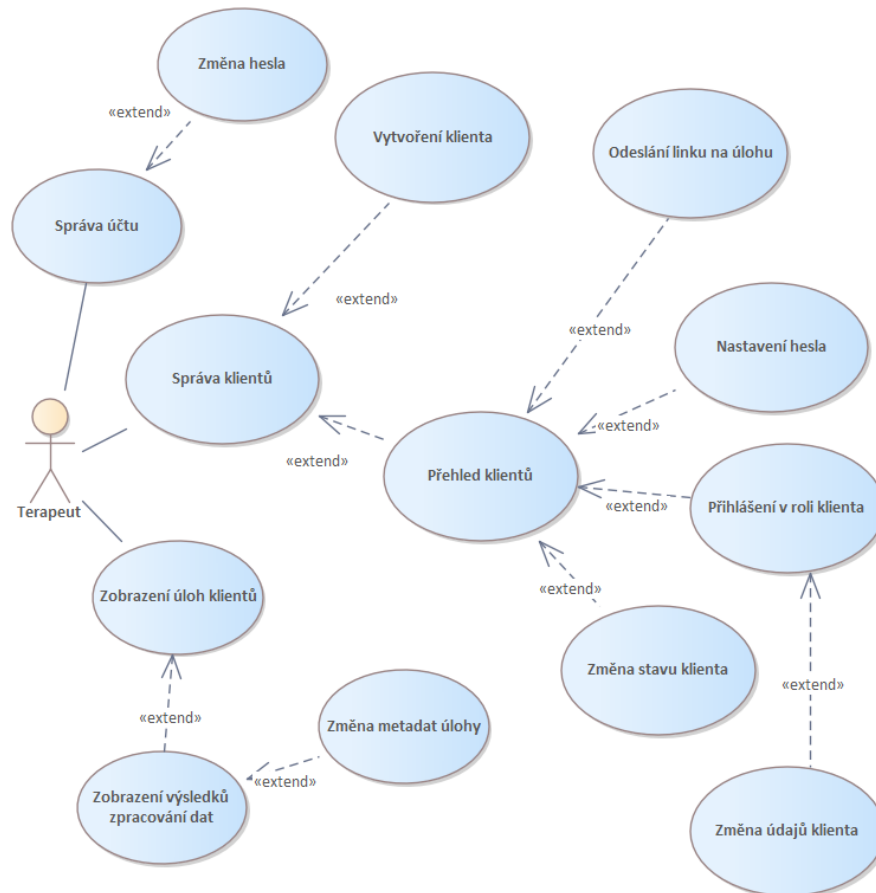
Odevzdaná úloha tedy prochází minimálně následujícími stavy:

- Čeká na zpracování
- Zpracovaná

Jakmile je úloha návaznou aplikací zpracována, je považována za zpracovanou a jako taková je zpřístupněna klientovi – aplikace nečeká na doplnění dalších polí terapeutem.

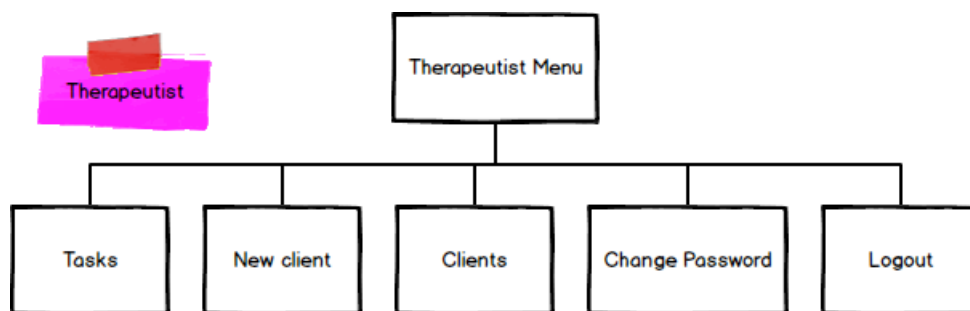
D.4.1.2 Případy užití terapeuta

Případy užití terapeuta jsou uvedeny na obrázku D.9.



■ **Obrázek D.9** Případy užití terapeuta

Případům užití odpovídá struktura aplikačního menu uvedená na obrázku D.10.



■ Obrázek D.10 Aplikační menu terapeuta

D.4.1.2.1 Změna hesla

Změna hesla terapeuta je stejná jako v případě klienta a je popsána v kapitole D.4.1.1.2.

■ Obrázek D.11 Vytvoření nového klienta

D.4.1.2.2 Vytvoření klienta

Terapeut může vytvořit nového klienta pomocí obrazovky na obrázku D.11. Po vytvoření nového klienta aplikace odešle klientovi link na vytvoření prvního hesla. Obrazovka pro vytvoření prvního hesla vychází z obrazovky na obrázku D.4, ale místo stávajícího hesla je použit link z e-mailu.

D.4.1.2.3 Přehled klientů

Terapeut může zobrazit přehled všech svých klientů na obrazovce na obrázku D.12. Obrazovka umožňuje hledání a podporuje stránkování. Terapeut může klientovi poslat link na vykonání nové úlohy vybraného typu, změnit jeho stav, vyvolat nastavení hesla a může se jako klient přihlásit – to mu umožňuje měnit údaje klienta a provádět všechny další operace jako klient.

Hledání klienta je implementováno pomocí textového hledání, protože počet klientů může být značný.

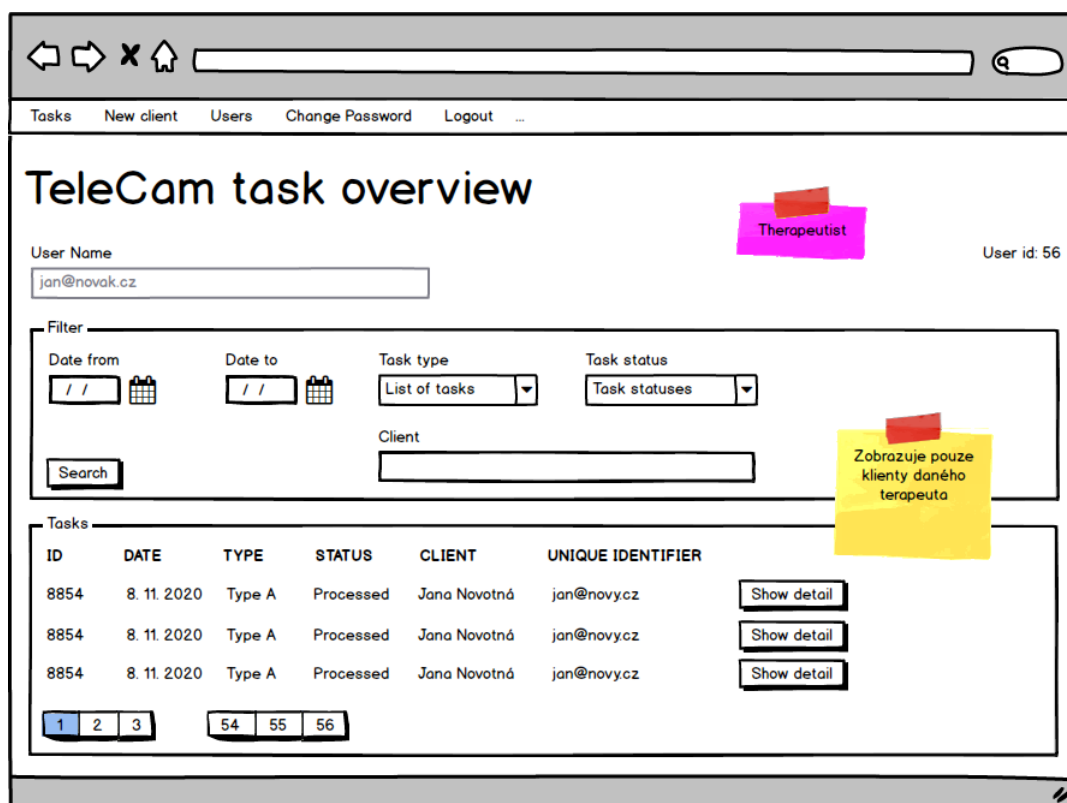
The screenshot shows a web browser window with a navigation menu (New User, Users, Tasks, Task Management, System Management, Logout) and a search bar. The main content area is titled 'TeleCam clients' and includes a 'Therapist' profile icon. Below the title is a 'User Name' field containing 'jan@novak.cz' and a 'User id: 56' label. A 'Filter' section contains fields for 'Date from', 'Date to', 'User Type' (set to 'List of tasks'), 'User Status' (set to 'Task statuses'), and a 'Full text' search box. A 'Search' button is located below the filter fields. The main part of the interface is a table titled 'Users' with columns for ID, DATE, TYPE, STATUS, NAME, and UNIQUE IDENTIFIER. Each row in the table has three buttons: 'Password reset', 'New task', and 'Log as'. The table shows three entries for 'Jana Novotná' with different statuses (Active, Blocked, Active). At the bottom of the table, there are pagination controls showing '1 2 3' and '54 55 56'.

ID	DATE	TYPE	STATUS	NAME	UNIQUE IDENTIFIER			
8854	8. 11. 2020	Client	Active	Jana Novotná	jana@novotna.c	Password reset	New task	Log as
8854	8. 11. 2020	Client	Blocked	Jana Novotná	jana@novotna.c	Password reset	New task	Log as
8854	8. 11. 2020	Client	Active	Jana Novotná	jana@novotna.c	Password reset	New task	Log as

■ Obrázek D.12 Přehled klientů

D.4.1.2.4 Zobrazení úlohy

Terapeut může zobrazit seznam všech úloh svých klientů pomocí obrazovky na obrázku D.13. Obrazovka umožňuje filtrování a stránkování. Filtr na klienta je proveden pomocí textového hledání, protože počet klientů může být značný.



■ **Obrázek D.13** Přehled úloh klientů terapeuta

Detail úlohy zobrazené na seznamu úloh může terapeut zobrazit na obrazovce na obrázku D.14. Terapeut může změnit data klienta, metadata úlohy nebo data ze zpracování úlohy. Při změně dat klienta nebo metadata úlohy dojde po upozornění k opakovanému zpracování úlohy.

Tasks New client Users Change Password Logout ...

TeleCam task detail

Therapist Admin

User Name: Jan Novák User id: 56 Task id: 8855

Client Name: Jan Novák Task Name: Task Name Therapist: Therapist

Personal Data

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label

Dynamic label: // [Calendar icon]

Task Specific Data

Dynamic label: [Video player controls]

Download file

Uložení změny vyvolá po upozornění nové zpracování úlohy

Dynamic label: Dynamic field

Dynamic label: Dynamic field

Dynamic label

Dynamic label: // [Calendar icon]

Task Response Data

Dynamic label: Dynamic field

Dynamic label: Dynamic field

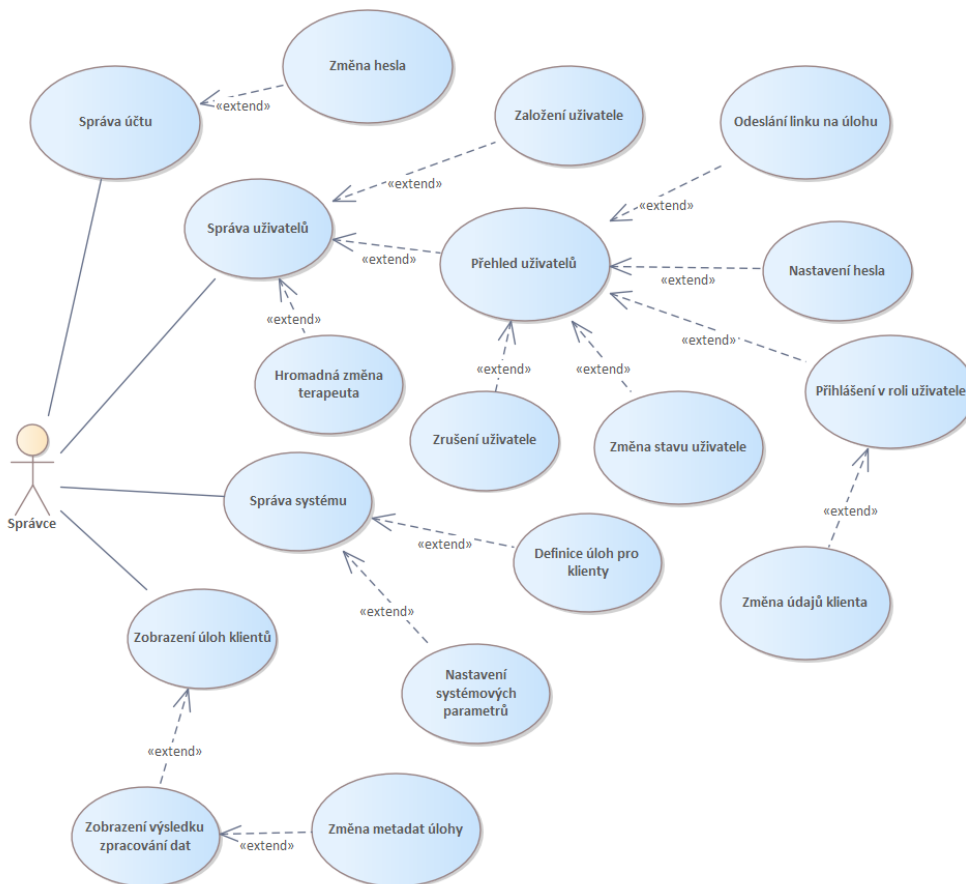
JSON DATA

Save changes

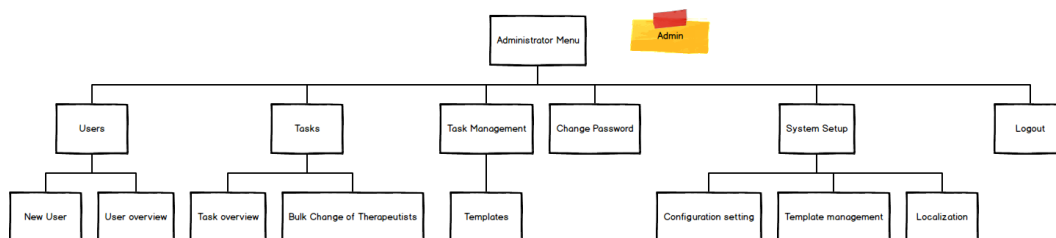
■ Obrázek D.14 Zobrazení úlohy terapeutem

D.4.1.3 Případy užití správce

Případy užití správce jsou uvedeny na obrázku D.15. Případům užití odpovídá struktura aplikačního menu uvedená na obrázku D.16.



■ **Obrázek D.15** Případy užití správce



■ **Obrázek D.16** Aplikační menu správce

D.4.1.3.1 Změna hesla

Změna hesla terapeuta je stejná jako v případě klienta a je popsána v kapitole D.4.1.1.2.

D.4.1.3.2 Založení uživatele

Založení uživatele je obdobné jako v případě založení klienta terapeutem a je popsáno v kapitole D.4.1.2.2. Na rozdíl od terapeuta může správce založit také jiného správce nebo terapeuta.

D.4.1.3.3 Přehled uživatelů

Správce může zobrazit přehled všech uživatelů na obrazovce na obrázku D.17. Obrazovka umožňuje hledání a podporuje stránkování. Správce může uživateli poslat link na vykonání nové úlohy vybraného typu, změnit jeho stav, vyvolat nastavení hesla a může se jako uživatel přihlásit – to mu umožňuje měnit údaje klienta a provádět všechny další operace jako uživatel. Může uživatele také odstranit.

Hledání uživatele je implementováno pomocí textového hledání, protože počet uživatelů může být značný.

Users Tasks Task Management Change password System Setup Logout ...

TeleCam users

User Name: john@doe.cz User id: 56

Filter

Date from: / / Date to: / / User Type: List of tasks User Status: Task statuses Full text: Search

Therapist: List of therapists

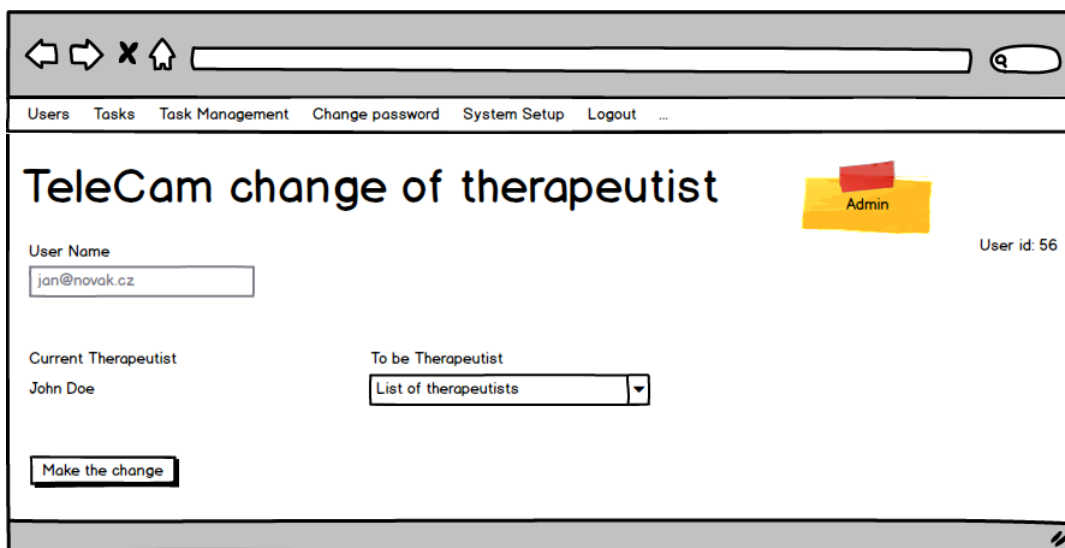
ID	DATE	TYPE	STATUS	NAME	THERAPEUTIST	UNIQUE IDENTIFIER
8854	8. 11. 2020	Client	Block	Jana Novotná	Jan Novák	Password reset Log as Delete
8854	8. 11. 2020	Therapist	Unblock	Jana Novotná		Password reset Log as Delete
8854	8. 11. 2020	Administrator	Block	Jana Novotná		Password reset Log as Delete

1 2 3 54 55 56

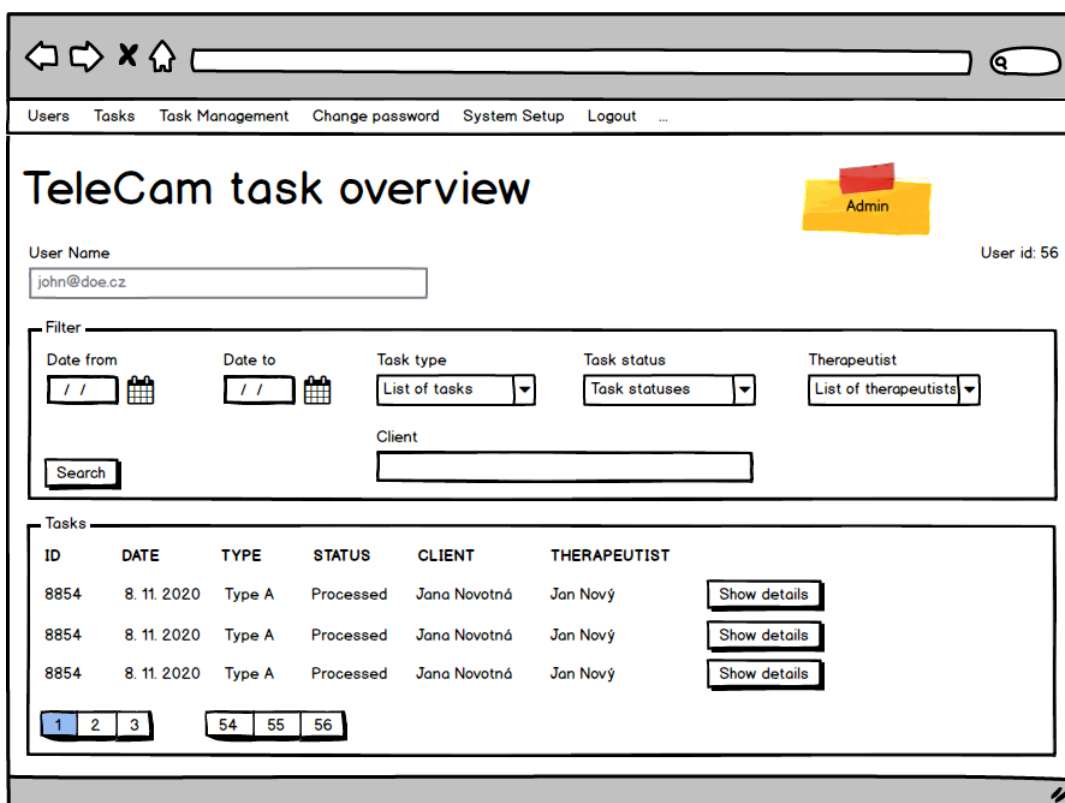
■ Obrázek D.17 Přehled uživatelů

D.4.1.3.4 Hromadná změna terapeuta

Správce může změnit terapeuta pro skupinu uživatelů se stejným terapeutem pomocí obrazovky na obrázku D.18.



■ Obrázek D.18 Hromadná změna terapeuta



■ Obrázek D.19 Přehled úloh

D.4.1.3.5 Zobrazení úlohy

Správce může zobrazit seznam všech úloh klientů pomocí obrazovky na obrázku D.19. Obrazovka umožňuje filtrování a stránkování. Filtrování na klienta je provedeno pomocí textového hledání,

protože počet klientů může být značný.

Zobrazení detailu úlohy je shodné jako v případě zobrazení terapeutem a je popsáno v kapitole D.4.1.2.4.

D.4.1.3.6 Definice úloh pro klienty

Správce definuje dostupné úlohy pomocí obrazovky na obrázku D.20. Obrazovka podporuje vyhledávání a stránkování. Úlohy jsou dostupné všem klientům bez rozdílu.

Právě jedna úloha je označena jako klientská data a definuje osobní data klientů. Tuto úlohu klient nemůže vybrat k provedení.

Speciálním případem úlohy je úloha sloužící k ověření formátu videa ke zpracování. Předpokládáme, že tato úloha bude identifikována pomocí nastavení systémových parametrů aplikace.

The screenshot shows a web application interface for managing task templates. The main heading is 'TeleCam templates'. Below the heading, there is a search filter section with fields for 'Date from', 'Date to', and 'Full text', and a 'Search' button. The main content area displays a table of task templates with the following columns: ID, DATE, STATUS, NAME, ORDER, and IS CLIENT DATA. The table contains three rows of data. Below the table, there are pagination controls showing '1', '2', '3' and '54', '55', '56'. At the bottom, there is a button labeled 'Add a new task template'.

ID	DATE	STATUS	NAME	ORDER	IS CLIENT DATA				
88	8. 11. 2020	Active	Template 1	23	<input checked="" type="checkbox"/>	Move up	Move down	OPEN	Delete
89	8. 11. 2020	Active	Template 1	24	<input type="checkbox"/>	Move up	Move down	OPEN	Delete
90	8. 11. 2020	Active	Template 1	25	<input type="checkbox"/>	Move up	Move down	OPEN	Delete

■ Obrázek D.20 Definice úloh pro klienty

Definici úlohy je možné otevřít a editovat. Nepoužité šablony úloh je možné odstranit.

Definice úlohy je složena z komponent. Aplikace umožňuje používat komponenty následujících typů:

- Statický text
- Číselné pole
- Textové pole
- Datum

- Checkbox
- Videonahrávka
- Soubor - Soubory mohou být typu video (avi, mp4) nebo jiné (jpg, txt, pdf, doc, docx, zip)

Při přidávání komponent do definice úlohy nebo při jejich editaci se v poli „Component Settings“ zobrazí pole specifická pro jednotlivé typy komponent tak, jak se naznačeno na obrázku D.21.

Podle definice úlohy je možné vytvořit kopii s jiným názvem.

The screenshot shows a web interface for editing a TeleCam template. At the top, there is a navigation menu with items: Users, Tasks, Task Management, Change password, System Setup, Logout. The page title is "TeleCam template detail" and the user is identified as "Admin" with "User id: 56".

Form fields include:

- User Name: Jan Novák
- Template ID: 64
- Template Name: My template

The "Template definition" section contains a table with the following data:

ID	COMPONENT TYPE	LABEL	ORDER	MANDATORY	PRE-FILL	
88	Text box	Váha / Weight	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Up Down Edit Delete
89	Video	Cvičení 1	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Up Down Edit Delete
90	Text box	Aktuální počasí	3	<input type="checkbox"/>	<input type="checkbox"/>	Up Down Edit Delete

Below the table, there is an "Add component" section with a dropdown menu "List of components" and buttons "Add to top" and "Add to bottom".

The "Component settings" section contains a redacted area labeled "Component specific settings".

At the bottom, there are "Save" and "Copy as new" buttons.

■ Obrázek D.21 Definice úlohy

D.4.1.3.7 Nastavení systémových parametrů

Správce nastavuje systémové parametry aplikace na obrazovce na obrázku D.22.

D.4.2 Struktura uživatelů

Správce nastavuje technickou konfiguraci aplikace, spravuje uživatelské přístupy, vztahy mezi terapeuty a klienty a konfiguruje úlohy pro klienty.

Klient provádí úlohy a prostřednictvím aplikace je předává ke zpracování. Klient má přiřazeného svého terapeuta. Klient může nabývat těchto stavů:

- Neaktivní - nemůže se přihlásit
- Aktivní produktivní - může se přihlásit a může provést novou úlohu.
- Aktivní neproduktivní - může se přihlásit, ale nemůže provést novou úlohu.

Terapeut má přístup k úlohám svých klientů.

Uživatel může být smazán. Terapeut může být smazán pouze v případě, kdy nemá přiřazené žádné klienty. Mohou k němu ale být navázány provedené úlohy.

■ Obrázek D.22 Nastavení systému

D.4.3 Nepodporované funkčnosti

V této verzi aplikace nejsou podporovány následující funkčnosti:

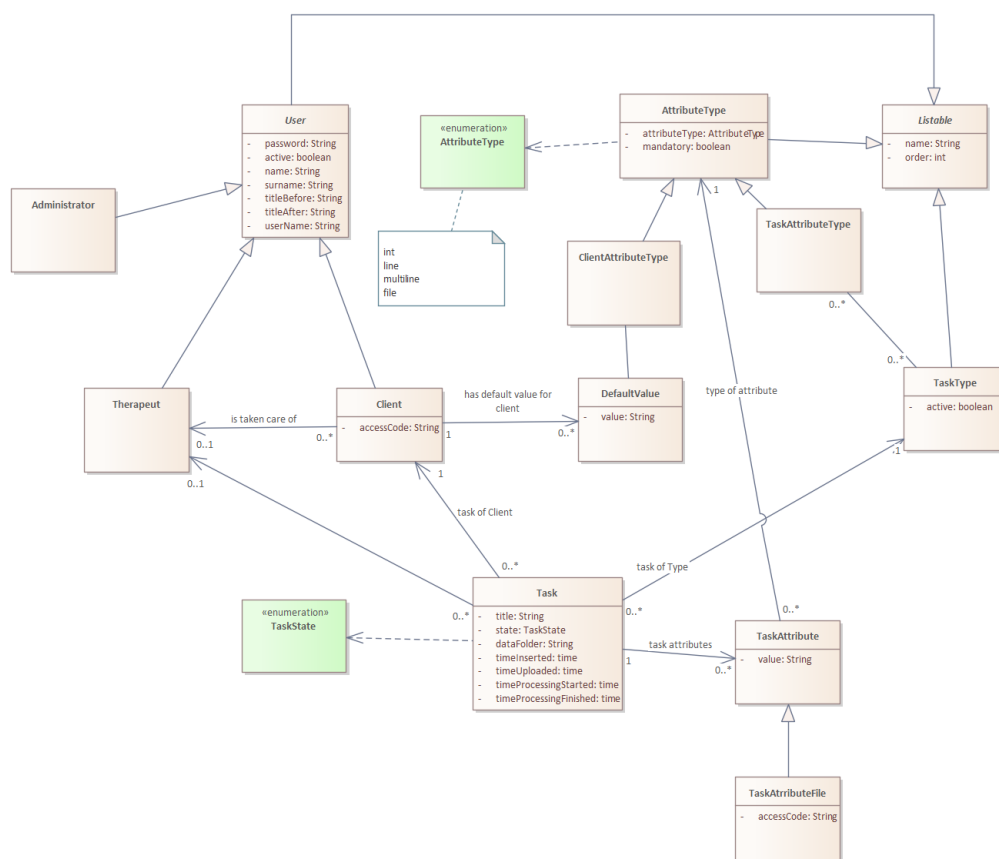
- Aplikace nepodporuje konverzační vlákna pacient – terapeut (obdoba chatu).
- Definice úloh je shodná pro všechny klienty bez rozdílu.
- Výsledkem zpracování úlohy není v této verzi aplikace definice grafu, která by byla aplikací interpretována.

D.4.4 Datový model

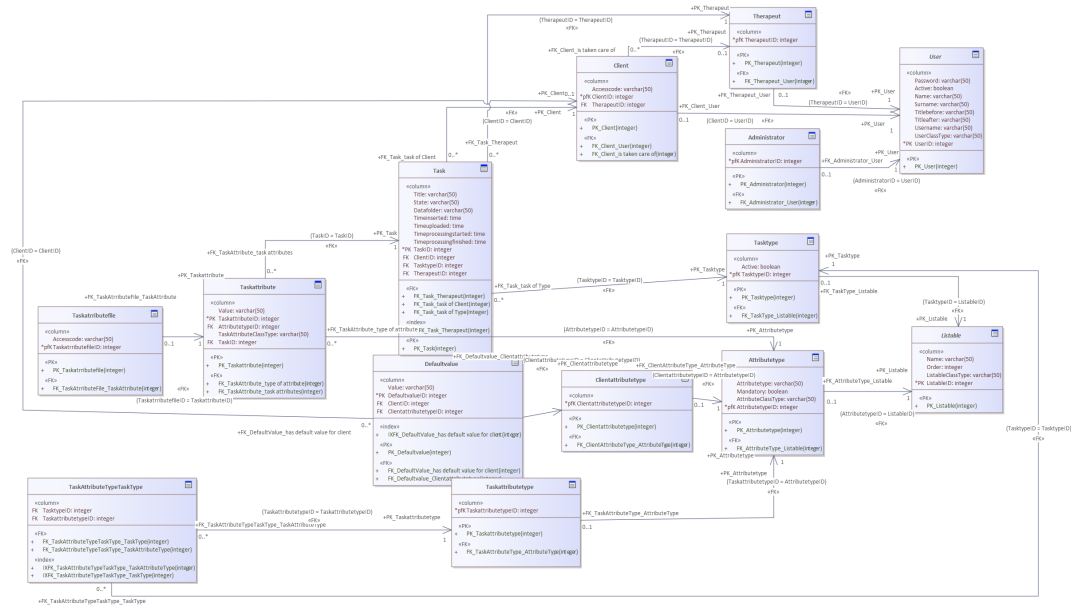
Datový model popsáný v této kapitole obsahuje datové struktury pro klíčové aplikační funkce. Datové struktury pro pomocné a servisní aplikační funkce (jako například systém šablon pro generování uživatelského rozhraní) není v této kapitole popsán a jeho popis bude obsažen až v aplikační analytické dokumentaci.

Logický datový model je na obrázku D.23. Fyzický datový model je na obrázku D.24. Klíčové atributy modelu:

- Vztah pacient – terapeut je n:1, zástupnost terapeutů není řešena.
- Definice úloh se použije při založení nové úlohy pacientem – tím se vytvoří pole, která dále žijí v instanci úlohy. Případná následná změna definice úlohy nemá na obsah odevzdaných úloh vliv. Obdobně je také zajištěno, že součástí úlohy je odkaz na tehdy platného terapeuta.



■ Obrázek D.23 Logický datový model



■ Obrázek D.24 Fyzický datový model

D.5 Technické, servisní a nefunkční požadavky

D.5.1 Bezpečnostní požadavky

Uživatelským identifikátorem je e-mailová adresa. Uživatelé se budou přihlašovat pomocí svého uživatelského identifikátoru a hesla. SMS kód nebo jiný doplňkový údaj nebude používán.

Obrazovka přihlášení uživatele je na obrázku D.25.

Obnova hesla uživatele bude provedena odesláním linku na e-mailovou adresu uživatele, o který uživatel požádá pomocí obrazovky na obrázku D.26. Bezpečnost uživatelského účtu je tedy limitována bezpečností e-mailové služby. Vytvoření nového hesla bude doprovázeno provedením Captcha testu.

Aplikace odešle uživateli link na vytvoření hesla. Obrazovka pro vytvoření hesla vychází z obrazovky na obrázku D.4, ale místo stávajícího hesla je použit link z e-mailu.

Při vkládání nového hesla bude aplikace indikovat sílu zvoleného hesla a nepovolí použití slabého hesla.

Aplikace bude detekovat počet neúspěšných požadavků na přihlášení v čase. V případě překročení určené meze bude přihlášení podmíněno provedením Captcha testu.

Odhlášení uživatele zruší trvalé přihlášení na všech počítačích, kde je nastaveno.

D.5.2 Velikost datových souborů

Aplikace musí být schopna zpracovat datové soubory o velikosti řádově stovek MB. Nahrání takto velkých souborů z uživatelského rozhraní na webový server musí probíhat po částech.

TeleCam login screen

User Name

Password

Permanent login

Login Reset password

Client Admin Therapist

■ Obrázek D.25 Přihlášení uživatele

TeleCam password reset

User Name

Request password reset

Client Admin Therapist

■ Obrázek D.26 Žádost o obnovu hesla

D.5.3 Podporované prohlížeče

Podporované prohlížeče jsou prohlížeče odvozené od Chromu a Firefoxu. Prohlížeče Microsoft Internet Explorer a Edge nejsou aktivně podporovány. Základním podporovaným operačním systémem uživatelského rozhraní je Microsoft Windows.

D.5.4 Komponenta pro nahrávání videa

Na operačních systémech Microsoft Windows a Android je k nahrávání videa použita komponenta MediaRecorder.

Preferovaným kodekem pro kódování videa je (v tomto pořadí) h.264, h.265, vp9 a vp8.

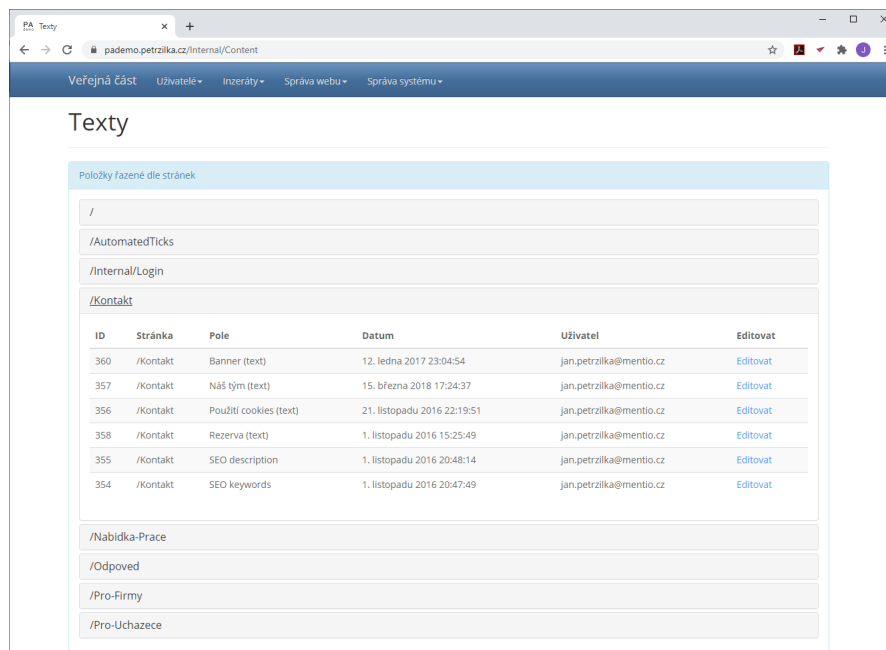
Aplikace se pokusí vynutit detekovaný frame rate alespoň 25 snímků za vteřinu a co největší rozlišení. Realistické rozlišení pro moderní výkonné notebooky je 1280 x 700 bodů.

Na zařízení s iOS bude k nahrání videa použita přednostně vestavěná aplikace, kdy webová aplikace nemá nad parametry nahrávky kontrolu. Předpokládaný video formát je QuickTime. Webová aplikace nebude schopna video v tomto formátu přehrát.

Aplikace bude disponovat funkcí pro zkusmé vytvoření video nahrávky a vyhodnocení jejího formátu.

D.5.5 Výběr webové kamery

Pro nahrávání bude použita výchozí systémová webová kamera. V případě, že v systému bude k dispozici více kamer, nebude webová aplikace nabízet jejich výběr.



■ Obrázek D.27 Ilustrace ke správě šablon

D.5.6 Generování uživatelského rozhraní

Uživatelské rozhraní bude generováno Java aplikací s využitím soustavy šablon obdobně, jako je to vidět na příkladu jiné aplikace na obrázku D.27 (aktuálně veřejně dostupné na adrese

<https://pademo.petrzilka.cz/Internal/Content> – nejedná se o java aplikaci).

D.5.7 E-mailový SMTP server

V produkční fázi bude používán externí (existující) SMTP server. V testovací fázi je možné použít interní SMTP server.

D.5.8 Integrovaní testování

Integrace této webové aplikace a návazné aplikace zpracovávající její data bude probíhat na souborové úrovni. Během testování aplikace budou složky se soubory synchronizovány pomocí rsync.

Malé soubory – při ověřování formátu videa – budou předávány v rámci http requestu, aby odpověď byla co nejrychlejší.

D.5.9 Lokalizace

Aplikace bude provedena v anglickém jazyce nebo v českém a anglickém jazyce. Dokumentace bude v českém jazyce.

D.5.10 Migrace dat

Žádná data z předchozího řešení nebudou migrována.

D.6 Výstupy práce

Budou dodány výstupy uvedené v tabulce D.4.

■ **Tabulka D.4** Výstupy práce

Výstup	Obsah	Popis
1	Analytická dokumentace	Popis datového modelu, popis implementačních možností a zdůvodnění výběru zvoleného řešení, popis implementovaných a provedených testů v výsledky
2	Instalační příručka	Zdrojové kódy aplikace, instalační soubory, popis instalace a administrace aplikace
3	Uživatelská příručka	Popis ovládání aplikace pro všechny skupiny uživatelů
4	Souhlas s užitím díla	Souhlas autora s užitím díla podle potřeb zadavatele
5	Podpora integračních a akceptačních testů	Podpora integračních testů pro další zpracování dat a podpora provedení akceptačních testů
6	Demonstrační instalace aplikace	Funkční aplikace bude zpřístupněna na serveru autora práce

D.7 Předpokládaný harmonogram

Předpokládaný časový harmonogram je uveden v tabulce D.5.

■ **Tabulka D.5** Harmonogram práce

Milník	Datum	Popis
1	13. 10. 2020	Zadání práce (dokončeno, téma schváleno 18. 10. 2020)
2	20. 12. 2020	Shoda o funkčnosti a struktuře aplikace
3	31. 12. 2020	Základní části aplikace mají základní funkčnost (server běží, front-end aplikace je generován a zobrazován).
4	21. 2. 2021	Aplikace funguje v rozsahu hlavních případů užití (přihlášení klienta, provedení úkolu, nahrání dat na server, stažení výsledku)
5	21. 3. 2021	Aplikace připravena k akceptačnímu testování v rozsahu všech uživatelských funkcí
6	4. 4. 2021	Dokončení akceptačního testování
7	18. 4. 2021	Dokončení funkční aplikace
8	25. 4. 2021	Dokončení uživatelské příručky
9	2. 5. 2021	Dokončení práce až na drobné dokončovací práce
10	13. 5. 2021	Nejzazší termín odevzdání práce

Obsah přiloženého média

Médium je přiloženo k výtisku práce v listinné podobě a jeho obsahje k dispozici ke stažení na adrese: <https://jan.petrzilka.cz/218-549/medium.zip>

readme.txt	stručný popis obsahu média
demo.txt	pozvánka na předvedení aplikace pro zájemce
thesis.pdf	tato bakalářská práce ve formátu PDF
prezentace.pdf	prezentace bakalářská práce ve formátu PDF
docs	dokumentace
├─ json	definice a příklady výstupních a vstupních souborů
├─ licence	licenční ujednání použitých nástrojů a komponent
├─ thesis	zdrojová forma této bakalářská práce ve formátu L ^A T _E X
sql	create script a insert script pro vytvoření databáze a její úvodní parametrizaci
src	zdrojové soubory aplikace a testů
├─ main	zdrojové soubory aplikace
├─ test	zdrojové soubory testů
pom.xml	konfigurační soubor nástroje Maven
target	výstup překladač aplikace a spustitelný soubor TeleCam.jar

 Kapitola F

Kontaktní údaje

Pokud se na aplikaci chcete podívat, domluvte si předvedení s využitím kontaktních údajů uvedených níže.

Pokud máte jakékoliv připomínky nebo dotazy, kontaktuje mě, prosím, podle potřeby.

Kontaktní údaje:

Jan Petrzílka

jan@petrzilka.cz

+420 777818097

