



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

SYSTÉM PRO VYKAZOVÁNÍ ZDRAVOTNÍ PÉČE

HEALTH CARE REPORTING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. DOMINIK PLACHÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2024

Zadání diplomové práce



156746

Ústav: Ústav informačních systémů (UIFS)
Student: **Plachý Dominik, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Inteligentní systémy
Název: **Systém pro vykazování zdravotní péče**
Kategorie: Informační systémy
Akademický rok: 2023/24

Zadání:

1. Prostudujte současné technologie pro tvorbu webových informačních systémů.
2. Prostudujte problematiku vykazování zdravotní péče pojišťovnam, požadovanými formáty dat a API jednotlivých pojišťoven.
3. Analyzujte požadavky na informační systém, který bude podporovat vykazování zdravotní péče pojišťovnam, včetně komunikace s jejich API a možností oprav výkazu na základě upozornění ze strany pojišťovny.
4. Navrhněte informační systém dle požadavků. Návrh konzultujte s vedoucím.
5. Implementujte navržený systém a otestujte jeho funkčnost na vhodném vzorku dat.
6. Zhodnotte dosažené výsledky a další možnosti pokračování tohoto projektu.

Literatura:

- VZP ČR. Metodika vyúčtování zdravotní péče [online]. [cit. 2023-10-18]. Dostupné z: <https://www.vzp.cz/poskytovatele/vyuctovani-zdravotni-pece/metodika-vyuctovani-aktualni-stav>
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015. ISBN: 978-80-251-4573-9.
- Welling, L., Thomson, L.: PHP a MySQL: Kompletní průvodce vývojáře. CPress, 2017.

Při obhajobě semestrální části projektu je požadováno:
Body 1-4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bartík Vladimír, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 17.5.2024
Datum schválení: 30.10.2023

Abstrakt

Vykazování zdravotní péče slouží k dokládání provedených zdravotních výkonů zdravotním pojišťovnám. Zdravotní pojišťovny z těchto výkazů jednotlivé provedené výkony ohodnocují a proplácejí. Pro úspěšné vykázání zdravotní péče musí systém vést evidenci provedených návštěv pacientů, vykonaných výkonů v rámci těchto návštěv a cest, které zdravotník musel podstoupit, aby se k pacientovi dostal. Cílem této práce je vytvořit část webového informačního systému, která bude umožňovat vybrat provedené výkony a cesty, zkontrolovat možné konflikty mezi nimi, případně umožnit opravu a nakonec vygenerovat speciální soubory KDAVKA a FDAVKA, které v sobě obsahují všechna vykazovaná data, musejí striktně dodržovat formát definovaný Veřejnou zdravotní pojišťovnou České republiky a jsou bez dalších úprav připraveny na odeslání zdravotní pojišťovně. Dále musí systém umožňovat správu vykázaných výkonů a cest pro případ, že je pojišťovna odmítne vyúčtovat. Všechna vytvořená vyúčtování musejí být také v systému dohledatelná s možností nahlédnutí do jejich částí.

Abstract

Healthcare reporting is used to document performed healthcare procedures to health insurance companies. Health insurance companies use these reports to evaluate and reimburse individual performed procedures. To successfully report health care, the system must keep records of carried out patient visits, the procedures performed during those visits and the trips the health care professional had to take to get to the patient. The aim of this thesis is to create a part of the information system that will allow selecting performed procedures and trips, checking possible conflicts between them, allowing correction if necessary, and finally generating special files KDAVKA and FDAVKA, that contain all the reported data, must strictly follow the format defined by the Veřejná zdravotní pojišťovna České republiky and are ready to be sent to the health insurance company without further modifications. In addition, the system must allow for the management of reported procedures and travel in case the insurance company refuses to bill for them. All generated bills must also be traceable in the system, with the possibility of inspecting all parts of them.

Klíčová slova

Informační systém, vykazování, KDAVKA, FDAVKA, vyúčtování, faktura, dávka, doklad

Keywords

Information system, reporting, KDAVKA, FDAVKA, billing, invoice, batch, document

Citace

PLACHÝ, Dominik. *Systém pro vykazování zdravotní péče*. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Bartík, Ph.D.

System pro vykazování zdravotní péče

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Vladimíra Bartíka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Dominik Plachý
16. května 2024

Poděkování

Chtěl bych poděkovat vedoucímu práce panu Ing. Vladimíru Bartíkovi, Ph.D. za jeho výborné vedení, rady a pomoc při vzniku práce.

Obsah

1	Úvod	2
2	Současné webové technologie	3
2.1	Informační systém	3
2.2	Klientská část systému	3
2.3	Serverová část systému	4
2.4	Data systému	4
2.5	Komunikace klientské a serverové části	4
2.6	Nástroje a praktiky ve vývoji webových systémů	4
2.7	Bezpečnost ve webových informačních systémech	5
3	Vykazování zdravotní péče pojišťovnám	7
3.1	Problematika vykazování zdravotní péče	7
3.2	Aplikační rozhraní pojišťoven	9
3.3	Formát vykazovaných dat	10
4	Analýza požadavků na informační systém	15
4.1	Identifikace požadavků	15
4.2	Specifikace požadavků pro systém XCarol	16
4.3	Specifikace požadavků pro vykazování zdravotní péče	18
5	Návrh informačního systému	22
5.1	Návrh základní struktury systému XCarol	22
5.2	Návrh vykazování zdravotní péče	25
6	Implementace	32
6.1	Implementace základní kostry systému XCarol	32
6.2	Implementace vykazování zdravotní péče	38
6.3	Testování	48
7	Závěr	51
	Literatura	52

Kapitola 1

Úvod

Zdravotní služby patří neodmyslitelně k životům nás všech. Moderní medicína není pouze o poskytování péče, ale také o efektivním a bezpečném spravování rozsáhlých informací týkajících se pacientů. K tomu nám slouží informační systémy, které se staly nezbytným prvkem zdravotnické infrastruktury. Tyto systémy usnadňují práci zdravotních pracovníků tím, že umožňují rychlé sdílení informací o pacientech na velkou vzdálenost a zjednodušují správu zdravotnických dat. Zároveň snižují byrokratickou zátěž díky automatizaci procesů.

Pro optimální fungování zdravotních služeb je klíčové zajistit adekvátní financování, o které se v České republice starají zdravotní pojišťovny. Konkrétní výkony provedené na pacientech jsou pečlivě identifikovány unikátními kódy dohodnutými mezi zdravotními službami a pojišťovnami. Na základě předloženého seznamu těchto identifikátorů je pojišťovna následně schopna proplatit zdravotní službě provedené výkony. Právě téma vykazování vykonaných výkonů pojišťovnám je hlavním zaměřením této diplomové práce.

Funkce vykazování zdravotní péče bude v rámci této práce integrována do komplexního webového informačního systému XCarol, který budou využívat především terénní zdravotní služby, v rámci kterých zdravotníci poskytují péči přímo v domácnostech pacientů. Výsledný systém poskytne zdravotníkům širokou škálu funkcí, včetně načítání, zadávání a analýzy různorodých dat o pacientech, a umožní jim také detailní evidenci provedených výkonů a záznamů o cestách za pacienty. Na základě těchto informací bude funkcionality vykazování zdravotní péče schopna automaticky vykazovat provedené výkony pojišťovnám.

Nejdříve je v kapitole 2 popsána teorie a moderní přístupy pro tvoření webových informačních systémů. Kapitola 3 je věnována problematice vykazování zdravotní péče pojišťovnám, včetně popisu aplikačního rozhraní a konkrétních formátů odesílaných souborů. V kapitole 4 je nejdříve rozebrána identifikace požadavků na informační systém, a poté jsou požadavky specifikovány. Na specifikaci požadavků navazuje kapitola 5, kde jsou ze specifikovaných požadavků konkrétně navrhnuté požadované části systému. Kapitola 6 popisuje implementaci a testování navržených funkcí a na závěr kapitola 7 zhodnocuje, co všechno se podařilo implementovat a co bude ještě potřeba vylepšit.

Kapitola 2

Současné webové technologie

Tato kapitola je zaměřena na popis aktuálních technologií pro tvorbu webových informačních systémů.

2.1 Informační systém

„Informační systém je integrovaná sada komponent pro sběr, ukládání a zpracování dat a poskytování informací, znalostí a digitálních produktů“ [6]. Pro zdravotní služby jsou informační systémy klíčovým prvkem, usnadňujícím správu a výměnu informací mezi zdravotnickými pracovníky, pacienty a dalšími stranami. Následující sekce této kapitoly se zaměří na konkrétní technologie, které přispívají k efektivnímu vývoji a fungování moderních webových informačních systémů.

2.2 Klientská část systému

Klientská část informačního systému, známá také pod anglickým výrazem „frontend“, představuje vizuální a interaktivní část systému, s níž uživatel přímo interaguje na obrazovce. Jejím hlavním úkolem je poskytovat uživatelsky příjemné prostředí pro přehledné zobrazení a pohodlnou manipulaci s daty. Zajišťuje tak uživatelsky přívětivý přístup k funkcím systému a přispívá k celkovému uživatelskému zážitku. Klientská část webového informačního systému běží v internetovém prohlížeči uživatele.

Pro vytváření klientské části slouží především jazyk HTML, který definuje strukturu obsahu webových stránek, a jazyk CSS, který se stará o vzhled prvků definovaných pomocí HTML. Pro dynamické interakce a manipulaci s obsahem stránky na straně klienta slouží skriptovací jazyky, nejčastěji JavaScript (případně TypeScript, což je pouze typovaný JavaScript), ale lze použít také PHP, Elm nebo Kotlin.

Pro pohodlnější vývoj klientské části se používají takzvané frameworky, jejichž účelem je zjednodušit a urychlit vývojový proces. Frameworky poskytují hotové struktury a nástroje, které usnadňují organizaci a správu kódu. Jsou navrženy tak, aby standardizovaly postupy a minimalizovaly opakovaný kód, což vede k efektivnějšímu vývoji a údržbě aplikací. Nejpoužívanějšími frameworky postavené na JavaScriptu (TypeScriptu) jsou React, Angular a Vue.js.

2.3 Serverová část systému

Serverová část informačního systému, též označovaná anglickým termínem „backend“, je zodpovědná za manipulaci s databází, poskytování dat pro klientskou část a přijímání dat z klientské části. Zajišťuje efektivní správu dat, implementuje pro uživatele neviditelnou logiku nad daty a poskytuje aplikační rozhraní (takzvané API), pomocí kterého probíhá komunikace s klientskou částí. Serverová část webového informačního systému běží na vzdáleném počítači – serveru, který musí pro správný chod systému zvládat obsluhovat všechny klienty současně.

Pro vývoj serverové části se používají například jazyky PHP, Node.js (platforma postavená na JavaScriptu), Python, Ruby nebo Java. Stejně jako v klientské části, tak i tady se používají frameworky pro snadnější a rychlejší vývoj. Mezi nejpoužívanější frameworky patří například Laravel či Symfony pro PHP, Nest.js pro Node.js nebo Django pro Python.

Pro snadnější manipulaci s daty mezi databází a zdrojovým kódem aplikace se využívá takzvaný „Object-Relational Mapping“ (ORM), který umožňuje mapování objektů v programovacím jazyce na relační struktury v databázi. Pro tento účel existují ORM frameworky, které zjednodušují práci s databázovými operacemi. Například TypeORM nebo Sequelize pro JavaScript/Node.js a Hibernate pro Javu.

2.4 Data systému

Data představují jádro každého informačního systému. Pro ukládání a správu dat slouží databázové systémy. Nejpoužívanější jsou relační databáze, například MySQL, PostgreSQL nebo Microsoft SQL Server, které jsou často používány pro strukturované a spojované datové modely. Dále lze použít NoSQL databáze, které nabízí alternativu pro ukládání nestrukturovaných nebo hierarchických dat. Mezi NoSQL databáze patří například MongoDB nebo Apache Cassandra.

2.5 Komunikace klientské a serverové části

Jak již bylo zmíněno v sekci o serverové části systému, tato část implementuje aplikační rozhraní, pomocí kterého spolu klientská a serverová část komunikují. Aplikační rozhraní je sada definicí a pravidel, která stanovuje, jak mohou klientská a serverová část vzájemně komunikovat a vyměňovat si data. Prakticky je to seznam `url` adres, které jsou namapovány na funkce.

V současné době jsou nejpoužívanějšími architekturami aplikačního rozhraní REST (Representational State Transfer) a GraphQL. Architektura REST využívá standardní metody HTTP, jako jsou GET, POST, PUT a DELETE, a pracuje s objekty reprezentujícími stav aplikace. Naopak při použití GraphQL klientská část systému specifikuje, jaká data potřebuje, a serverová část vrací odpověď obsahující pouze tato požadovaná data.

2.6 Nástroje a praktiky ve vývoji webových systémů

Vývoj webových informačních systémů je náročný proces, který vyžaduje efektivní nástroje. Tato sekce tyto nástroje krátce popisuje.

Integrovaná vývojová prostředí (IDE)

Integrovaná vývojová prostředí poskytují komplexní sadu nástrojů pro psaní, testování a ladění zdrojového kódu. Příkladem prostředím může být například Visual Studio Code, Sublime Text, či JetBrains IntelliJ IDEA. Všechna tato zmíněná prostředí navíc podporují rozšíření vyvíjené vývojářskou komunitou (pluginy), což každému umožňuje téměř neomezené možnosti v přizpůsobení svého vývojového prostředí podle vlastních potřeb. Například se v roce 2023 výrazně rozšířil plugin GitHub Copilot, který pomocí umělé inteligence nabízí vývojáři zdrojový kód na doplnění a dokáže výrazně urychlit práci.

Správa verzí

Správa verzí se netýká pouze webových informačních systémů, ale úplně jakéhokoliv IT projektu a zajišťuje ji distribuovaný systém Git. Git umožňuje vývojářům sledovat změny v kódu, spolupracovat na projektech a zajišťovat bezpečné verzování s možností vrátit se ke kterékoliv dřívější verzi. Pro centrální správu repositářů a spolupráci v týmu pak slouží například platformy GitHub, GitLab nebo JetBrains Space.

Automatické testování

Webové informační systémy jsou často velice komplexní a rozsáhlé se spoustou okrajových případů. Automatické testy jsou v těchto systémech nezbytné, protože není v silách vývojáře po každé změně otestovat, že tato změna nezpůsobila chybu kdekoli jinde v systému. Často využívanými nástroji jsou například Jest, Selenium a Cypress.

Správa balíčků

Občas je pro vývoj webového informačního systému potřeba funkce nebo komponenta, kterou již někdo vytvořil, ale není součástí základních balíčků. Pro jednoduchou a efektivní integraci takovýchto funkcí nebo komponent do systému slouží nástroje pro správu balíčků, například npm pro JavaScript a Node.js.

2.7 Bezpečnost ve webových informačních systémech

Bezpečnost ve webových informačních systémech je velice důležitá, protože systém často ukládá velké množství citlivých dat. Systém tedy musí být navržený tak, aby data chránil, udržoval svoji integritu a zabránil neoprávněným přístupům.

Fundamentálním prvkem bezpečnosti je autentizace a autorizace uživatelů. Pouze oprávnění uživatelé mohou mít přístup k citlivým datům a funkcím a nikdo jiný se k nim nesmí dostat. Autentizace ověřuje totožnost uživatele a autorizace řídí jeho přístupová práva.

Zranitelným místem může být také přenos citlivých dat přes internet. Používá se proto protokol HTTPS, který zajišťuje šifrovaný přenos dat mezi klientem a serverem, což minimalizuje riziko odposlechu či úpravy dat během přenosu.

Systém musí také odolat různým typům útoků. Například „SQL injection“, jehož podstatou je přímé vložení SQL příkazu do dat, který může obsahovat například příkaz pro vymazání celé databáze. Dalšími příklady útoků můžou být „cross-site scripting“ (XSS, vkládání zdrojového kódu do webové stránky) a „cross-site request forgery“ (CSRF, využívání důvěryhodnosti přihlášeného uživatele, když tento uživatel neúmyslně vykonává akce, které útočník naprogramoval). Pro ochranu proti útokům je klíčové implementovat

bezpečnostních mechanismy, například filtraci vstupů, escapování výstupů a použití bezpečnostních hlaviček.

Nad systémem by ideálně mělo běžet monitorování, které umožňuje detekovat potenciální nebezpečné aktivity.

Posledním bodem v této sekci, avšak ne posledním ve věci bezpečnosti systému, je školení uživatelů. Je důležité uživatele naučit, jak systém správně používat a učit je správné bezpečnostní postupy – například nemít přístupové heslo zapsané na veřejném místě, nikomu heslo nesdělovat nebo neodcházet na veřejném místě od počítače bez odhlášení.

Kapitola 3

Vykazování zdravotní péče pojišťovnám

Vykazování zdravotní péče pojišťovnám představuje klíčový proces v rámci zdravotnické administrativy, spojující poskytovatele zdravotní péče se zdravotními pojišťovnami. Financování zdravotních služeb za provedené výkony na pacientech je závislé právě na tomto procesu. Tato kapitola rozebírá problematiku vykazování zdravotní péče, formátování vykazovaných dat a aplikační rozhraní pojišťoven, s nímž systémy komunikují při automatizovaném vykazování.

3.1 Problematika vykazování zdravotní péče

Zdravotničtí pracovníci pečlivě dokumentují každý provedený výkon, cestu, podané léky, pobyt pacienta ve zdravotnickém zařízení, rozhovory s rodinou pacienta a mnoho dalších věcí. Veškeré hrazené služby jsou detailně stanoveny zákony České republiky, konkrétně ustanovením § 13 zákona č. 48/1997 Sb. o veřejném zdravotním pojištění – znění od 01.07.2023 [7].

Poskytovatel zdravotních služeb všechna tato data předává zdravotní pojišťovně, která provedené výkony pečlivě analyzuje. Data podléhají spoustě podmínek a omezení, které nesmějí být pro úspěšné vykázání porušeny. O těchto podmínkách podrobně pojednává dokument [2], ze kterého budou vycházet následující odstavce.

Vykazovaná data obsahující informace o provedených výkonech se nazývají dávky. Pojišťovna od poskytovatele zdravotních služeb přebírá dávky podle konkrétních ustanovení ve smlouvách. Společně s dávkami přebírá také faktury. Nad přebranými daty jsou nejprve provedeny vstupní kontroly (soubory jsou čitelné, neobsahují viry, jsou obsaženy správné soubory, atd.). Pokud data touto fází úspěšně projdou, jsou zavedena do informačního systému pojišťovny a jsou zahájeny další kontroly a validace – tentokrát už jsou kontrolovány přímo obsahy souborů. Například je kontrolováno, jestli odpovídá počet dávek uvedený ve faktuře a skutečně dodaný počet dávek nebo jestli nejsou některé výkony vykázány vícekrát, než je jejich limit v určitém časovém období. Na konec probíhá kontrola revizním lékařem pojišťovny, který je především odpovědný za vyhodnocení (proplacení, úprava, odmítnutí) speciální položky v dávce, ale stále také může odmítnout běžnou položku nebo snížit hodnotu dokladu. Jestliže je v kterékoliv fázi kontrol nalezena chyba, pojišťovna doklad odmítne a vrátí jej zpět poskytovateli zdravotních služeb, který může doklad opravit a znovu

předložit ve smluveném termínu. Pokud je s chybou vrácena faktura, tak se neopravuje, ale je vystavena nová s novým evidenčním číslem.

Vyhodnocování dávek probíhá mezi denními uzávěrkami. Do doby denní uzávěrky mohou revizní lékaři provádět povolené změny v dokladech. Po denní uzávěrce již změny v dokladech dělat nelze a pojišťovně vznikají finanční závazky vůči poskytovateli zdravotních služeb.

Poskytovatel zdravotních služeb je v každém případě informován pojišťovnou o výsledku zpracování a vyhodnocení dokladů. Jako doklad o výsledku slouží takzvané sestavy. Podoba sestav se může u každé zdravotní pojišťovny lišit. Následující popis se týká pouze sestav, které dodává Všeobecná zdravotní pojišťovna.

Sestava se skládá ze záhlaví, těla a souhrnu. Záhlaví obsahuje základní údaje o sestavě (označení sestavy, datum vytvoření sestavy, číslo denní uzávěrky, identifikace poskytovatele zdravotních služeb, atd.). Tělo sestavy je psáno v řádcích rozdělených do sloupců, kde každý řádek obsahuje identifikaci dávky dokladů, identifikaci dokladů a součty sloupců v těle sestavy. Souhrn udává celkové součty, datum a podpis. Konkrétně se sestavy týkají například protokolu o kontrole datového nosiče, denní uzávěrky dávek k proplacení (stav a ocenění dávek k proplacení po denní uzávěrce) nebo denní uzávěrky odmítnutých dávek, dokladů a položek (rozpis odmítnutých dávek s uvedením důvodu odmítnutí).

Obecná pravidla pro vykazování provedených výkonů

Pro vykazování provedených výkonů platí několik obecných pravidel, která jsou uvedena v následujících bodech [5].

- Zdravotní pojišťovna proplácí jednou provedený výkon pro jednoho pacienta pouze jednomu poskytovateli. Například pokud je pacient vyšetřen obvodním lékařem a následně poslán do nemocnice ke stejnému vyšetření, obvodní lékař a nemocnice vykazují tyto výkony nezávisle na sobě a pojišťovna oba výkony nezávisle proplatí.
- Zdravotní pojišťovna hradí výkony pro jednoho pacienta v jednom dni u jednoho poskytovatele zdravotních služeb pouze jedenkrát, pokud není u výkonu uvedeno jinak. Výjimku tvoří například výkony prováděné na párových orgánech. V tomto případě může pojišťovna proplatit výkon dvakrát za den, pokud byl proveden na levém i pravém orgánu. I zde však existují výjimky a rozhodující jsou vždy podmínky specifikované u daného výkonu.
- Vyúčtovat jednomu pacientovi jeden typ výkonu u jednoho poskytovatele dvakrát za den je možné pouze v případě, že dojde k nové návštěvě téhož lékaře z důvodu náhlé změny zdravotního stavu. Například pacient dojde dvakrát za den na převaz popálenin.
- Výkon je hrazen zdravotní pojišťovnou pouze v případě, že byl proveden v plném rozsahu. Například odběr krve ze žíly u dítěte nelze účtovat, pokud nemohl být plně dokončen z důvodu nespolupráce pacienta.
- Všechny hrazené služby, které byly účtovány prostřednictvím provedených výkonů, musí být pečlivě zdokumentovány v zdravotnické dokumentaci pacienta. Zápis do této dokumentace je neoddělitelnou součástí každého provedeného výkonu. Dokumentace výkonu se sestává z jeho definice (jednotlivé kroky postupu při výkonu), spotřebovaných materiálů a spotřebovaných léčivých přípravků.

- Aby mohl být výkon vykázán, musí být proveden přesně jak je definován. Příkladem může být operace, která je z důvodu nejrůznějších možných komplikací prováděna laparoskopicky. Takový výkon nelze vykázat.
- Výkon nemůže být součástí jiného výkonu, aby mohl být vykázán.
- Na jednoho pacienta lze při jeho jedné návštěvě vykázat pouze jeden výkon jedné odbornosti.
- Zdravotní pojišťovna je oprávněna neuhradit neoprávněně vykázaný výkon. Například mohou být při jedné pacientově návštěvě současně vykázány dva výkony, které ale nesmějí být vykazovány společně.
- Výkony, které mohou být podle seznamu výkonů účtovány více než jednou za den, mohou být vyúčtovány pouze tolikrát, kolikrát byl naplněn celý čas jejich provedení také uvedený v seznamu výkonů. Čas zahájení a ukončení těchto výkonů je nutnou součástí dokumentace pacienta. Pokud se u těchto výkonů ve vykázaných datech vyskytne nějaká nepřesnost nebo nelze jednoznačně určit časy zahájení a ukončení, zdravotní pojišťovna uhradí pouze jeden provedený výkon.

Jak již bylo zmíněno v úvodu, tato práce se bude zabývat především vykazováním zdravotní péče terénních zdravotních služeb. Pro tyto terénní výkony platí pravidlo, že musejí mít v dokumentaci o provedení povinně uvedeno datum a čas návštěvy pacienta.

3.2 Aplikační rozhraní pojišťoven

Obecný princip aplikačního rozhraní byl již rozebrán v sekci 2.5. Jedná se o přístupový bod serveru, přes který lze se serverem komunikovat, posílat dotazy a požadavky. Často je z důvodu citlivých dat přístup podmíněn autentizací (přihlášením pomocí zaregistrovaných údajů nebo certifikátem).

V problematice vykazování zdravotní péče jsou na aplikační rozhraní pojišťoven odesílána vykazovaná data a zpět jsou přijímány výsledky jejich vyhodnocení. V České republice zajišťuje vykazování zdravotní péče Všeobecná zdravotní pojišťovna České republiky (dále VZP), která přijímá výkazy určené pro všechny české zdravotní pojišťovny a následně jim je preposílá. Systémy podporující funkcionalitu vykazování tak posílají všechny doklady týkající se vykazování na jednu adresu, nezávisle na tom, pod jakou pojišťovnou je pacient vedený.

VZP pro komunikaci s jejím aplikačním rozhraním provozuje takzvanou B2B komunikaci. Poskytovatel zdravotních služeb musí nejdříve s pojišťovnou uzavřít smlouvu o předávání dat. Po vyplnění dokumentů vytvoří pojišťovna přístup do aplikace VZP Point a pošle poštou aktivační klíč. Po aktivaci klíče v aplikaci je komunikace mezi poskytovatelem zdravotních služeb a pojišťovnou zprovozněna. Pro poskytovatele softwaru, který má s pojišťovnou komunikovat za poskytovatele zdravotních služeb, platí podobný postup. Po uzavření smluv obdrží poskytovatel softwaru certifikát, který pak přikládá společně s dotazy na aplikační rozhraní VZP.

VZP na svém aplikačním rozhraní poskytuje služby, které jsou detailně popsány v dokumentu [1], ze kterého také vycházejí následující body pojednávající o typech poskytovaných služeb.

- Synchronní služby – výsledek dotazu přichází jako odpověď na odeslaný dotaz. Odpověď se dostaví v rámci několika vteřin. Do synchronních služeb patří například kontrola stavu a druhu pojištění pacienta, kontrola stavu smlouvy poskytovatele zdravotních služeb s VZP nebo ověření platnosti průkazu pojištěnce.
- Asynchronní služby – zpracování trvá déle a odpověď přichází na koncový bod (end-point) tazatele. Adresa tohoto koncového bodu je uvedena v rámci odeslaného dotazu. Asynchronní služby se dělí na služby zpracovávané bez zásahu pracovníků VZP a služby vyžadující zásah pracovníků VZP. Službám prováděným bez zásahu pracovníků trvá odpovědět v řádech minut až hodin. Patří sem například přehled o platbách zaměstnavatele, údaje o zaměstnavateli nebo získání seznamu registrovaných pojištěnců u zdravotní pojišťovny. Službám vyžadující zásah pracovníků VZP trvá odpovědět v řádech hodin až dnů. Některé z těchto služeb informují průběžně o stavu zpracování. Spadají sem služby týkající se vykazování zdravotní péče.
- Abonované služby – patří do nich pouze služba získávání číselníků. Služba funguje na principu předplatného. To znamená, že v rámci nastavení B2B účtu si lze zaregistrovat odběr aktualizací jednotlivých číselníků. Kdykoliv se jeden z odebíraných číselníků aktualizuje, přijde odběrateli tato aktualizace na jeho koncový bod (end-point) specifikovaný při registraci odběru. Díky této službě lze udržovat potřebné číselníky aktuální.

B2B prostředí prochází každý měsíc pravidelnou údržbou. Data údržeb se zveřejňují vždy rok dopředu na webových stránkách VZP¹. Zpravidla to bývá ve čtvrtek od 20:00 do 24:00 a je tedy potřeba s tím při vývoji počítat.

3.3 Formát vykazovaných dat

Pro úspěšné zpracování provedených výkonů a následné proplacení ze strany zdravotní pojišťovny je nezbytné data správně naformátovat a odeslat na aplikační rozhraní pojišťovny. O způsobu odeslání dat pojednává předchozí podkapitola a tato podkapitola se zaměřuje na rozbor problematiky jejich formátování.

Vykazovaná data obsahují informace o provedených výkonech zdravotníky a jejich cest za pacienty za dané období a ukládají se ve striktně definovaném formátu do souboru KDAVKA, který se pak odesílá na aplikační rozhraní pojišťovny. Tento soubor může mimo jiné obsahovat hodnoty definované číselníky VZP. Spolu se souborem KDAVKA je možné odesílat i soubor FDAVKA, který obsahuje faktury. Nejprve následuje sekce věnující se číselníkům a za ní následují sekce s podrobnou analýzou formátu souborů KDAVKA a FDAVKA.

Číselníky

Všeobecná zdravotní pojišťovna (dále VZP) definuje kromě formátu souborů s vykazovanou péčí také sadu číselníků².

Číselníky obsahují typy zdravotních výkonů, diagnóz, léků a dalších relevantních informací. Každá položka v číselnících má unikátní kód, na který se lze při vykazování zdravotní

¹Aktuální naplánované údržby B2B: <https://www.vzp.cz/e-vzp/informace-pro-sw-firmy/b2b-komunikace/odstavky-b2b-prostredi>

²Sady číselníků lze nalézt zde: <https://www.vzp.cz/poskytovatele/ciselniky>

péče odkazovat. Mezi číselníky patří například kategorie pacienta, doprava, mezinárodní klasifikace nemocí, zdravotní pojišťovny nebo smluvní odbornosti pracovišť.

Číselníky lze stáhnout buď přímo z webových stránek VZP, nebo přes aplikační rozhraní pojišťovny. Stažené soubory mají striktně definovaný formát a striktně definovaný název, který se řídí podle toho, o jaký číselník se jedná (například pro kategorie pacienta má soubor jméno KATEGPAC.XXXXX, kde XXXXX je verze číselníku). Soubory obsahují pouze ASCII znaky v kódování CP852 (PC LATIN2). Každý řádek souboru představuje jednu položku číselníku a každá položka se skládá z proměnného počtu atributů oddělených čárkou. Význam jednotlivých atributů je popsán jednak přímo u daného číselníku na webových stránkách VZP a jednak v oficiální dokumentaci k číselníkům [4].

Popis jednoho atributu se skládá z následujících částí:

- Zkratka – jednoznačně určuje atribut, například zkratka KOD, která označuje atribut, který obsahuje identifikační kód položky číselníku.
- Typ – určuje, jakého typu je hodnota atributu. Může nabývat 4 hodnot: C (znakový atribut), N (číselný atribut), D (datum ve formátu DDMMYYYY) a \$ (peněžní atribut). Nenumerické hodnoty atributů musejí být uzavřeny v uvozovkách („“).
- Délka – maximální počet znaků, ze kterého se může hodnota atributu skládat. Pro číselné a peněžní atributy se v případě desetinných hodnot zapisuje stylem <délka celého čísla včetně desetinné tečky>. <délka desetinné části>
- Popis – slovní popis atributu. Typicky sděluje celý název atributu, z něhož bývá vytvořena zkratka atributu, a dále může obsahovat dodatečný popis nebo například jakýsi „podčíselník“ definující možné zkratky a jejich význam, které mohou tvořit hodnotu atributu.

Například číselník s názvem „Kategorie pacienta“ má atributy definované v tabulce 3.1 a obsah samotného souboru v 1.

Tabulka 3.1: Popisu atributů pro číselník „Kategorie pacienta“.

Název	Typ	Délka	Popis
KOD	C	1	Kód kategorie
NAZ	C	254	Název kategorie
TXT	C	254	Text definice
POD	N	4	Příplatek k ošetrovacímu dni hospitalizace - počet bodů dle pravidel ve vyhlášce č. 134/1998 Sb.

Soubor 1: Obsah číselníku „Kategorie pacienta“.

1	"0","Pacient na propustce","Vykáže se každý den, kdy je pacient na propustce",0
2	"1","Pacient soběstačný","Pacient je nezávislý na základní ošetrovatelské péči nebo dítě nad 10 let",0
3	"2","Pacient částečně soběstačný","Pacient je částečně soběstačný, sám se obslouží s dopomocí, je schopen pohybu mimo lůžko s dopomocí či samostatně na invalidním vozíku nebo dítě od 6 do 10 let",75
4	"3","Pacient vyžadující zvýšený dohled","Lucid.pacient, neschopný pohybu mimo lůžko ani s dopomocí či samost.na invalidním vozíku, vyžad. téměř úplnou obsluhu nebo psychic.alterov. pacient vyžadující zvýš. dohled, příp.nutné přechod. omezení pohybu či farmakol.zklidnění nebo dítě od 2 do 6 let",150
5	"4","Pacient imobilní","Lucidní, zcela imobilní pacient, případně inkontinentní, vyžaduje ošetrovatelskou pomoc při všech úkonech nebo dítě do 2 let věku",225
6	"5","Pacient v bezvědomí","Pacient je v bezvědomí, případně v deliriózním stavu",300

Jak je v obsahu souboru číselníku Kategorie pacienta vidět, dodržuje formát definovaný tabulkou 3.1. Každý řádek tedy začíná jednoznačným kódem, pokračuje názvem kategorie, který může mít až 254 znaků, následuje delší definice kategorie a celý řádek končí maximálně čtyřmístným číslem udávajícím příplatek k ošetřovanému dni hospitalizace. Znakové atributy (typu C) jsou uvedeny v uvozovkách, číselné ne.

Soubor KDAVKA

Soubor KDAVKA slouží pro vykazování provedených výkonů zdravotníky a jejich cest za pacienty. Obsah souboru KDAVKA je definován dokumentem od VZP [3], ze kterého tato podkapitola vychází.

Stejně jako soubory s číselníky, i soubor KDAVKA může obsahovat pouze ASCII znaky v kódování CP852 (PC LATIN2). Jméno souboru je standardizováno na KDAVKA.XXX, kde XXX představuje kód zdravotní pojišťovny pacienta odpovídající jedné položce z číselníku „Zdravotní pojišťovny“. V rámci jednoho vyúčtování (odeslání souborů na aplikační rozhraní pojišťovny) lze odeslat pouze jeden soubor KDAVKA.XXX.

Soubor se skládá z takzvaných vět, z nichž každá tvoří jeden řádek souboru. Soubor se skládá z dávek, kde každá je uvedena úvodní větou, která obsahuje základní informace o dávce, a následují věty vykazující provedené zdravotní výkony, nebo cesty (podle typu dávky). Pro jednotlivé typy dávek je definováno několik typů vět. Vždy je jedním z těchto typů vět úvodní věta dávky a dále to můžou být věty uvádějící informace o provedeném výkonu, vykonané cestě a mnoho dalších. Všechny věty se skládají z několika atributů pevné délky, které nejsou ničím odděleny. Všechny věty jednoho typu tak musejí mít vždy stejnou délku. Z toho důvodu je potřeba striktně dodržovat délky jednotlivých atributů, a to i v případě nevyplnění nepovinného atributu, nebo vyplnění atributu krátkou hodnotou. V konkrétních případech mohou být pravidla pro vyplňování atributů do plné délky upravena, jinak obecně platí následující:

- Znakové atributy – vyplňují se zleva a zbývající délka atributu je vyplněna mezerami. Pokud má jejich hodnota odpovídat hodnotě v číselníku, musí rozsahu této hodnoty

odpovídat úplně přesně, včetně možných předřazených nul. Atribut bez hodnoty je vyplněn mezerami.

- Numerické atributy – je povoleno použití pouze předřazených mezer a čísel „0“ až „9“. Nesmí se uvádět znaménko. V případě nevyplnění nepovinného atributu je potřeba atribut vyplnit mezerami (nebo nulami) a na poslední pozici vpravo vepsat „0“.
- Datum – povoleny jsou pouze číslice „0“ až „9“. Při nevyplnění je atribut vyplněn mezerami.
- Peněžní atributy – je povoleno použití předřazených mezer a čísel „0“ až „9“ stejně jako u numerických atributů. Navíc se zde ještě může použít desetinná tečka. Každé číslo je nutné vyplnit minimálně s dvěma desetinnými místy. Prázdný atribut je vyplněn mezerami (nebo nulami) a na posledních čtyřech pozicích vpravo musí obsahovat „0.00“.

Jednotlivé atributy pro různé typy vyúčtování popisuje dokument od VZP [3]. Atributy jsou popsány následujícími parametry:

- Zkratka – stejně jako u číselníků jednoznačně identifikuje atribut.
- Typ – určuje, jakého typu je hodnota atributu. Platí zde stejná pravidla jako v předchozí sekci týkající se číselníků.
- Délka – definuje přesnou délku hodnoty atributu. Pokud je hodnota prázdná, musí prostor vyplnit mezery. Pokud je hodnota krátká musejí se mezery vložit před ni (v případě numerických nebo peněžních atributů), nebo za ni (v případě textových atributů). Pro definici délky desetinného čísla platí stejná pravidla jako v předchozí sekci u číselníků. Na konci seznamu všech atributů jednoho typu věty je ještě navíc uvedena i celková délka věty, kterou musí věta bezpodmínečně dodržet.
- Začátek – označuje pozici ve větě, na které má atribut začínat. Pozice se číslují od nuly (první atribut začíná na pozici 0) a pozice začátku atributu se tak musí rovnat součtu délek všech předcházejících atributů.
- Popis – podrobněji popisuje atribut.

Příkladem nechť je „Vyúčtování výkonů v ústavní péči“, konkrétně popis atributů 6. typu věty - „kategorie pacienta“ z kapitoly 4.2 [3]:

Tabulka 3.2: Popis atributů pro 6. typ věty „Vyúčtování výkonů v ústavní péči“.

Název	Typ	Délka	Začátek	Popis
TYP	C	1	0	Typ věty „K“ – kategorie pacienta
KTYP	C	1	1	Druh kategorie podle číselníku „Kategorie pacienta“
KDNI	N	2	2	Počet dní v kategorii
TYPK	C	1	4	Doplňek typu věty kategorie pacienta. Rezerva, nevyplňuje se.
Celkem délka věty 5				

Dokument [3] obsahuje popis několika desítek různých typů vyúčtování. Ale jak již bylo zmíněno v úvodu, tato práce se primárně zaměřuje na terénní zdravotní služby, a proto jsou pro tuto chvíli pro soubor KDAVKA relevantní pouze „Poukaz na vyšetření/ošetření“ (kapitola 4.06) a „Vyúčtování cest lékaře v návštěvní službě“ (kapitola 4.36).

Soubor FDAVKA

Soubor FDAVKA je určen pro předávání žádostí o vyúčtování (faktury za dávky) a žádostí o předběžnou úhradu (faktury za období) a předává se na aplikační rozhraní zdravotní pojišťovny společně se souborem KDAVKA. Přesné jméno souboru musí být FDAVKA.XXX, kde XXX je kód zdravotní pojišťovny pacienta, za kterého je právě péče vykazována. Obsah souboru je podrobně popsán v kapitole 4.31 v dokumentu [3] od VZP. Jednotlivé atributy souboru jsou v dokumentu specifikovány stejným principem, jak je uvedeno v předchozí sekci o souboru KDAVKA a pro obsah souboru platí i stejné zásady pro kódování.

Soubor FDAVKA začíná záhlavní větou, která obsahuje základní informace o faktuře a kontrolní údaje úplnosti přijímaných dokladů. Za záhlavím následují věty, které mohou být čtyř různých typů: „Žádost o vyúčtování“ („V“), „Žádost o předběžnou úhradu“ („Z“), „Dávka zahrnutá do žádosti o vyúčtování“ („D“) a „Specifikace komunikačního rozhraní“ („K“). V závorkách jsou uvedeny jednoznakové kódy jednotlivých typů vět, kterými věty kvůli rozpoznání jejich typu vždy začínají. Záhlavní věta má kód „F“. Atributy za kódem typu vět jsou pak už pro jednotlivé typy vět odlišné.

Kapitola 4

Analýza požadavků na informační systém

Analýza požadavků představuje klíčový krok při vývoji informačního systému, neboť výsledná kvalita, použitelnost a uživatelská přívětivost systému jsou výrazně ovlivněny pečlivým a systematickým zhodnocením požadavků uživatelů a zainteresovaných stran. Tato kapitola se zaměřuje na identifikaci, specifikaci a analýzu požadavků, které budou základním kamenem pro následující fáze vývoje. Cílem je zajistit, že systém bude plně odpovídat potřebám uživatelů a splní stanovené cíle efektivně a spolehlivě.

4.1 Identifikace požadavků

Identifikace požadavků vzniká konzultacemi softwarového inženýra a cílového uživatele systému. Během těchto konzultací společně stanovují požadované funkcionality výsledného systému a na jakých zařízeních a v jakých podmínkách má systém bezchybně fungovat. Dále spolu jednájí i o obchodních záležitostech, jako jsou termíny dodání, způsob dodání, cena za vývoj, licenční podmínky pro zdrojový kód, řešení úprav a oprav zdrojových kódů po datu dodání a další. Ideální stav návrhu systému po identifikaci požadavků je komplexní a detailní přehled o funkcionalitách, požadavcích a specifikacích, který pokrývá všechny běžné i okrajové případy používání a je vytvořen tak, že je technicky možné ho realizovat a uživatel je stoprocentně spokojen s výsledkem. Takového ideálního stavu je však téměř nemožné dosáhnout, a to zejména proto, že zadavatel nemá plně jasno ve svých potřebách a požadavcích, kdy ví, jak má funkcionalita obecně fungovat, ale nedokáže ji přesně popsat nebo si neuvědomí všechny okrajové případy, které mohou nastat. Také ale může dojít k pochybení ze strany vývojářů, například přehlédnutí pravděpodobných technických problémů, které následně nastanou při vývoji či fungování navržené funkcionality. Identifikaci požadavků tak nikdy není možné považovat za finální a bývá průběžně upravována tak, aby co nejvíce vyhovovala potřebám zákazníka, ale zároveň byl nutný co nejmenší zásah do již vytvořených částí systému.

Proces identifikace požadavků systému XCarol probíhal téměř rok formou série několika desítek schůzek mezi vývojáři a zástupci zdravotních služeb, které budou výsledný systém používat. Během tohoto období byl systém důkladně definován. Konkrétně vznikly návrhy jednotlivých stránek systému se všemi potřebnými detaily – na jakých místech se budou zobrazovat jaká data, jak budou interagovat jednotlivá tlačítka, jak se budou jednotlivé stránky a jejich části chovat, jaké budou na stránkách výjimky v zobrazení či ve funkcionalitě

závisející na zobrazených datech, přihlášeném uživateli či používaném zařízení. Následující sekce obsahují výsledky definice požadavků.

4.2 Specifikace požadavků pro systém XCarol

Tato sekce obsahuje základní obecné požadavky na systém XCarol, na jehož základech bude postavena funkcionality vykazování zdravotní péče. Specifikace požadavků pro vykazování zdravotní péče je popsána v následující sekci 4.3

Systém XCarol bude webový informační systém pro zdravotní služby, které v něm povedou evidenci pacientů, zaměstnanců, vykonaných prací a návštěv a uživatelé přes něj budou moci také komunikovat. Systém vychází ze systému WebCarol¹, který je aktuálně cílovými zdravotními službami používaný a jehož uživatelé budou přecházet právě na XCarol. Obecně bude systém XCarol obsahovat všechny funkcionality jako systém WebCarol, ale bude uživatelsky přívětivější, bude mít modernější vzhled a bude obsahovat nové funkce.

Uživatelé

Uživatelé systému budou zdravotní pracovníci, tedy převážně lékaři a zdravotní sestry. Uživatelům budou přidělovány role a práva, která budou určovat, do kterých částí systému budou mít přístup a jaká data uvidí. Administrace systému bude obsahovat kompletní správu uživatelů s možnostmi aktivovat či deaktivovat účet uživatele.

Organizace

Systém bude používán různými uživateli pracujícími v navzájem nezávislých organizacích. Veškerá data spadající pod tyto organizace musejí být striktně oddělena od ostatních organizací, aby se například v přehledu provedených návštěv v rámci organizace Charita Brno nezobrazovaly návštěvy provedené v rámci jiných organizací. Uživatel i systém tedy musí mít neustále povědomí o tom, jaká organizace je právě vybrána a jaká data se tak mají zobrazovat. Vybraná organizace musí zůstat neměnná nezávisle na navigaci uživatele po systému, dokud sám uživatel její výběr nezmění. Každý uživatel bude mít administrátorský nastavitelná práva na jednotlivé organizace, která budou určovat, jestli se mu daná organizace bude nabízet ve výběru organizací.

Služby

Hierarchicky pod organizacemi se budou nacházet služby. Služby budou typicky zastupovat jednotlivé typy zdravotní péče, které jejich nadřazená organizace poskytuje. Každá služba může poskytovat právě jeden typ zdravotní péče, který je určen takzvanou odborností². Každá odbornost je identifikována tříčíselným kódem, který je u běžných odborností mezi zdravotníky dobře známý. Systém proto bude ve většině případů zobrazovat pouze kód odbornosti. Systém XCarol bude zatím podporovat následujících 5 odborností:

- 720 – paliativní medicína
- 902 – fyzioterapeut

¹Úvodní informace k systému WebCarol lze najít zde: <https://webcarol.cz>

²Seznam všech odborností lze najít zde: <https://szv.mzcr.cz/Ciselnik/Odbornost>

- 913 – ošetrovatelská péče v sociálních službách
- 925 – domácí péče
- 926 – domácí paliativní péče o pacienta v terminálním stavu

Data služeb (například informace o složkách pacientů, záznamy z návštěv a vyšetření) spadají pod jednotlivé služby a platí zde stejné principy jako výše u organizací – uživatel i systém musí být jasné, jaká služba je právě vybrána, podle vybrané služby se musí řídit zobrazování dat, uživatel musí mít možnost přepínat mezi službami spadajícími do vybrané organizace a přístup uživatelů do jednotlivých služeb musí být řízen nastavitelnými právy.

Typy služeb

Služba bude obsahovat velké množství nastavení. Ze základních nastavení to budou například název, odbornost, logo či barva. Pokročilejší nastavení bude ale také umožňovat nastavení zobrazení jednotlivých stránek spadajících pod služby (nastavení názvu stránky v levém navigačním menu a u některých stránek také nastavení jejich rozložení) a používání dostupných funkcí (nastavení, zda bude/nebude umožněno jejich používání, případně pak detailní nastavení daných funkcí).

Nastavovat všechna tato rozsáhlá nastavení pro každou službu zvlášť by bylo příliš zdouhavé. Z tohoto důvodu bude systém obsahovat takzvané typy služeb. Typy služeb budou nastavitelné v administraci systému, takže k jejich nastavení budou mít přístup pouze uživatelé se speciálními právy. Typy služeb budou sloužit jako šablony pro tvoření samotných služeb a zároveň budou umožňovat administrátorské zásahy do nastavení všech služeb daného typu současně, protože budou obsahovat totožné nastavení jako samotné služby, ale změna v tomto nastavení se propíše do nastavení všech služeb vytvořených z tohoto typu. Nastavení samotné služby tak slouží především pro malé úpravy pro konkrétní účely dané služby a je podřízeno nastavení typu služby.

Například dokud je v nastavení typu služby povolena funkcionality „Zadat práci“, zobrazuje se nastavení funkcionality „Zadat práci“ i v nastavení služby a je tak možné tuto funkci povolit/zakázat podle potřeb dané služby. Pokud je ale funkcionality „Zadat práci“ zakázána už v nastavení typu služby, tak se v nastavení služby ani záložka s nastavením funkcionality „Zadat práci“ nezobrazí, je automaticky zakázaná a v nastavení služby nemůže být nijak aktivována.

Složky pacientů

V rámci služeb bude vedena evidence složek pacientů. Složka pacienta bude obsahovat všechny záznamy o zdravotní péči poskytnuté danému pacientovi v rámci dané služby. Systém bude podporovat detailní zadání práce, pomocí kterého budou zdravotní pracovníci evidovat cesty za pacienty, provedené zdravotní výkony, podané léky a naměřené hodnoty.

Pacienti

Pacienti nebudou součástí žádné organizace ani služby. Se službami budou propojováni pomocí zmíněných složek pacienta. Pacient tak bude moci být vyšetřován v rámci několika služeb současně a pro každou z těchto služeb bude mít zvlášť vytvořenou složku pacienta. Samotný pacient bude mít obsahovat data, která souvisí přímo s ním jako osobou. Budou to například veškeré základní informace (jméno, bydliště, datum narození, atd.), kontaktní osoby, diagnózy, naměřené hodnoty nebo informace o zdravotním pojištění.

Poukazy

Poukazy budou představovat předpisy zdravotní péče. Budou je vydávat lékaři pro konkrétní složky pacientů. Každý poukaz bude obsahovat hlavní diagnózu, začátek a konec platnosti poukazu a zdravotní výkony, které smějí být v rámci návštěv spadajících pod daný poukaz prováděny.

Ostatní požadavky

V systému bude integrována e-mailová schránka, datová schránka a plánovací kalendář. Uživatelé s vyššími oprávněními budou mít možnost plánovat terénním pracovníkům cesty za pacienty, generovat a exportovat různorodé statistiky a vykazovat poskytnutou zdravotní péči pojišťovně, což je hlavním tématem této práce a podrobné specifikaci je věnována následující sekce. Systém bude podporovat živý chat, hlasové hovory i video hovory. Části systému, které budou používány v terénu, musejí fungovat i bez připojení k internetu a po opětovném připojení k internetu se musí data synchronizovat.

4.3 Specifikace požadavků pro vykazování zdravotní péče

Vykazování zdravotní péče bude zautomatizovaný proces, jehož cílem bude vytvořit a odeslat dávky a faktury za provedené výkony a cesty na aplikační rozhraní zdravotní pojišťovny, poskytnout uživateli prohlédnutí výsledku vyúčtování a případně mu poskytnout možnost opravit chyby v odmítnutých dávkách nebo fakturách. Pro přístup do této funkcionality bude uživatel potřebovat speciální oprávnění. Běžný uživatel se tak k vykazování nedostane.

Vykazování bude vždy vykonáváno v rámci jedné služby a za nějaké uplynulé časové období, jehož délku a začátek bude možné libovolně zvolit s podmínkou, že začátek a konec období jsou ve stejném roce. Uživatel bude mít na výběr, jakým zdravotním pojišťovně provedenou zdravotní péči vykáže, z nichž musí vybrat alespoň jednu, ale může i víc. Výkony, které mají být vykázány (spadají do vybrané služby, byly provedeny ve vybraném časovém období, byly provedeny na pacientovi, jehož zdravotní pojišťovna je jedna z vybraných a ještě nebyly vykázány), budou před vyúčtováním kontrolovány, zda neporušují nějaké z pravidel stanovených pojišťovnou. Jednou z takových kontrol bude například ověření, že výkon 06318 nebyl jednomu pacientovi zadán vícekrát než třikrát za den. Pokud je některé z pravidel porušeno, bude systém na tuto chybu upozorňovat a umožňovat nápravu – buď automatickou, pokud to lze, nebo manuální.

Po vyřešení všech konfliktů a chyb ve výkonech a cestách odpovídajících uživateli zadání ještě systém zobrazí podrobný přehled všeho, co bude vytvořené vyúčtování zahrnovat, pro finální potvrzení. Bude možné si zde prohlédnout detaily jednotlivých faktur, dávek a dokladů, jak budou zapsány v souborech KDAVKA a FDAVKA. Faktury a dávky budou automaticky očíslovány podle přednastavených pravidel, ale uživatel bude mít možnost číslování upravit, pokud tato pravidla neporuší. Také bude umožněno upravit základní údaje u faktur (datum vystavení, datum splatnosti, atd.). Po potvrzení vytvoření vyúčtování systém vygeneruje soubory KDAVKA a FDAVKA a nabídne je uživateli ke stažení do lokálního úložiště. Uživatel je pak bude moci mimo systém odeslat zdravotní pojišťovně.

Systém bude poskytovat přehled vyúčtovaných výkonů a cest. Uživatel bude moci vyhledat konkrétní vyúčtované výkony a cesty podle zdravotní pojišťovny, čísla dávky a čísla dokladu. Takto vyhledané výkony a cesty bude možné manuálně odmítnout (změnit stav

z „Vyúčtovaný“ na „Odmítnutý“), což bude potřeba v případě, že zdravotní pojišťovna vyúčtování odmítne (například z důvodu kolize mezi vykázanými výkony, kterou se nepodařilo odhalit při kontrolách při odesílání). Takto odmítnutý výkon nebo cestu bude následně možné znovu vyúčtovat pomocí takzvaného opravného vyúčtování, nebo stornovat s udáním důvodu. Důvody budou na výběr 2: „Nebude uhrazeno pojišťovnou“ a „Omylem zadáno“. Stornování má za účel oddělit nevyúčtovatelné výkony nebo cesty od ostatních, ale zároveň jejich zachování v systému kvůli různým přehledům a statistikám. Stornované výkony a cesty bude každopádně možné přesunout zpět do odmítnutých.

Systém bude také obsahovat stránky se zpracovanými a nezpracovanými vyúčtováními. Ze zpracovaných vyúčtování bude možné znovu stáhnout soubory KDAVKA a FDAVKA do lokálního úložiště, nebo také tyto vyúčtování nastavit jako nezpracované – smazat příslušné soubory KDAVKA a FDAVKA. Naopak u nezpracovaných vyúčtování bude možnost je znovu zpracovat, přičemž se vygenerují nové soubory KDAVKA a FDAVKA.

Jednotlivé faktury, dávky i doklady bude možné zobrazit na samostatných detailních stránkách, které o nich budou zobrazovat veškeré detailní informace.

Vyúčtování

Vyúčtování bude seskupovat faktury vytvořené společně během jednoho procesu tvoření vyúčtování pro jednu kombinaci vstupních parametrů. Jedno vyúčtování bude vždy provedeno pro výkony a cesty právě jedné služby a jednoho časového období. Výstupem vyúčtování budou soubory KDAVKA a FDAVKA pro každou zdravotní pojišťovnu, kterou uživatel při tvoření vyúčtování vybral.

Faktury

Faktury mají být automaticky generovány při tvoření vyúčtování. Bude vygenerována jedna faktura za každou kombinaci zdravotní pojišťovny, druhu pojistného vztahu a měsíce, která obsahuje nějaké nevyúčtované zdravotní výkony nebo cesty za pacienty.

Faktury budou obsahovat základní údaje o službě, pod kterou spadá vykazovaná péče, základní fakturační údaje (datum vystavení, datum splatnosti, fakturovaná částka, atd.) a údaje o smlouvě s cílovou zdravotní pojišťovnou (číslo pojišťovny, druh pojistného vztahu).

Každé faktuře bude při vytvoření vygenerováno číslo, které musí být unikátní v rámci nadřazené služby a roku. Uživatel bude mít možnost toto číslo upravit, pokud zachová unikátnost čísla a faktura ještě nebude součástí zpracovaného vyúčtování. Čísla faktur budou ve formátu RRRRNNN, kde RRRR je rok, ve kterém byly provedeny vykazované zdravotní výkony a cesty spadající pod fakturu, a NNN je pořadové číslo faktury v daném roce v nadřazené službě. První faktura za záznamy z roku 2024 tedy bude mít automaticky vygenerované číslo 2024001.

Dávky

Dávky budou součástí faktur a budou pod sebe seskupovat doklady jednoho typu. Zpravidla budou faktury obsahovat 2 dávky s typy „Dávka poukazů na V/O“ a „Dávka vyúčtování cest“ – každá bude jednoho typu. Dávka typu „Dávka poukazů na V/O“ pod sebe bude seskupovat doklady týkající se vykazovaných zdravotních výkonů a dávka typu „Dávka vyúčtování cest“ bude seskupovat doklady vykazující cesty pracovníků za pacienty.

Dávka bude mít stejně jako faktura vygenerované číslo, které musí být unikátní v rámci roku a zdravotní pojišťovny, za kterou bude nadřazená faktura, a tedy i samotná dávka,

vytvořena. Formát generovaného čísla dávky bude RRNNNN, kde RR je poslední dvojčíslí z čísla roku, ve kterém byly zahrnuté zdravotní výkony či cesty provedeny, a NNNN je pořadové číslo dávky v daném roce pro danou zdravotní pojišťovnu. První dávka za záznamy z roku 2024 tak bude mít vygenerované číslo 240001.

Kromě unikátního čísla dávky bude dávka obsahovat například informace o již zmíněném typu dávky, charakteru dávky nebo verzi datového rozhraní.

Doklady

Doklady budou spadat pod dávky a podle typu nadřazené dávky se budou lišit jejich vlastnosti.

Doklad pod dávkou typu „Dávka poukazů na V/O“ bude obsahovat informace o vykazovaných výkonech. Při generování dokladů tohoto typu jsou nejdříve vykázané výkony rozděleny podle složek pacientů, na kterých jsou provedeny, a poté jsou ještě rozděleny podle poukazů, v rámci kterých jsou výkony provedeny. Každý doklad tak obsahuje skupinu výkonů, které byly provedeny na jednom pacientovi v rámci jednoho poukazu. Cena tohoto typu dokladu bude vypočítána vynásobením bodů za provedené výkony cenou za bod, kterou má daná služba s cílovou zdravotní pojišťovnou ve smlouvě. Počty bodů za jednotlivé výkony uvádí číselník zdravotních výkonů od VZP³.

Doklad dávky typu „Dávka vyúčtování cest“ bude uvádět informace o absolvovaných cestách zdravotních pracovníků za pacienty. Samotné informace o cestách budou součástí dat návštěv, které budou pod doklady tohoto typu seskupovány. Doklady budou generovány zvláště za každého pracovníka, který vykazované cesty absolvoval. Výpočet ceny za tyto doklady se bude odvíjet od parametrů smlouvy uzavřené mezi nadřazenou službou a cílovou zdravotní pojišťovnou. Smlouva může být nastavena třemi způsoby:

- Paušál za návštěvu – cena za doklad se vypočítá vynásobením počtu návštěv a smlouvenou paušální částkou za jednu návštěvu. Nezáleží na ujetých kilometrech pracovníků.
- Paušál za měsíc – za všechny cesty za jedním pacientem během jednoho měsíce je účtována smlouvená částka nezávisle na počtu návštěv nebo ujetých kilometrech.
- Podle ujetých kilometrů – cenu za doklad tvoří smlouvená částka za jeden ujetý kilometr vynásobená ujetými kilometry.

Obecně pro doklady obou typů platí pravidlo, že nemohou obsahovat více než 99 záznamů a zároveň záznamy z jednoho dne nemohou být rozděleny do více dokladů. Při přesazení limitu 99 záznamů v jednom dokladu bude potřeba doklad rozdělit na více dokladů.

Také doklady mají mít svoje číslo, které musí být unikátní v rámci roku a zdravotní pojišťovny. Číslo dokladu bude prosté pořadové číslo začínající od 1. Vedle čísla dokladu budou doklady obsahovat ještě sériové číslo, které bude určovat jejich pořadí v rámci dávky, do které patří.

Provedené výkony

Provedené výkony jsou součástí návštěv pacientů a budou patřit pod doklady spadající pod dávky typu „Dávka poukazů na V/O“. Provedený výkon bude obsahovat informace

³Aktuální číselník zdravotních výkonů ve formátu pdf lze stáhnout zde: <https://www.vzp.cz/poskytovatele/ciselniky/zdravotni-vykony>

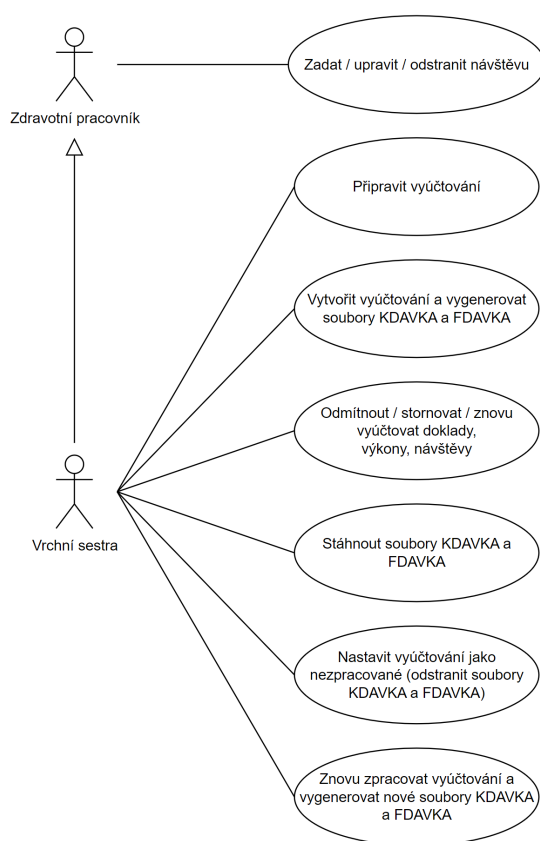
o výkonu (kód, název, délka trvání, cena, atd.), počet provedení, datum provedení převzaté z nadřazené návštěvy a stav vyúčtování.

Návštěvy

Návštěvy budou spadat pod doklady dávek typu „Dávka vyúčtování cest“. Z dat návštěv budou pro účely vyúčtování důležitá data o cestách pracovníků – odkud a kam pracovník cestoval, kolik bylo ujetu kilometrů a kdo cestu vykonal. Návštěvy budou také ukládat informace o složce pacienta, pro kterého byla návštěva provedena, a to primárně kvůli informaci o jeho zdravotní pojišťovně, aby jí mohla být cesta vykázána. Návštěvy budou také pro účely vyúčtování obsahovat informaci o stavu vyúčtování.

Diagram případů užití

Výsledkem analýzy požadavků na funkcionalitu vyúčtování je následující diagram případů užití.



Obrázek 4.1: Diagram případů užití funkcionality vykazování zdravotní péče

Diagram na obrázku 4.1 zobrazuje 2 typy uživatelů. Jedním je běžný zdravotní pracovník, který provádí návštěvy u pacientů a zadává, upravuje a odstraňuje je ze systému. Dalším typem uživatele je zde uvedená vrchní sestra, ale může to být jakýkoliv uživatel s právy k přístupu do sekce vyúčtování. Vrchní sestry budou obsluhovat veškeré funkce související s vyúčtováním, které jsou již uvedené na začátku této sekce 4.3 a v diagramu jsou shrnuté.

Kapitola 5

Návrh informačního systému

Tato kapitola nejdříve popisuje návrh základní struktury systému XCarol, na které je celý systém postaven. Následně je podrobně popsán návrh funkcionality pro vykazování zdravotní péče.

5.1 Návrh základní struktury systému XCarol

Návrh základní struktury informačního systému XCarol navazuje na obecné požadavky na systém uvedené v sekci 4.2. Nejedná se přímo o téma této práce, ale pro kontext je potřeba obecnou strukturu systému uvést. Základní entity a vztahy mezi nimi vystihuje ER diagram na obrázku 5.1.

Entity v ER diagramu 5.1 obsahují pro přehlednost pouze nejzákladnější atributy. V následujících odstavcích jsou jednotlivé entity detailněji rozebrány.

Uživatel

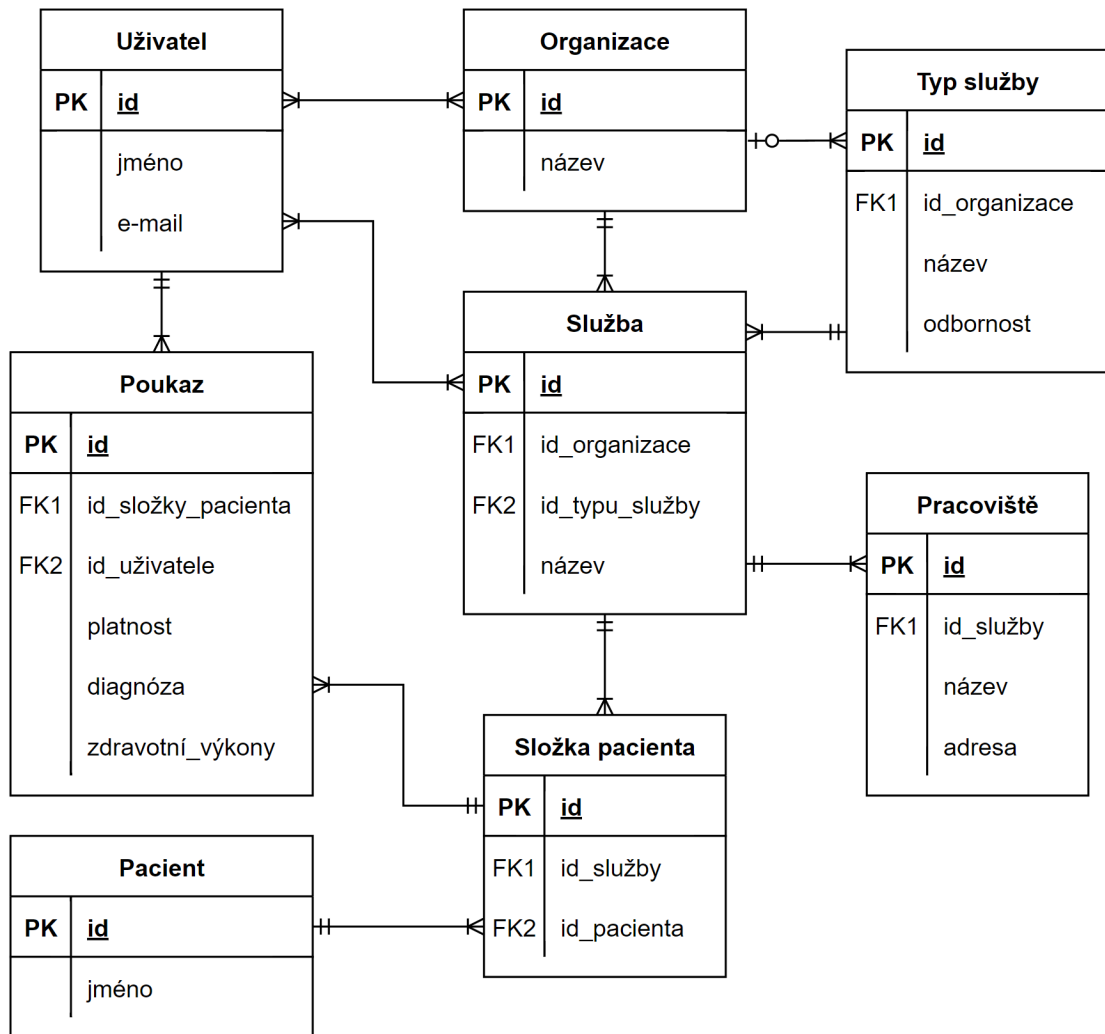
Entita „Uživatel“ představuje kohokoliv, kdo bude systém používat. Většinou to budou lékaři, zdravotní sestry a ostatní zdravotní pracovníci. Mimo uvedené atributy bude entita obsahovat další základní údaje, jako telefonní číslo, adresa bydliště, zda je účet aktivní, zašifrované přístupové údaje, uložené nastavení systému, atp. Mezi „Uživatel“ a „Organizace“, respektive „Služba“, je vztah M:N, který představuje práva uživatelů na přístup do jednotlivých organizací, respektive služeb.

Organizace

„Organizace“ představuje nejvyšší subjekt, který je spravován samostatně nezávisle na ostatních organizacích. Pod organizace spadají jednotlivé služby. Organizace může být například „Charita Prostějov“ a mezi její služby budou patřit například „Zdravotní služba Charity Prostějov“, „Fyzioterapie Charity Prostějov“, atp.

Služba a Pracoviště

„Služba“ představuje jeden z typů zdravotní péče, které její nadřazená organizace poskytuje. Pod službu spadají jednotlivá „Pracoviště“, která tvoří zázemí pro zaměstnance služby. Pracoviště jsou uváděna ve vyúčtování provedených zdravotních výkonů.



Obrázek 5.1: ER diagram základní struktury systému

Typ služby

„Typ služby“ tvoří šablonu pro tvoření nových služeb. Každá služba spadá pod jeden typ služby. Typ služby podřízeným službám udává odbornost, která určuje, jaký typ péče služba poskytuje. Kromě odbornosti obsahuje typ služby veškerá nastavení služby, jejichž změna se bezpodmínečně propíše do nastavení všech podřízených služeb. V diagramu je uvedena nepovinná vazba na organizaci, která značí, že typ služby může, ale nemusí spadat pod organizaci. Pokud typ služby spadá pod nějakou organizaci, lze tento typ služby využít pouze při tvoření služeb v rámci této organizace. V opačném případě může být typ služby použit při tvoření služeb v rámci kterékoliv organizace. Právě z důvodu této nepovinné vazby mezi typem služby a organizací je nutné mít vazbu i přímo mezi službou a organizací, i když je v případě existence vazby mezi typem služby a organizací redundantní.

Složka pacienta

„Složka pacienta“ patří právě k jednomu pacientovi a právě jedné službě. Seskupuje veškerá data o pacientovi relevantní pro službu, ke které patří. Těmito daty mohou být záznamy z provedených vyšetření, lékařské zprávy, atd.

Pacient

„Pacient“ je entita představující osobu, na které je prováděna nějaká zdravotní péče. Kromě uvedeného jména obsahuje podobné základní údaje jako „Uživatel“, k tomu navíc informace týkající se zdravotní pojišťovny, u které je zaregistrovaný (zdravotní pojišťovna, druh pojistného vztahu, číslo pojištěnce), informace o zdravotním stavu a kontaktní osoby. Jeden pacient může mít založených několik složek u různých služeb.

Poukaz

„Poukaz“ je lékařský předpis zdravotní péče na nějaké období. Patří k právě jedné složce pacienta a obsahuje období platnosti, diagnózu a zdravotní výkony, které lze v rámci poukazu provádět. „Uživatel“ spojený s poukazem je lékař, který poukaz vydal. K jedné složce pacienta může být připojeno klidně několik poukazů, avšak jejich období platnosti se nesmějí překrývat.

Návrh uživatelského rozhraní

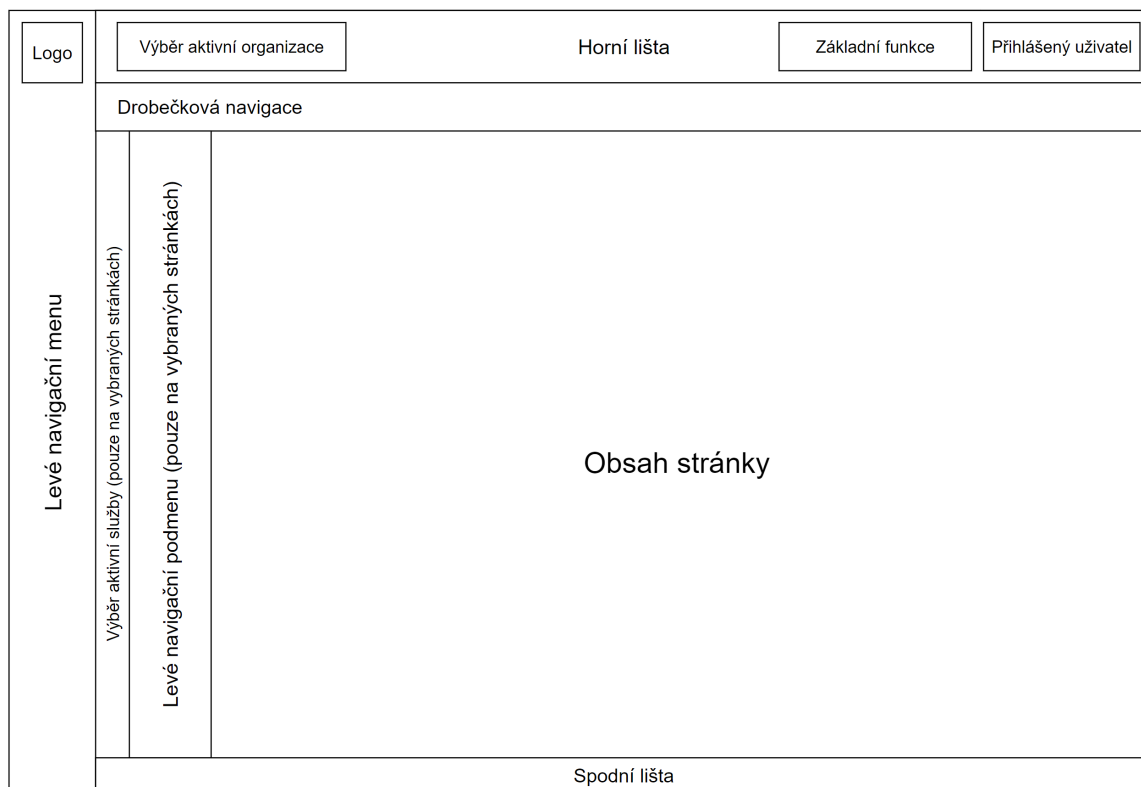
Základní rozhraní systému bude tvořit levé menu, horní lišta a spodní lišta, které jsou popsány v následujících odstavcích.

Levé menu bude navigační a bude sloužit pro navigaci uživatele mezi hlavními stránkami systému. Bude obsahovat základní kategorie (například Nástěnka, E-mailová schránka, Zdravotní služby, Nastavení), které se budou kliknutím rozbalovat na seznam jednotlivých stránek dané kategorie (například kliknutím na Zdravotní služby se rozbalí seznam odkazů na stránky Složky, Zadat práci, Plánovací kalendář, Vyúčtování a další). Některé stránky systému budou umožňovat navigaci do velkého množství podstránek (například detail složky pacienta, který se bude skládat ze stránky se základními informacemi, stránky s kontakty, stránky s poukazy a mnoha dalších), a proto se bude na těchto stránkách zobrazovat podmenu, které bude ležet vpravo vedle levého menu. V případě stránek s daty spadajícími pod nějakou službu (například zmíněný detail složky pacienta) se bude název této služby zobrazovat vertikálně mezi hlavním levým menu a prvním podmenu. Bude zde také možnost jejího přepnutí. Nad kategoriemi v levém menu bude umístěno logo systému odkazující na hlavní stránku.

Horní lišta bude ve směru zleva obsahovat výběr aktivní organizace, který, jak vyplývá z analýzy 4.2, musí stále uživatele informovat, v jaké organizaci se právě nachází a dávat mu možnost ji změnit. Dále budou na horní liště obecné funkce jako nápověda, vyhledávání, zvětšení na celou stránku, notifikace a poslední navštívené stránky. Také zde bude zobrazený aktuálně přihlášený uživatel s možností přesunu na jeho profil a nastavení. Pod horní lištou, ale stále její součástí, bude drobečková navigace zobrazující hierarchii aktuálně otevřené stránky s možností navigace na některou z nadřazených stránek.

Spodní lišta bude obsahovat základní obecné informace o systému (serverový čas, autorství, atp.).

Umístění jednotlivých prvků základu uživatelského rozhraní shrnuje wireframe na obrázku 5.2.



Obrázek 5.2: Wireframe základního uživatelského rozhraní systému XCarol

5.2 Návrh vykazování zdravotní péče

Návrh samotného vyúčtování navazuje na požadavky na vykazování zdravotní péče uvedené v sekci 4.3.

Základní návrh entit a jejich vztahů je zachycen v ER diagramu na obrázku 5.3.

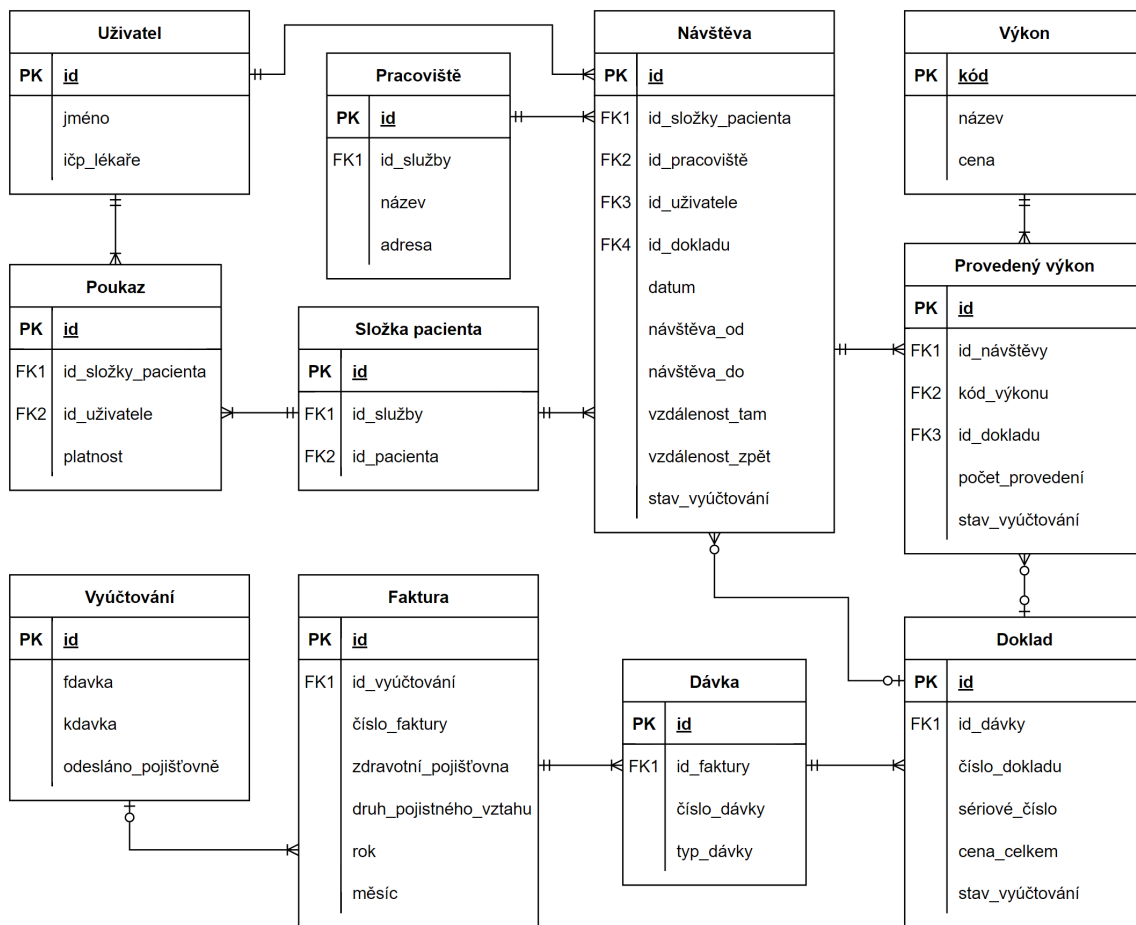
ER diagram 5.3 se s předchozím diagramem 5.1 překrývá v několika entitách zobrazených v 5.3 vlevo nahoře, které zde bylo nutné uvést znovu, protože se přímo týkají vykazování. Stejně jako v předchozím ER diagramu, i v tomto diagramu obsahují entity pouze základní atributy.

Uživatel, Složka pacienta, Poukaz a Pracoviště

Entity „Uživatel“, „Složka pacienta“, „Poukaz“ a „Pracoviště“ a vztahy mezi nimi jsou totožné jako v předchozím diagramu 5.1. „Uživatel“, „Složka pacienta“ a „Pracoviště“ jsou napojeny na návštěvu.

Návštěva

„Návštěva“ představuje jedno navštívení pacienta zdravotníkem. Pacient je představován entitou „Složka pacienta“ a zdravotník entitou „Uživatel“. Váže se také na pracoviště, ve kterém daný zdravotník aktuálně působí. Pro vyúčtování cest jsou důležité atributy „cesta



Obrázek 5.3: ER diagram části systému týkající se vykazování zdravotní péče

tam“ a „cesta zpět“. Informace o čase, od kdy do kdy návštěva trvala, jsou důležité pro nadřazené pracovníky, aby měli přehled o svých zaměstnancích.

Provedený výkon a Výkon

Během návštěvy jsou prováděny zdravotní výkony, které definuje VZP číselníkem. Každý „Výkon“ je identifikován unikátním kódem a mimo něj obsahuje informace o názvu, ceně nebo například odbornosti, pod kterou spadá. Entita „Provedený výkon“ představuje jeden zdravotní výkon provedený během jedné návštěvy. Tento výkon může být v některých případech proveden i vícekrát, což je důvod pro existenci atributu „Počet provedení“. Každý „Provedený výkon“ samostatně ukládá stav vyúčtování, protože uživatel může vyúčtované výkony odmítat jednotlivě.

Doklad

„Doklad“ je hierarchicky nejnižší složkou vyúčtování. Jeho obsah je dán typem dávky, pod kterou spadá. Typ nadřazené dávky určuje, jestli je aktivní vztah „Doklad“ – „Provedený výkon“, nebo „Doklad“ – „Návštěva“. Pokud je typ dávky „Dávka poukazů na V/O“, tak pod sebe doklad zahrnuje provedené výkony (je aktivní první z uvedených vztahů), kde každý doklad obsahuje pouze výkony provedené na jednom pacientovi v rámci jednoho

poukazu. Pokud je naopak typ nadřazené dávky „Dávka vyúčtování cest“, tak pod sebe každý takovýto doklad sdružuje návštěvy provedené jedním zdravotním pracovníkem (je aktivní druhý z uvedených vztahů). Více o teorii tvoření dokladů pojednává podkapitola „Doklady“ v sekci 4.3.

Dávka

„Dávka“ seskupuje doklady jednoho typu určeného atributem „typ dávky“. Tento typ může nabývat dvou hodnot: „Dávka poukazů na V/O“ a „Dávka vyúčtování cest“. Dvě dávky opačných typů společně spadají pod jednu fakturu a jejich data tvoří dohromady obsah souboru KDAVKA. Více o dávkách uvádí podkapitola „Dávky“ v sekci 4.3.

Faktura

Pod „Faktury“ spadají typicky 2 dávky odlišných typů. Faktury jsou tvořeny zvláště za každou zdravotní pojišťovnu, druh pojistného vztahu a měsíc, což je důvod k uvedení těchto atributů v diagramu. Data faktur tvoří obsah souboru FDAVKA. O fakturách více pojednává podkapitola „Faktury“ v sekci 4.3.

Vyúčtování

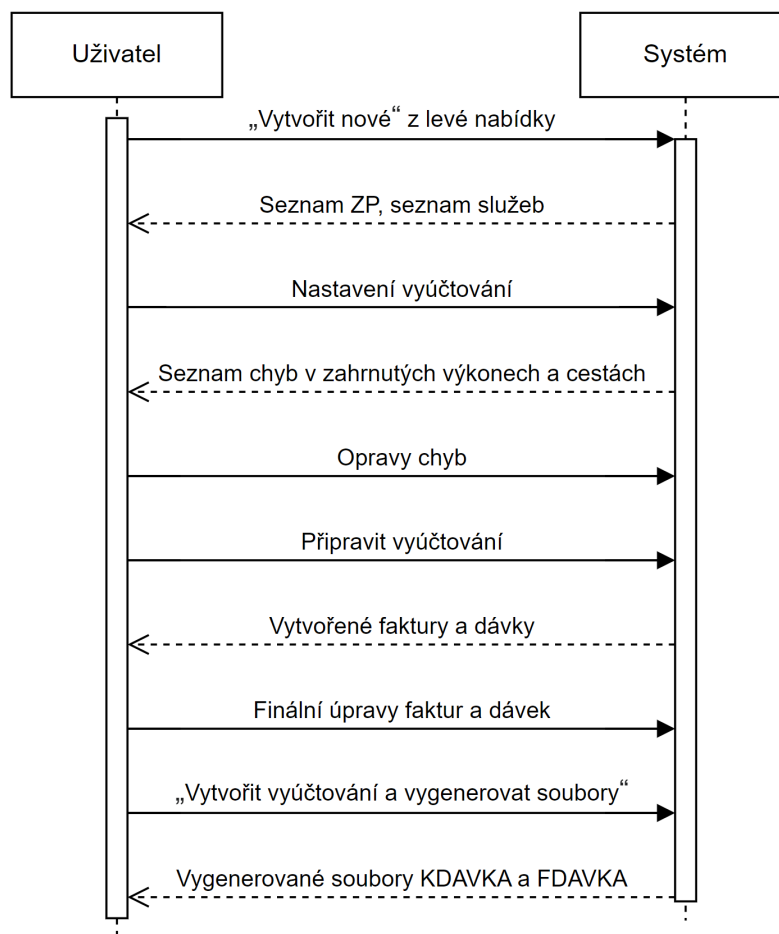
„Vyúčtování“ zahrnuje všechny faktury vytvořené společně jedním procesem vytvoření vyúčtování, které byly vytvořeny pro jednu zdravotní pojišťovnu. Entita „Vyúčtování“ je nad fakturami vytvořena ve chvíli, kdy jsou pro tyto faktury a dávky vygenerovány soubory FDAVKA a KDAVKA, které jsou poté v této entitě uloženy. Faktury, které spadají pod nějaké „Vyúčtování“, jsou implicitně označeny jako zpracované (jejich data jsou zahrnuta v souborech FDAVKA a KDAVKA), naopak faktury bez nadřazeného „Vyúčtování“ jsou nezpracované (jejich data ještě nejsou součástí žádných souborů FDAVKA a KDAVKA). Nezpracované faktury a jejich entity lze upravovat, zpracované faktury a jejich entity ne.

Návrh případů užití

Funkce spojené s vyúčtováváním budou dostupné přes levé menu systému a zobrazení této položky bude řízeno právy přihlášeného uživatele. Po kliku na položku „Vyúčtování“ v levé nabídce se rozbalí seznam jednotlivých stránek, které budou s vyúčtováním souviset. Těmito stránkami budou „Vytvořit nové“, „Odmítnuté doklady“, „Nevyúčtovaná práce“ a „Vytvořená vyúčtování“. Dále bude systém obsahovat stránky detailů jednotlivých složek vyúčtování, tedy stránky „Detail faktury“, „Detail dávky“, „Detail dokladu“. Všechny tyto stránky popisují následující sekce.

Vytvořit nové vyúčtování

Stránka „Vytvořit nové“ slouží pro vytvoření nového vyúčtování. Celou sekvenci kroků zobrazuje diagram 5.4.



Obrázek 5.4: Systémový diagram sekvence kroků vykonávaných uživatelem a systémem při tvoření nového vyúčtování

Prvním krokem uživatele v diagramu 5.4 je kliknutí na „Vytvořit nové“ v levé nabídce systému. Systém následně poskytne nastavení, s jakým se má vyúčtování tvořit, a to: výběr zdravotních pojišťoven, výběr služby, výběr časového období a typ vyúčtování. Jakmile jsou všechny tyto výběry vyplněny, tak systém zareaguje základními kontrolami výkonů a cest, které odpovídají uživatelským vstupům. Tyto kontroly se spouští vždy znovu při jakékoliv změně v některém z výběrů. Pokud jsou při kontrolách nalezeny chyby, tak je systém vypíše. Každou chybu může uživatel rozkliknout a zobrazí se mu návštěvy, kterých se chyba přímo týká. Některé jednodušší chyby může uživatel opravit rovnou na aktuální stránce úpravou návštěv. Po úpravě se znovu provedou kontroly. Dokud bude systém detekovat nějaké chyby, nepovolí pokračování na další krok tvoření vyúčtování. Jakmile nejsou pro uživatelem vybrané zadání pro vytvoření vyúčtování detekovány žádné chyby, zobrazí se tlačítko „Připravit vyúčtování“.

Po kliku na „Připravit vyúčtování“ systém na pozadí vygeneruje kompletní strukturu vyúčtování (všechny potřebné faktury, dávky a doklady) a uživateli je vypíše. Uživatel si může detaily jednotlivých entit prohlédnout a některé parametry i upravit (například čísla faktur a dávek nebo data splatností faktur). Jakmile bude uživatel s aktuální podobou

faktur, dávek a dokladů spokojený, tak klikne na tlačítko „Vytvořit vyúčtování a vygenerovat soubory“.

V tuto chvíli systém na pozadí vygeneruje obsahy souborů FDAVKA a KDAVKA a nabídne je uživateli ke stažení do lokálního úložiště. Obsažené faktury jsou od této chvíle zahrnuty pod entitu „Vyúčtování“, jsou tak implicitně označeny jako zpracované a zahrnutým výkonům a návštěvám je nastaven stav vyúčtování na „Vyúčtované“.

Odmítnuté doklady

Stránka „Odmítnuté doklady“ slouží pro manuální měnění stavů vyúčtování u jednotlivých vyúčtovaných výkonů a cest. Skládá se celkem z 5 podstránek: „Vyhledání dokladu“, „Odmítnuté výkony“, „Odmítnuté návštěvy“, „Stornované výkony“ a „Stornované návštěvy“. Všechny tyto podstránky sdílí výběr služby, který slouží pro přesnější filtrování zobrazených záznamů.

Při otevření stránky je automaticky aktivní podstránka „Vyhledání dokladu“. Ta slouží k vyhledání dokladů a jejich výkonů, či návštěv (cest) se stavem „Vyúčtovaný“. Vyhledávání se provádí pomocí již zmíněného výběru služby dostupného pro všech 5 podstránek, dále pomocí zdravotní pojišťovny, typu dávky, čísla dávky a volitelně lze vyhledávat i podle čísla dokladu. Všechna tato vyhledávací pole jsou výběry a čísla dávek a dokladů se průběžně načítají podle ostatních výběrů. Po vyplnění výběrů se zobrazí seznam dokladů a provedených výkonů (při vybraném typu dávky „Dávka poukazů na V/O“), nebo návštěv (při vybraném typu dávky „Dávka vyúčtování cest“). Jakmile jsou takto záznamy zobrazeny, je možné je buď jednotlivě, nebo hromadně odmítnout (nastavit jejich stav vyúčtování na „Odmítnutý“).

Podstránky „Odmítnuté výkony“ a „Odmítnuté návštěvy“ zobrazují všechny výkony, respektive návštěvy, spadající pod vybranou službu a se stavem vyúčtování „Odmítnutý“. Uživatel zde může odmítnuté záznamy buď nastavit zpět na vyúčtované, nebo je stornovat. Při stornování se ještě zobrazí modální okno s výběrem důvodu stornování, který bude k daným stornovaným výkonům, respektive návštěvám, uložen.

Podstránky „Stornované výkony“ a „Stornované návštěvy“ zobrazují všechny výkony, respektive návštěvy, spadající pod vybranou službu a se stavem vyúčtování „Stornovaný“. Je zde možnost zobrazené záznamy převést zpět na odmítnuté.

Nevyúčtovaná práce

Stránka „Nevyúčtovaná práce“ slouží pro přehled všech návštěv, které ještě nebyly vyúčtovány. Seznam nevyúčtovaných prací je možné filtrovat pomocí výběru služby a výběru datového rozmezí, ve kterém byly návštěvy vykonány.

Vytvořená vyúčtování

Po kliku na položku „Vytvořená vyúčtování“ v levém menu, se tato položka ještě jednou rozbálí a zobrazí se čísla jednotlivých zdravotních pojišťoven, pro které již bylo vytvořeno alespoň jedno vyúčtování. Uživatel si ze zobrazených zdravotních pojišťoven jednu vybere a stránka „Vytvořená vyúčtování“ se otevře s daty relevantními právě pro tuto pojišťovnu.

Stránka „Vytvořená vyúčtování“ se skládá ze 3 podstránek: „Dávky poukazů na V/O“, „Dávky vyúčtování cest“ a „Faktury“. Všechny tyto stránky sdílí výběr služby pro přesnější filtrování, základní informaci o pojišťovně, kterou uživatel vybral v levém menu před

otevřením stránky, a tlačítko odkazující na stránku „Nezpracovaná vyúčtování“, která je popsána v další podkapitole.

Podstránky „Dávky poukazů na V/O“ a „Dávky vyúčtování cest“ obsahují seznamy dávek, které jsou součástí zpracovaných vyúčtování, které spadají pod vybranou službu. Každá z těchto stránek obsahuje dávky typu uvedeného v názvu této podstránky. Uživatel si odsud může otevřít detaily jednotlivých dávek.

Podstránka „Faktury“ zobrazuje seznam zpracovaných vyúčtování spadajících pod vybranou službu. Je odsud možné stáhnout si soubory KDAVKA a FDAVKA, které byly k danému vyúčtování vygenerovány, do lokálního úložiště, aby je mohl uživatel odeslat zdravotní pojišťovně. Uživatel zde také může zrušit vyúčtování, což způsobí smazání vygenerovaných souborů KDAVKA a FDAVKA a přesun všech spadajících faktur a jejich součástí do nezpracovaných vyúčtování.

Nezpracovaná vyúčtování

Na stránku „Nezpracovaná vyúčtování“ se lze dostat ze stránky „Vytvořená vyúčtování“, což znamená, že je automaticky aktivní filtrování podle zdravotní pojišťovny, kterou uživatel zvolil v levé nabídce při otvírání vytvořených vyúčtování.

Stránka „Nezpracovaná vyúčtování“ poskytuje vedle implicitního filtrování podle zdravotní pojišťovny také filtrování pomocí výběru služby. Zobrazují se zde všechny faktury s jejich dávkami, které nejsou součástí žádného zpracovaného vyúčtování, a tedy jejich data nejsou součástí žádných souborů KDAVKA a FDAVKA. Je možné si tu proklikat detaily jednotlivých faktur a dávek a také je možné jednotlivé faktury nebo dávky trvale odstranit. Pod všemi těmito seznamy se nachází tlačítko „Zpracovat a vytvořit soubory KDAVKA a FDAVKA“, které způsobí, že se faktury a dávky znovu zaobalí do zpracovaného vyúčtování a vygenerují se nové soubory KDAVKA a FDAVKA.

Detail faktury

Detail faktury obsahuje veškeré informace o jedné faktuře. Obsahuje základní informace jako číslo faktury, datum vystavení, datum splatnosti, rok, měsíc a další. Pokud je faktura nezpracovaná (její data nejsou součástí žádného souboru FDAVKA), tak je možné tyto údaje měnit, jinak jsou zobrazeny pouze staticky pro přehled.

Dále detail faktury obsahuje seznam dávek, které pod fakturu spadají. Přes tento seznam se lze dostat na detaily jednotlivých dávek. Nakonec se ještě na stránce zobrazuje fakturovaná částka rozepsaná pro jednotlivá pracoviště.

Detail dávky

Detail dávky uživateli poskytuje veškerá data jedné dávky. Zobrazuje rozsáhlé informace o dávce, kde některé je možné měnit, pokud není nadřazená faktura součástí zpracovaného vyúčtování.

Na stránce se také nachází seznam dokladů spadajících pod dávku. Ze seznamu si lze otevřít detaily jednotlivých dokladů, detaily složek pacientů, detaily poukazů nebo detaily zdravotních pracovníků.

Detail dokladu

Obsah stránky „Detail dokladu“ se liší typem nadřazené dávky.

Pokud je typ dávky „Dávky poukazů na V/O“, tak stránka zobrazuje mnoho základních informací o dokladu, nadřazené dávce a faktuře, seznam provedených výkonů, které do dávky patří a ve kterém je možné vyhledávat podle potřeby, a nakonec také seznam diagnóz převzatý z příslušného poukazu.

Pokud je typ dávky „Dávka vyúčtování cest“, tak stránka kromě podobných základních informací, jako v předchozím případě, obsahuje seznam zahrnutých návštěv.

Kapitola 6

Implementace

Kapitola implementace popisuje konkrétní použité technologie a postupy, které přispívají k převodu teoretického návrhu systému na funkční software. Nejprve je popsána implementace základní struktury systému XCarol a následně implementace funkcionality vyúčtování, která je vystavěná na základní struktuře systému a je hlavním tématem této práce.

6.1 Implementace základní kostry systému XCarol

Implementace základní kostry systému navazuje na její návrh ze sekce 5.1. V následujících odstavcích jsou popsány použité technologie, pomocné třídy vytvořené speciálně pro vývoj tohoto systému a práce s nimi. Žádná ze zde popsaných pomocných tříd nebyla součástí implementace této diplomové práce, ale všechny byly při implementaci používány.

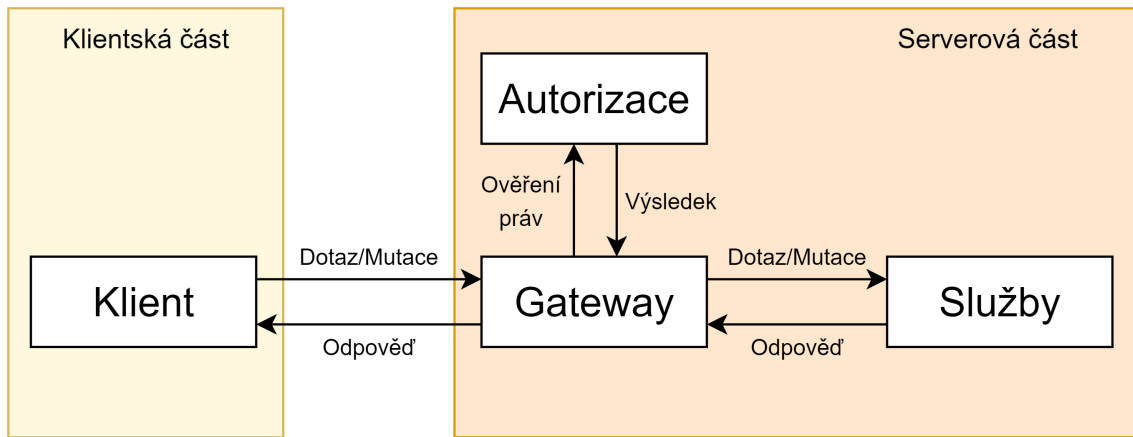
Použité technologie

Použité technologie v této diplomové práci kopírují technologie, kterými je vytvořena základní kostra systému XCarol, a nebylo proto možné si je k implementaci této práce svobodně vybrat. Celý systém XCarol je tvořen v programovacím jazyce TypeScript. Pro klientskou část systému (frontend) je použit framework Angular s využitím knihovny komponent PrimeNG a CSS knihovny PrimeFlex. Pro serverovou část systému (backend) je použito serverové prostředí Node.js s frameworkem Nest.js. Jako databázový systém je použito PostgreSQL. Pro práci s databází v rámci serverové části systému je využit framework TypeORM. K propojení klientské a serverové části slouží GraphQL. Systém a jeho závislosti jsou kontejnerizovány pomocí Dockeru, takže jednotlivé části systému běží odděleně v samostatných kontejnerech. Jednotlivé kontejnery (balíčky) jsou popsány v následující podsekci.

Základní struktura systému XCarol

Systém se skládá z několika oddělených balíčků. Na obrázku 6.1 je znázorněn tok dat mezi balíčky, které jsou podstatné pro tuto práci a jsou následně popsány níže.

V grafu na obrázku 6.1 je znázorněno, že komunikace je iniciována dotazem, nebo mutací. Dotaz a mutace jsou 2 typy požadavků klientské části na serverovou část. Dotaz (query) se používá pro operace, které mají za cíl získání dat z databáze bez jakýchkoli úprav databáze, například načtení složek pacientů, aby mohly být vypsány uživateli. Naopak mutace



Obrázek 6.1: Graf základní vzájemné komunikace balíčků systému

(mutation) se používá pro operace, jejichž účel je upravit data v databázi, tedy typicky vytvoření, úprava a odstranění nějaké entity.

Klient

Balíček „Klient“ obsahuje celou klientskou část systému, která je vytvořena ve frameworku Angular. Nachází se zde zdrojové kódy jednotlivých stránek a jejich komponent a třídy starající se o komunikaci se serverovou částí systému.

Vše, co uživatel systému vidí na obrazovce, jsou komponenty. Komponenta je ve frameworku Angular základním stavebním kamenem uživatelského rozhraní. Komponenta je část uživatelského rozhraní složená z libovolného množství jiných komponent a základních HTML prvků a může být opětovně používána v různých částech systému. Speciálním případem komponenty je komponenta celé jedné stránky, která se ale od ostatních komponent liší pouze tím, že je v navigačním modulu přiřazena nějaké url adrese. Zdrojové kódy jedné komponenty systému jsou tvořeny typicky čtyřmi soubory, které se vždy nacházejí společně v samostatném adresáři. Jejich přípony a účely uvádí následující body:

- `.component.ts` – základní soubor každé komponenty. Je zde uveden název komponenty, který ostatní komponenty použijí, pokud chtějí její obsah vložit do svého. Komponenta je v tomto souboru definována jako třída, která uvádí odkazy na ostatní soubory zmíněné v následujících bodech a obsahuje definice vstupů a výstupů, atributů a metod, včetně jejich implementace. Vstupy a výstupy slouží pro komunikaci komponenty s rodičovskou komponentou, což je komponenta, která ji zahrnuje do svého obsahu. Atributy a metody třídy mohou sloužit pro předzpracování dat pro jejich následné zobrazení.
- `.component.html` – soubor, ve kterém je pomocí základních HTML prvků, PrimeNG komponent a komponent vytvořených v rámci tohoto systému definován vizuální obsah komponenty. Data z příslušné třídy popsané v předchozím bodě lze v HTML přímo zobrazovat pomocí dvojitého složených závorek. Jednotlivým prvkům lze díky frameworku Angular nastavit podmíněčné zobrazení (pomocí `*ngIf`), nebo jich vložit několik cyklicky za/pod sebe (pomocí `*ngFor`).

- `.component.scss` – soubor s kaskádovými styly pro předchozí `.html` soubor. Primárně se však používají třídy definované CSS knihovnou PrimeFlex, takže tento soubor je přítomný pouze pokud je potřeba definovat nějaké specifické CSS třídy.
- `.store.ts` – soubor se stavovým úložištěm komponenty. Slouží pouze pro danou komponentu, žádná jiná do něj nemá přístup. Tvoří jej třída, přes kterou lze přistupovat do globálního stavového úložiště celého systému a na třídu, která odesílá požadavky a přijímá odpovědi od serverové části systému. Ve stavu ve stavovém úložišti komponenty se nacházejí všechna data, která si komponenta vyžádala od serveru. Třída v souboru `.store.ts` dědí všechny tyto zmíněné funkce z třídy `ComponentStoreFactory`, což je pomocná třída implementovaná speciálně pro systém XCarol. Třída `ComponentStoreFactory` zbavuje programátora nutnosti práce s jednotlivými složkami stavového úložiště uvedenými v následujícím odstavci a celý proces zjednodušuje na prosté zavolání metody požadavku a následné získání dat ze stavového úložiště komponenty.

Vedle adresáře s adresáři jednotlivých komponent se nachází adresář obsahující soubory globálního stavového úložiště. Používá se zde RxJS Store, který umožňuje spravovat stav aplikace pomocí reaktivního programování. Základní strukturu tohoto přístupu tvoří:

- Stav – definuje jednotlivé položky stavu aplikace a jejich výchozí hodnoty.
- Akce – jednotlivá možná volání úložiště. Definují názvy jednotlivých volání a jejich vstupní parametry. Patří sem akce volané z komponent a akce volané přímo z ostatních složek úložiště, například při získání odpovědi ze serveru. Například akce pro vyžádání seznamu složek je `getFolders` a po úspěšném vrácení všech složek v odpovědi je zavolána akce `getFoldersSuccess`.
- Efekty – jsou navázány na jednotlivé akce. Jakmile je daná akce zavolána, efekt obstará zavolání metody odpovídající požadavku volané akce požadavku, požadavek se pošle na server a přijme se odpověď. Následně po přijetí odpovědi efekt zavolá akci, která odpovídá původně volané akci a výsledku volání. Při použití příkladu z minulého bodu bude efekt navázán na akci `getFolders` a při úspěšném přijetí dat zavolá akci `getFoldersSuccess`.
- Reduktory – jsou navázány na akce a stav stavového úložiště. Podle volané akce a jejích parametrů upravuje stav stavového úložiště. Například při zavolání akce `getFolders` nastaví odpovídající reduktor pole složek ve stavu na prázdné a při zavolání `getFoldersSuccess` nastaví jiný odpovídající reduktor pole složek ve stavu na data získaná v odpovědi.
- Selektory – slouží pro získání dat ze stavu stavového úložiště komponentami. Typicky je tedy pro každou položku ve stavu vytvořen vlastní selektor, ale mohou být vytvořeny i selektory navíc, které budou data ze stavu před předáním komponentě ještě nějak upravovat nebo zpracovávat.

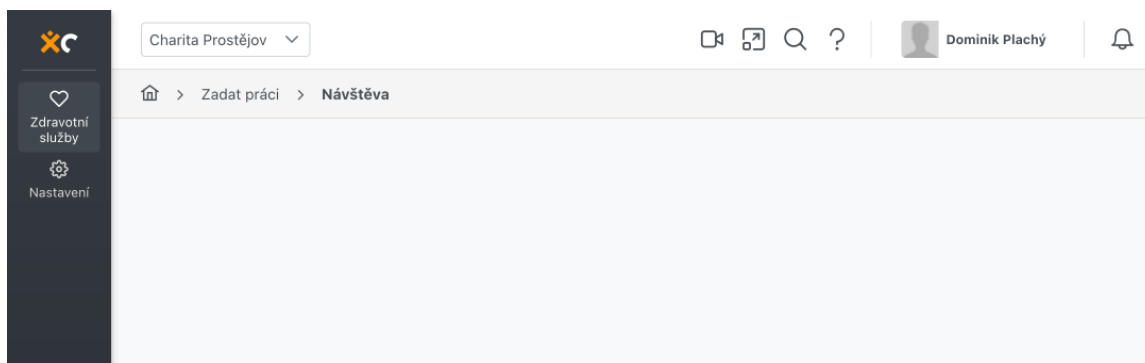
Globální stavové úložiště systému XCarol slouží pro ukládání a načítání základních dat, která jsou potřebná na několika místech. Tato data už nemusejí jednotlivé komponenty znovu načítat, protože jsou přednačtena při každém otevření jakékoliv stránky uživatelem. Ve stavu globálního stavového úložiště jsou ukládány všechny organizace a služby právě

vybrané organizace, ke kterým má přihlášený uživatel přístup. Pro obě tyto položky jsou definovány selektory, takže každá komponenta, která například potřebuje seznam služeb, pouze zavolá selektor `selectServices`.

O samotné volání serverové části aplikace se starají třídy s názvem končícím na `Service`, a to jedna za každý balíček na serveru. Pro tuto práci byl použit jediný balíček, a to „Služby“ (uvedený také na obrázku 6.1). Tento název má z důvodu práce s daty služeb. Třída volající server, která volá koncové body balíčku služeb, tak má název `ServiceService`. Jsou v ní definovány metody volající jednotlivé koncové body, které server poskytuje. Součástí každého volání serveru v těchto metodách je výpis požadovaných polí, které mají přijít v odpovědi, pro GraphQL. Požadavky s definovanou strukturou odpovědi jsou odesílány pomocí třídy `Apollo` z knihovny `apollo-angular`.

Všechny zdrojové kódy, které se týkají výhradně systému XCarol, se nacházejí v podadresáři (modulu) `content`. Zde se nachází soubory základní komponenty, která tvoří plátno pro zobrazení jakýchkoli komponent systému XCarol. Dále je zde soubor `content.module.ts`, který uvádí všechny komponenty, které modul `content` obsahuje, a soubor `content-routing.module.ts`, který definuje všechny dostupné url adresy modulu a přiřazuje k nim komponenty stránek, které se mají v případě navigace na danou url adresu zobrazit.

Uživatelské rozhraní je vytvořeno podle návrhu ze sekce 5.1 a výsledek je zobrazen na obrázku 6.2. Zobrazení levého navigačního menu má celkem 5 různých módů a uživatel si může vybrat, který mu nejvíce vyhovuje. Na obrázku 6.2 je použito úzké zobrazení, při kterém se zobrazují pouze ikony a názvy hlavních kategorií a po kliknutí na ně se zobrazí seznam podstránek. Dalšími možnostmi jsou plné zobrazení menu (rovnou jsou vidět i všechny podstránky každé kategorie), vysouvací menu (stejně jako plné zobrazení, ale vysune se zleva až jakmile uživatel najede myší k levému okraji a při vzdálení myši se opět zasune), extra úzké (jsou zobrazeny pouze ikony hlavních kategorií a po kliknutí se zobrazí podstránky stejně jako u menu na obrázku) a horizontální (menu se zobrazuje nahoře nad horní lištou s výběrem organizace).



Obrázek 6.2: Základ uživatelského rozhraní

Gateway a Autorizace

„Gateway“ tvoří mezikrok mezi klientskou částí a volaným balíčkem serverové části. Jakmile například zmíněná třída `ServiceService` vyšle nějaký požadavek na server, tak tento požadavek nejdříve dorazí na „Gateway“.

Po přijetí požadavku se nejdříve provede kontrola, jestli byl požadavek vůbec vyslán na základě akcí přihlášeného nezablokovaného uživatele. Pokud je toto v pořádku, je ještě ově-

řeno, že má daný uživatel nastavená práva na požadavek, který provádí. Jakmile kterákoliv z těchto kontrol selže, tak „Gateway“ požadavek neposílá dál, ale vrací zpět s chybovým kódem. Pokud není nalezen problém s právy uživatele, tak je požadavek odeslán dál odpovídajícímu balíčku. Do dat požadavku lze ještě přidat další informace o přihlášeném uživateli, ke kterým nemusí mít volající komponenta přístup, například identifikátor přihlášeného uživatele nebo identifikátor vybrané organizace.

Zdrojové kódy v „Gateway“ jsou také rozděleny do adresářů podle jednotlivých balíčků systému. Každý takový adresář obsahuje resolvery, ve kterých se volá zmiňovaná kontrola práv, a objekty dotazů a mutací. Kontrola přihlášení a platnosti účtu uživatele je spuštěna na základě přítomnosti dekorátoru `@UseGuards` a kontrola práv uživatele na danou akci je spuštěna na základě dekorátoru `@UseOperation`, do jehož parametrů se uvede název entity a akce. Například u odstranění složky pacienta je dekorátor `@UseOperation('folder.delete')`.

Balíčky

„Balíčky“, neboli `packages`, jsou serverové části, jejichž úkolem je zpracovávat dotazy od klientské části a pracovat s databází. Na obrázku 6.1 je uveden přímo balíček „Služby“, který je jediným používaným v rámci této práce. Dalšími balíčky, které serverová část systému XCarol obsahuje, jsou například „Organizace“ (`organization`) nebo „Správa uživatelů“ (`user-management`).

Ke každému balíčku, ale mimo jeho adresář, ještě patří soubory, které přesně odpovídají kontrolerům v balíčku a jsou seskupeny do adresáře `<název balíčku>-client`. Tyto soubory přijímají GraphQL požadavky odeslané z „Gateway“ a přeposílají je kontrolerům v balíčku jako požadavky typu `POST`. Účel celého tohoto adresáře je, že „Gateway“ i ostatní balíčky takto mají povědomí o jednotlivých koncových bodech balíčku, aniž by si musely importovat jednotlivé kontrolery.

Každý balíček obsahuje kontrolery, které jako první z celého balíčku přijímají požadavky z klientské části. Typicky se zde nachází jeden kontroler za každou tabulku v databázi. Kontroler volá odpovídající metodu z odpovídající třídy z adresáře `/domain/service` v balíčku. Výsledek je poté buď rovnou odeslán zpět jako odpověď, nebo jsou ještě před odesláním odpovědi výsledná data přemapována do objektů, které klientská část v odpovědi očekává.

V adresáři `/domain/service` se soubory, se kterými kontrolery při zpracování požadavků komunikují. Tyto soubory obsahují třídy, jejichž metody odpovídají koncovým bodům v kontrolerech. Mimo tyto veřejné metody ještě mohou obsahovat různé privátní metody pro zpracování složitějších dotazů nebo odpovědí. Každá takováto třída importuje potřebné třídy repositářů z adresáře `/infrastructure/typeorm/repository`, jejichž metody využívá na jednoduché požadavky na databázi – vytvoř, uprav, odstraň, najdi. Třídy v adresáři `/domain/service` implementují veškerou logiku nad vstupními a vrácenými daty.

Třídy v adresáři `/infrastructure/typeorm/repository` jsou takzvané repositáře a obsahují přímou manipulaci s databází, která bývá prováděna pomocí frameworku TypeORM. Obsahují metody, které provádějí jednoduché procesy s databází a nestarají se o logiku kolem nich. Tyto metody většinou vrací objekty (případně pole objektů) jako výsledky požadavků na databázi.

Kromě obsluhy požadavků na databázi odeslaných z klientské části balíčky definují jednotlivé tabulky, a to pomocí TypeORM modelů. Tyto modely jsou třídy uvedené TypeORM dekorátorem `@Entity`, v jehož parametru se explicitně uvádí název databázové tabulky, která bude tyto entity obsahovat. Uvnitř těchto tříd (modelů) jsou definované jednotlivé

sloupce se všemi jejich vlastnostmi. Také jsou zde uvedeny vztahy s ostatními entitami, a to pomocí dekorátorů `@OneToMany` pro vztah 1:N, `@ManyToOne` pro vztah N:1 a `@ManyToMany` pro vztah M:N. Díky takto definovaným vztahům mezi modely (tabulkami) je snadnější získávání dat z databáze pomocí TypeORM dotazu v případě potřeby spojování více tabulek v jednom dotazu.

V základních konfiguracích balíčků je nastavení, které ovlivňuje, jestli se změny v modelech mají okamžitě synchronizovat s databází. Například přidání sloupce do modelu způsobí okamžité přidání sloupce do odpovídající tabulky. Toto nastavení je sice pohodlné při vývoji systému, ale nese s sebou rizika nechtěné úpravy či odstranění dat. V systému XCarol je proto toto nastavení ve všech balíčcích vypnuto a tabulky databáze se upravují pomocí migrací. Každý balíček má ve své databázi tabulku, do které si ukládá migrace, které již byly provedeny, aby nebyly provedeny znovu. Migrace lze psát ručně, nebo použít TypeORM příkaz v příkazovém řádku, který pro daný balíček porovná aktuální podobu tabulek databáze a modelů a automaticky vytvoří migraci, která tabulky v databázi převede na stav odpovídající aktuálním podobám modelů.

DataGrid

Specialitou systému XCarol, která se týká všech částí popsaných výše, je takzvaný DataGrid. DataGrid výrazně zjednodušuje programování načítání seznamů dat, typicky pro tabulky. Výběr vrácených dat bývá často ovlivněn různými filtry a je potřeba je různě řadit. Navíc je často potřeba seznamy dat stránkovat, aby se najednou nenačítaly tisíce položek a nezatěžoval se tím server a webový prohlížeč uživatele. Proto je pro systém XCarol implementováno několik pomocných tříd a objektů, které splňují všechny zmíněné vlastnosti a dělají jejich konfiguraci jednodušší.

Dotazy využívající výhody DataGridu začínají stejně jako všechny ostatní v klientské části systému. Zde se sestaví objekt typu `IGridRequest`, který obsahuje filtry, řazení a stránkování. Filtry jsou typu `[[IGridRequestFilter]]`, kde jsou implicitně mezi položkami pole první úrovně logické operátory `AND` a mezi položkami pole druhé úrovně logické operátory `OR`. Samotný objekt filtru obsahuje informace o tom, kterého sloupce se týká, jaký operátor se má pro podmínku použít a hodnotu, se kterou se má hodnota daného sloupce podle daného operátoru porovnat. Položka řazení je typu `[[IGridRequestSorter]]`, kde jednotlivé položky jsou jednotlivá řazení, jak mají jít za sebou. Samotný objekt `IGridRequestSorter` obsahuje název sloupce a směr řazení. Stránkování se skládá z čísla stránky, která se má načíst, a počtu položek na jedné stránce.

Objekt vytvořený podle popisu v předchozím odstavci se odešle jako parametr dotazu serverové části. Balíčky obsahují speciálně pro DataGrid adresář `/infrastructure/typeorm/grid`, kde se nacházejí třídy pro jednotlivé entity, na kterých je potřeba provádět dotazy DataGridu. Tyto třídy implementují získání dat z databáze pomocí TypeORM dotazu, v rámci kterého provádí spojování tabulek a selekci potřebných dat. Dále uvádějí názvy možných filtrů, k nim odpovídající sloupce z uvedeného TypeORM dotazu a jednotlivé možnosti řazení také spolu se sloupci z TypeORM dotazu, kterých se dané řazení skutečně týká. Filtrování a řazení tak není nutně potřeba provádět pouze nad sloupci dané tabulky, ale také nad sloupci připojených tabulek nebo nad daty zpracovanými pomocí PostgreSQL funkcí.

Třídy z podadresáře `/infrastructure/typeorm/grid` dědí z třídy `TypeORMGrid`, která se stará o převod nastavení z třídy `gridu` a vstupních dat na dotaz do databáze, a to včetně stránkování. Třída `TypeORMGrid` také výsledná data obalí do objektu `<název en-`

tity>GridObject, který obsahuje pole výsledných položek, použité filtry, řazení a informace o stránkování – číslo aktuálně vrácené stránky, počet položek na jedné stránce, celkový počet položek v databázi, které odpovídaly zadaným filtrům, a číslo poslední stránky.

Výsledný objekt se vrátí jako odpověď klientské části, kde si z něj daná komponenta typicky vezme vrácené položky, nebo je objekt přímo vložen do komponenty tabulky, která je uzpůsobena pro práci s těmito typy objektů a automaticky na základě obsažených informací zobrazí položky a pod nimi dostupné stránkování, včetně nastavení počtu položek na stránku.

6.2 Implementace vykazování zdravotní péče

Vykazování zdravotní péče hierarchicky spadá pod služby, takže všechny zdrojové kódy implementující cokoliv ohledně vykazování jsou umístěny do adresářů `services`, a to v klientské části, v gateway i v serverové části, kde se takto dokonce nazývá celý jeden balíček a jeho databáze. Všechny webové stránky, které patří k vykazování, mají v url adrese prefix `/billing`.

Databázové tabulky pro entity „Vyúčtování“, „Faktura“, „Dávka“ a „Doklad“ byly vytvořeny a napojeny na upravené stávající entity přesně podle ER diagramu na obrázku 5.3. Následující podsekcce popisuje implementaci samotného vytvoření vyúčtování a další podsekcce pak popisuje implementaci ostatních funkcí a stránek, které byly také v rámci této diplomové práce vytvořeny.

Implementace vytvoření vyúčtování

Každá implementace nějaké funkcionality v tomto projektu se skládá z oddělených implementací klientské a serverové části.

Klientská část vytvoření vyúčtování se skládá celkem ze dvou webových stránek. Na první stránce uživatel nastaví parametry pro přípravu vyúčtování a na druhé si prohlédne celou strukturu vygenerovaného vyúčtování, případně upraví detaily a potvrdí vytvoření vyúčtování.

Serverová část vytvoření vyúčtování musí kromě běžných úkolů (například poskytnutí seznamu služeb) vybrat všechny návštěvy spadající do vybraných parametrů, všechny tyto návštěvy otestovat, jestli neporušují nějaká pravidla pro úspěšné vyúčtování, a následně na základě těchto návštěv vygenerovat faktury, dávky a doklady. Po potvrzení vytvoření vyúčtování uživatelem musí serverová část potvrzené faktury, dávky a doklady zahrnout do vytvořeného vyúčtování, všem dokladům, provedeným zdravotním výkonům a cestám nastavit stav vyúčtování na „Vyúčtovaný“ a v poslední řadě vygenerovat soubory KDAVKA a FDAVKA a poskytnout je uživateli ke stažení do lokálního úložiště.

V následujících podsekcích budou implementace všech těchto uvedených stránek v klientské části a funkcí v serverové části více rozebrány.

Klientská část přípravy vyúčtování

Přípravu vyúčtování provádí uživatel na stránce, která se otevře po kliknutí na „Vytvořit nové“ v kategorii „Zdravotní služby“ a podkategorii „Vyúčtování“ v levém menu. Url adresa této stránky končí na `/create`. Tuto konkrétní stránku tvoří komponenta `BillingNewCreatePageComponent` a slouží pro přípravu vyúčtování. Vzhled stránky se všemi jejími funkcemi zobrazuje obrázek 6.3. Samotné vytvoření vyúčtování pak probíhá na další

stránce, která je hierarchicky podstránkou stránky pro přípravu vyúčtování a je popsána v následující podsekcí.

Charitní ošetřovatelská služba Žižkova 3

Pátek 13.2024 - Neděle 31.3.2024

Nevyúčtovaná práce v předchozích obdobích

Řádné vyúčtování | Dodatečné vyúčtování | Opravné vyúčtování

111 - Všeobecná zdravotní pojišťovna České republiky
 201 - Vojenská zdravotní pojišťovna České republiky
 205 - Česká průmyslová zdravotní pojišťovna
 207 - Oborová zdravotní pojišťovna zaměstnanců bank a pojišťoven
 209 - Zaměstnanecká pojišťovna ŠKODA
 211 - Zdravotní pojišťovna Ministerstva vnitra ČR
 213 - RBP, zdravotní pojišťovna
 333 - Pojišťovna VZP, a.s.

Výpis chyb ve vyúčtování

Evženie Neumann
 Výkon 06321 může být zadán pouze jednou za 7 dní.

Jitka Kerekešová
 Výkon 06349 může být zadán maximálně jednou za den.

Oprava vybrané chyby ve vyúčtování

	Návštěva od	Návštěva do	Délka návštěvy	Délka výkonů	Vybrané výkony na návštěvě	Pracovník	Adresa poskytování	Složka klienta	ZP
📁	11:15	11:25	10	10	06321 2x : 06325 06336 06328	Anna Čonková	Rváčov 112, Vysočina - Rváčov, Česko	Evženie Neumannová	111
📁	11:10	11:20	10	10	06328 06321 2x : 06325 06336	Lucie Bučilová	Rváčov 112, Vysočina - Rváčov, Česko	Evženie Neumannová	111

Obrázek 6.3: Vzhled webové stránky pro přípravu vyúčtování

Po otevření stránky a inicializaci komponenty `BillingNewCreatePageComponent` pošle klientská část serverové části požadavek o seznam všech zdravotních pojišťoven uložených v databázi. Ke každé z načtených pojišťoven se následně pomocí standardní Angular třídy `FormBuilder` vytvoří formuláře pro ukládání hodnot z checkboxů, kterými bude uživatel označovat, pro které všechny pojišťovny chce vyúčtování připravit. Z globálního úložiště jsou pomocí selektoru získány všechny již přednačtené služby, které spadají do aktuálně vybrané organizace a přihlášený uživatel má právo na jejich zobrazení. Datové rozmezí je automaticky nastaveno na předchozí měsíc a typ vyúčtování na „Řádné vyúčtování“.

Všechny změny ve výběru služby, datového rozmezí, typu vyúčtování a zaškrtování zdravotních pojišťoven jsou detekovány pomocí standardní funkce `.subscribe()`. Pokud jsou všechny 4 zmíněné vstupy řádně vyplněny, zavolá se koncový bod `getBillingErrors`, o jehož fungování detailně pojednává podsekcce 6.2. V odpovědi přijdou 3 položky:

- **errors** – seznam chyb (konfliktů) mezi návštěvami (zdravotními výkony), které odpovídají uživatelským vstupům. Každá chyba se skládá z unikátního identifikátoru (pro odlišení od ostatních chyb), kódu chyby, textu chyby (na obrázku 6.2 zobrazeným pod jménem pacienta v obou chybách) a pole návštěv, které s chybou přímo souvisí.
- **unbilledWork** – boolean položka, která udává, zda zadaným parametrům kromě datového rozmezí odpovídají nějaké návštěvy, které byly provedeny před zadaným časovým rozmezím. Tato hodnota ovlivňuje zobrazení žlutého tlačítka „Nevyúčtované práce v předchozích obdobích“.
- **anyWorksIncluded** – boolean položka, která udává, jestli zadaným parametrům odpovídají vůbec nějaké návštěvy (jestli se v databázi nacházejí nějaké návštěvy spadající do vybrané služby, zadaného období a byly provedeny na pacientovi, jehož zdravotní pojišťovna je jedna z vybraných). Tato hodnota ovlivňuje, jestli je zobrazeno finální potvrzovací tlačítko „Připravit vyúčtování“, nebo je zobrazena hláška, která uživatele informuje, že pro zadané parametry není co účtovat.

Pokud je v získané odpovědi neprázdný seznam s chybami ve vyúčtování, zobrazí se tyto chyby na pravé straně vedle výběrů zdravotních pojišťoven. Každá z těchto chyb zobrazuje

jméno pacienta, u kterého byly návštěvy obsahující chyby provedeny, a text chyby. Uživatel může kliknutím na chybu zobrazit všechny návštěvy související s chybou. Na obrázku 6.2 jsou zobrazeny návštěvy související s první chybou. Tato chyba upozorňuje, že signální výkon 06321 byl u jednoho pacienta vykázán více než jednou za 7 dní. Uživatel může v tomto případě pro opravu otevřít detail jedné ze zobrazených návštěv (kliknutím na modrou ikonu složky vlevo na řádku), odstranit výkon 06321 z vykázaných a vrátit se zpět. V tuto chvíli by měla chyba zmizet.

Žluté tlačítko „Nevyúčtované práce v předchozích obdobích“ vpravo nahoře se zobrazí, pokud má v odpovědi koncového bodu `getBillingErrors()` položka `unbilledWork` hodnotu `true`. Kliknutím na toto tlačítko se aktuální stránka přepne na zobrazení seznamu nevyúčtovaných prací před zadaným obdobím. Tento seznam je implementován v komponentě, která na aktuální stránce v případě zobrazení pouze nahradí obsah a obsahuje tlačítko zpět, které přepne zobrazení aktuální stránky znovu na přípravu vyúčtování. Nijak se tedy nemění url adresa.

Pokud je seznam chyb v odpovědi prázdný a položka `anyWorksIncluded` obsahuje hodnotu `true`, tak se nezobrazí žádný seznam chyb a pod seznamem pojišťoven se zobrazí modré potvrzovací tlačítko „Připravit vyúčtování“. Po kliknutí na toto tlačítko se odešle požadavek na koncový bod `prepareBilling`, jehož fungování je detailně popsáno v druhé části podsekcce 6.2. Pokud při přípravě vyúčtování nenastala v serverové části nějaká chyba, tak odpověď obsahuje dočasný identifikátor připraveného vyúčtování. Komponenta `BillingNewCreatePageComponent` po obdržení tohoto identifikátoru uživatele přesměruje na url adresu `/create/confirm/:id`, která obsahuje stránku pro potvrzení a vytvoření vyúčtování popsanou v následující podsekcce. Na konec zmíněné url adresy se místo `:id` dosadí právě obdrženy dočasný identifikátor.

Klientská část vytvoření vyúčtování

Na stránku pro potvrzení a vytvoření vyúčtování se uživatel může dostat pouze automatickým přesměrováním z přípravy vyúčtování popsané v předchozí podsekcce. Url adresa stránky pro potvrzení a vytvoření vyúčtování je `/create/confirm/:id`, kde `:id` je dočasný identifikátor připraveného vyúčtování. Stránku tvoří komponenta `BillingNewConfirmPageComponent` a slouží pro uživatele, aby si tvořené vyúčtování prohlédl, zkontroloval, případně upravil detaily a následně vytvoření potvrdil. Podoba stránky je zachycena na obrázku 6.4.

Faktury										
Číslo faktury	Rok	Měsíc	Datum vystavení	Datum splatnosti	Datum odeslání	Fakturovaná částka	Dávek ve faktuře	Druh pojistného vztahu	ZP	Zrušit vyúčtování
2024002	2024	3	4.5.2024	4.6.2024		9 261,- Kč	2	Veřejné zdravotní pojištění	111	
Celkem:						9 261,- Kč	2			

Číslo dávky	Rok	Měsíc	Typ dávky	ZP	Druh pojistného vztahu	Charakter dávky	Počet dokladů v dávce	Číslo dokladů	Počet bodů za doklady v dávce	Cena za doklady v dávce
240007	2024	3	Dávka poukazů na V/O	111	Veřejné zdravotní pojištění	Původní	2	8 - 9	1029	2 478,- Kč
240008	2024	3	Dávka vyúčtování cest	111	Veřejné zdravotní pojištění	Původní	1	10 - 10	0	6 783,- Kč
Celkem:							3			9 261,- Kč

Vytvořit vyúčtování a stáhnout soubory

Obrázek 6.4: Vzhled webové stránky pro vytvoření vyúčtování

Po otevření stránky a inicializaci komponenty `BillingNewConfirmPageComponent` se vezme dočasný identifikátor uvedený v url adrese místo `:id` a předá se jako parametr koncovým bodům `getInvoices` a `getBillingBatches`. Ty nezávisle na sobě vrátí seznamy faktur a dávek, které pod dočasný identifikátor vyúčtování spadají. Po načtení každého z obou seznamů se pomocí standardní Angular třídy `FormBuilder` vytvoří formuláře pro ukládání upravitelných parametrů faktur, respektive dávek. Faktury mají upravitelné číslo faktury, rok, měsíc, datum vystavení, datum splatnosti a datum odeslání. Dávky mají upravitelné pouze číslo dávky. Oba seznamy se po jejich načtení zobrazí do tabulek společně s poli pro úpravu.

Každý řádek v tabulce faktur obsahuje upravitelná pole číslo faktury, rok, měsíc, datum vystavení, datum splatnosti a datum odeslání. Hodnota ve sloupci „Fakturovaná částka“ obsahuje součet cen za spadající dávky a je klikatelná. Po kliknutí na fakturovanou částku se zobrazí modální okno s rozpisem fakturované částky podle typů dávek (kolik je fakturováno za výkony a za dopravu) a podle jednotlivých pracovišť. Sloupec „Dávek ve faktuře“ obsahuje počet dávek, které pod fakturu spadají (typicky 2, jedna od každého typu). Kliknutím na toto číslo se zobrazí modální okno s tabulkou dávek, které pod danou fakturu spadají. Pro případ na obrázku 6.4, kde byla vytvořena pouze jedna faktura a 2 dávky k ní, toto modální okno postrádá smysl, protože pouze zobrazí znovu stejné dávky, ale ve vyúčtováních s více fakturami, a tedy i více dávkami, kde není na první pohled zřetelné, které dávky patří pod kterou fakturu, se pro toto zobrazení pouze dávek spadajících pod fakturu najde využití. Celou fakturu včetně jejích dávek je možné odstranit pomocí ikony červeného koše vpravo na řádku. Taková faktura nebude zahrnuta do vytvořeného vyúčtování a všechny návštěvy pod ni spadající bude možné vyúčtovat znovu v rámci jiného vyúčtování.

Řádky v tabulce dávek obsahují pouze jedno upravitelné pole, a to číslo dávky. Jinak řádky dávek zobrazují několik informací převzatých z nadřazené faktury. Hodnota ve sloupci „Počet dokladů v dávce“ je zároveň odkaz na stránku detailu dané dávky, ve kterém jsou jednotlivé doklady zahrnuté do dávky zobrazeny. Hodnoty ve sloupci „Číslo dokladů“ udávají nejnižší a nejvyšší číslo zahrnutých dokladů. Doklady jedné dávky jsou tvořeny společně, takže všechna čísla dokladů mezi uvedenými čísly také odpovídají dokladům v dané dávce. Kliknutím na čísla dokladů se otevře modální okno umožňující uživateli přečíslování všech dokladů dané dávky.

Veškeré úpravy faktur a dávek, které je v tabulkách možné provádět, se ukládají okamžitě po změně hodnoty. Výjimkou jsou čísla faktur a čísla dávek, která podléhají průběžným kontrolám, protože musí dodržovat unikátnosti – čísla faktur musí být unikátní v rámci služby a čísla dávek v rámci zdravotní pojišťovny. Tyto kontroly jsou implementovány v rámci asynchronních validátorů, které jsou políčkům nastaveny při tvoření formulářů pomocí třídy `FormBuilder`. Tyto validátory jsou volány při každé změně textu v daných polích. Kontrola unikátnosti čísla faktury se provádí pokusným zavoláním koncového bodu `getInvoices` s parametry zadaného čísla faktury a nadřazené služby. Unikátnost čísla dávky se kontroluje zavoláním koncového bodu `getBillingBatches` s parametry zadaného čísla dávky a zdravotní pojišťovny, které je přiřazena nadřazená faktura. Pokud tyto koncové body vrátí více než jeden záznam, nebo pouze jeden záznam, který ale neodpovídá právě upravovanému záznamu, tak to znamená, že nastal konflikt v unikátnosti, úprava se neuloží a uživatel je hláškou na tento konflikt upozorněn. Pokud konflikt nového čísla faktury/dávky a čísel faktur/dávek v databázi nenastal, tak je úprava teprve uložena.

Jakmile je uživatel spokojen s podobou faktur, dávek a dokladů, tak klikne na modré potvrzovací tlačítko „Vytvořit vyúčtování a stáhnout soubory“. Tím se pošle požadavek na koncový bod `createBilling`, v jehož parametrech budou identifikátory jednotlivých

faktur, které mají být ve finálním vyúčtování zahrnuty. Pokud přijde odpověď v pořádku, tak se všechna upravitelná pole v obou tabulkách nahradí prostými neupravitelnými texty a uživateli bude nabídnuto stažení archivu obsahujícího soubory KDAVKA a FDAVKA.

Serverová část přípravy vyúčtování

Serverová část přípravy vyúčtování musí obstarat odpovědi na dva hlavní požadavky – kontrola chyb ve vyúčtování a příprava vyúčtování.

Kontrola chyb vyúčtování se nachází na koncovém bodu `getBillingErrors` a v parametrech vyžaduje identifikátor služby, datum od, datum do, typ vyúčtování a seznam zdravotních pojišťoven. Na základě těchto parametrů jsou z databáze získány provedené výkony a návštěvy, které zde poskytují data o cestování pracovníků. K provedeným výkonům jsou z databáze získány návštěvy, během kterých byly provedeny. Oba seznamy návštěv jsou následně spojeny a vyfiltrovány, aby výsledný seznam obsahoval pouze unikátní návštěvy. Následně je seznam unikátních návštěv rozdělen do podseznamů, kde se návštěvy v každém takovém podseznamu týkají pouze jedné složky pacienta a žádné dva podseznamy se netýkají stejné složky pacienta.

Tyto podseznamy jsou jednotlivě navzájem nezávisle testovány na možné chyby vyúčtování. Například jednou z kontrol je ověření, že signální výkon 06349 byl zadán maximálně jednou za den. V tomto případě se seznam návštěv u jednoho pacienta rozdělí na podseznamy pro jednotlivé dny a tyto podseznamy jsou testovány, zda neobsahují více než jeden vykázaný výkon 06349. Pokud ano, tak je do výsledného seznamu chyb přidána položka s odpovídajícím kódem chyby a zprávou a do seznamu souvisejících prací jsou vloženy obě návštěvy, které výkon 06349 v daný den obsahovaly.

Struktura odpovědi, kterou koncový bod vrací, je popsána v podsekcí 6.2. Hodnoty položek `unbilledWork` a `anyWorksIncluded` jsou nastaveny podle toho, jestli jsou v databázi nalezeny nějaké nevyúčtované návštěvy, které odpovídají vstupním parametrům vloženým do koncového bodu `getBillingErrors` s tím, že pro položku `unbilledWork` je podmínka na datum nastavena na „menší než datum od“.

Druhým důležitým koncovým bodem pro přípravu vyúčtování je `prepareBilling` pro vygenerování faktur, dávek a dokladů. Vstupní parametry tohoto koncového bodu jsou stejné jako u předchozího `getBillingErrors`.

Nejprve se získá číslo příští faktury, které je vygenerováno pomocí roku, pro který je vyúčtování tvořeno, a pořadového čísla, které je o 1 vyšší než aktuálně nejvyšší číslo faktury z aktuálního roku. Protože výslednou odpovědí koncového bodu `prepareBilling` má být dočasný identifikátor vygenerované struktury vyúčtování, byla oproti návrhu přidána ještě navíc entita `UnprocessedBilling` a k ní odpovídající tabulka `unprocessed_billings`. Před generováním faktur je v databázi tato entita vytvořena a její identifikátor se následně vytvořeným fakturám uloží.

Celé generování faktur, dávek a dokladů je prováděno zvlášť pro každou zdravotní pojišťovnu vybranou uživatelem. V rámci generování pro jednu danou pojišťovnu jsou už také získány následující čísla dávky a dokladu podobným principem jako u faktur. Podle vstupních parametrů jsou z databáze vytaženy všechny provedené výkony a návštěvy (pouze pro vyúčtování cest). Oba tyto seznamy jsou rozděleny na podseznamy podle druhu pojistné smlouvy pacienta, na kterého se vážou. Následující procesy jsou prováděny zvlášť nad každým z těchto podseznamů patřících pod jeden druh pojistné smlouvy pacienta. Seznamy provedených výkonů a návštěv jednoho druhu pojistné smlouvy jsou dále rozděleny na další podseznamy podle měsíce.

Z takto rozdělených seznamů jsou vytvořeny faktury – jedna pro provedené výkony a návštěvy každého druhu pojistné smlouvy a měsíce. Většina dat faktury je odvozena z dat provedených výkonů a návštěv. Jako číslo faktury je použito dříve získané číslo následující faktury a hned po použití je toto číslo inkrementováno pro použití další fakturou. Fakturám je také uložen dočasný identifikátor entity **UnprocessedBilling**.

Jakmile je v rámci skupiny jednoho druhu pojistného vztahu a jednoho měsíce vytvořena faktura, tak se přechází na tvoření dávek. Každé faktuře jsou vytvořeny dvě dávky – jedna pro vyúčtování provedených výkonů a jedna pro vyúčtování cest. Čísla dávek jsou dávkám přiřazovány stejným principem jako u faktur – každá dávka uloží aktuální následující číslo dávky a ihned jej inkrementuje.

Po vytvoření dávek jsou tvořeny doklady. Ty jsou podle typu nadřazené dávky tvořeny buď pouze pro provedené výkony, nebo pouze pro návštěvy. Pokud jsou doklady tvořeny pro provedené výkony, tak je seznam provedených výkonů postupně rozdělen na podseznamy podle složek pacienta, podle poukazů platných ve chvíli provedení výkonu a nakonec tak, aby v jednom podseznamu nebylo více než 99 záznamů. Poté mohou být doklady pro jednotlivé podseznamy zahrnující provedené výkony vytvořeny. Daným provedeným výkonům se nakonec ještě uloží identifikátor dokladu, pod který byly zahrnuty. Pokud jsou doklady tvořeny pro návštěvy, tak je seznam návštěv rozdělen na podseznamy podle pracovníků, co návštěvy provedli a následně se také provede rozdělení vzniklých podseznamů, aby v žádném nebylo více než 99 záznamů. Pro každý výsledný podseznam návštěv je vytvořen doklad a odpovídajícím návštěvám je uložen identifikátor dokladu.

Jak bylo v předchozím odstavci pro oba případy dokladů uvedeno, výsledné podseznamy záznamů nesmějí obsahovat více než 99 položek a ještě navíc položky z jednoho dne musí být přítomny společně pouze v jednom podseznamu. Proces získání indexů, na kterých se má vstupní seznam rozdělit, aby splňoval tyto podmínky, zachycuje algoritmus 1.

Algoritmus 1: NAJDI INDEXY, NA KTERÝCH SE MÁ SEZNAM ZÁZNAMŮ ROZDĚLIT

Vstup: záznamy (*záznamy seřazené podle data*)

Výstup: indexy rozdělení

```

1   $X = \emptyset$            //indexy rozdělení
2   $d = 1970-01-01$        //datum předchozího záznamu
3   $c = 0$                  //čítač záznamů z aktuálního data
4   $i = -1$                 //index, na kterém bylo datum naposledy rozdílné

5  for  $j = 0; j < \text{počet}(\text{záznamy}); j++$  do
6       $c = c + 1$ 
7      if  $d \neq \text{datum}(\text{záznamy}[j])$  then
8           $d = \text{datum}(\text{záznamy}[j])$ 
9           $i = j$ 
10     end if
11     if  $c > 99$  then
12         if  $i > 0$  then
13             Přidej  $i$  do  $X$  //přidej index do pole indexů rozdělení
14         end if
15          $c = j - i + 1$ 
16     end if
17 end for
18 return  $X$ 

```

Po vytvoření dokladů každé dávky se ještě ověří, že opravdu byly nějaké doklady vytvořeny. Pokud ne, tak se dávka rovnou smaže a pokud jsou takto smazány obě dávky faktury, tak je smazána i samotná faktura. Po ukončení všech procesů pro všechny podseznamy výkonů a návštěv se odešle odpověď s identifikátorem dočasně vytvořené entity `UnprocessedBilling`.

Serverová část vytvoření vyúčtování a generování souborů

Serverová část pro vytvoření vyúčtování má za úkol převést uvedené faktury, dávky, doklady, související provedené výkony a návštěvy na zpracované vyúčtování a vygenerovat soubory `KDAVKA` a `FDAVKA` podle striktně definovaných pravidel.

Obě tyto funkce spadají pod jeden koncový bod – `createBilling`. Tento koncový bod přijímá seznam identifikátorů faktur, které mají být do vytvořeného vyúčtování zahrnuty a v odpovědi vrací identifikátor vytvořeného vyúčtování a url adresu odkazující na archiv vygenerovaných souborů.

Proces tvoření vyúčtování v serverové části začíná získáním všech faktur, jejichž identifikátory byly obdrženy ve vstupních parametrech, z databáze. Následně jsou ke všem těmto fakturách načteny jejich dávky a poté doklady všech těchto dávek. Na základě identifikátorů všech těchto dokladů jsou v databázi aktualizovány provedené výkony a návštěvy, které pod tyto doklady spadají, na stav vyúčtování „Vyúčtovaný“.

Každý jeden seznam dokladů načtený pro každou dávku je seřazen podle pořadového čísla dokladu v dávce (aby za sebou byly doklady týkající se stejné složky pacienta), pomocí funkce `.map()` je seznam objektů dokladů převeden na seznam identifikátorů dokladů. Všechny takto průběžně vytvořené seznamy jsou spojeny do jednoho seznamu obsahujícího identifikátory všech dokladů, ze kterých mají být vygenerovány soubory. Podobně jsou pro generování souborů předpřipraveny relevantní provedené výkony a návštěvy. Ty nejsou ukládány do seznamu, ale do dvou objektů (jeden pro výkony a jeden pro návštěvy), kde klíči jsou identifikátory dokladů a hodnotami seznamy výkonů/návštěv patřících tomuto dokladu. Seznam identifikátorů dokladů je ještě rozdělen podle cílových zdravotních pojišťoven (protože pro každou je vytvořeno vyúčtování zvlášť, včetně souborů `KDAVKA` a `FDAVKA`) a společně s objekty s provedenými výkony a návštěvami je předán funkci pro vygenerování souborů.

Generování souborů se musí řídit striktními pravidly formátování uvedenými v [3]. Soubor `KDAVKA` musí nejdříve obsahovat data o všech vykázaných výkonech a až poté musí uvádět data o všech vykonaných cestách. Soubor `FDAVKA` zase musí obsahovat ceny a počty bodů za jednotlivé zahrnuté dávky (získané sečtením hodnot spadajících dokladů) a také celkovou cenu a počet bodů za všechny dávky ve vyúčtování. Kvůli tomu je proces generování rozdělen na sběr informací a následně samotné generování obsahů souborů.

Při sběru informací se iteruje přes všechny identifikátory dokladů. Každý doklad je na základě svého identifikátoru samostatně získán z databáze. Pro získání jednoho dokladu z databáze se používá dotaz, který společně s dokladem vrátí také informace o nadřazené dávce, faktuře, zdravotní pojišťovně a službě. Všechna tato data jsou důležitá při generování souborů a jsou při iterování doklady sbírána.

Samotné generování vychází z dokumentu [3] – kapitola „4.31. Faktura - žádost o vyúčtování“ pro soubor `FDAVKA` a kapitoly „4.06. Poukaz na vyšetření/ošetření“ a „4.36. Vyúčtování cest lékaře v návštěvní službě“ pro vykázaní provedených výkonů a cest v souboru `KDAVKA`. Všechny řádky a jednotlivé atributy v nich mají pevně danou délku, na kterou jsou vkládané hodnoty nataženy pomocí funkcí `.padEnd(<délka_atributu>, ' ')` pro tex-

tové atributy a `.padStart(<délka_atributu>, '0')` pro číselné atributy. Mezi jednotlivé řádky souborů jsou vloženy znaky konce řádku `\r\n`.

K vytvoření souborů je využit balíček `FileManager`, který je implementovaný speciálně pro systém XCarol. Nejdříve se pomocí tohoto balíčku soubor v úložišti založí, po čemž se v odpovědi vrátí identifikátor souboru a url adresa, na kterou se má obsah souboru nahrát. Při nahrávání se kromě této url adresy a obsahu souboru ještě specifikuje kódování a hlavička uvádějící typ obsahu požadavku. Jakmile je obsah souboru na danou url adresu nahrán, zavolá se funkce `confirm()` balíčku `FileManager`, které se vloží dříve získaný identifikátor a v odpovědi přijde objekt nahraného souboru typu `FileObject`, ze kterého se vezme url adresa, která bude uložena k entitě daného „Vyúčtování“ v databázi.

Aby mohly být všechny soubory, které jsou vytvořené při jednom vyúčtování, staženy dohromady, je potřeba je vložit do archivu. K tomu zde slouží knihovna `AdmZip`, které se pomocí její metody `.addFile` postupně přidávají obsahy všech souborů společně s jejich jmény. Nakonec se pomocí funkce `.toBuffer()` získá řetězec s obsahem celého archivu, uloží se do souborového úložiště (stejně, jak bylo popsáno v předchozím odstavci) a získá se url adresa uloženého souboru, která je odeslána klientské části, kde si z ní uživatel archiv stáhne do lokálního úložiště.

Implementace ostatních stránek vyúčtování

Tato sekce popisuje detaily implementací zbývajících stránek, které se týkají vyúčtování a byly také vytvořeny v rámci této diplomové práce. Jedná se o tyto stránky:

- Odmítnuté doklady – slouží pro manuální změny stavů dokladů, výkonů a návštěv.
- Nevyúčtované práce – zobrazuje seznam ještě nevyúčtovaných prací.
- Vytvořená vyúčtování – poskytuje seznamy vyúčtovaných faktur a dávek. Také obsahuje odkaz na nezpracovaná vyúčtování.
- Detail faktury – zobrazuje všechny informace o faktuře, včetně spadajících dávek.
- Detail dávky – zobrazuje všechny informace o dávce, včetně spadajících dokladů.
- Detail dokladu – zobrazuje všechny informace o dokladu, včetně spadajících provedených výkonů, nebo provedených návštěv.

Implementace klientských a serverových částí všech těchto stránek jsou detailněji popsány v následujících podsekcích.

Odmítnuté doklady

Základ stránky „Odmítnuté doklady“ tvoří komponenta `BillingRefusedPage` a nachází se na url adrese `/refused`. Stránka se skládá z 5 podstránek, kde každou tvoří samostatná komponenta a jsou popsány níže. Vzhled stránky se zobrazovanou podstránkou „Vyhledání dokladu“ je vyobrazen na obrázku 6.5.

Služba
Zdravotní služba Charity Prostějov

Vyhledání dokladu Odmítnuté návštěvy Odmítnuté výkony Stornované návštěvy Stornované výkony

Zdravotní pojišťovna
111 - Všeobecná zdravotní pojišťovna České republiky

Typ dávky
Dávka poukazů na V/O

Číslo dávky
240003 (2024/3)

Číslo dokladu

Číslo dokladu	Datum od	Datum do	ZP	Druh pojistného vztahu	Klient	Číslo pojistěnce	Poukaz	Stav
2	28.4.2024	29.4.2024	111	Veřejné zdravotní pojištění	Silvie Ježková	7482761	Poukaz DP č.2, platnost do 29.1.2027	Vyúčtováný

Odmítnout celý doklad

<input type="checkbox"/>	Datum	Kód	Zkrácený název	Návštěva	Odbornost	Počet	Pracovník	Stav
<input type="checkbox"/>	28.4.2024	80051	PRVNÍ VYŠETŘENÍ A ZAV...	➤	720	1	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	28.4.2024	80053	KOMPLEXNÍ VYŠETŘENÍ ...	➤	720	2	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	28.4.2024	80055	KONTROLNÍ VYŠETŘENÍ ...	➤	720	1	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	28.4.2024	91935	(DRG) ČASNÁ INDIKACE ...	➤	720	1	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	29.4.2024	91935	(DRG) ČASNÁ INDIKACE ...	➤	720	1	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	29.4.2024	80051	PRVNÍ VYŠETŘENÍ A ZAV...	➤	720	1	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	29.4.2024	80055	KONTROLNÍ VYŠETŘENÍ ...	➤	720	1	Lucie Bučilová	Vyúčtováný
<input type="checkbox"/>	29.4.2024	80053	KOMPLEXNÍ VYŠETŘENÍ ...	➤	720	2	Lucie Bučilová	Vyúčtováný

Odmítnout vybrané výkony

Obrázek 6.5: Vzhled webové stránky pro odmítnuté doklady

Na obrázku 6.5 je vyhledávání nastaveno na typ dávky „Dávka poukazů na V/O“, a proto výsledky tvoří tabulka dokladů a tabulka provedených výkonů. V případě vybrání typu dávky „Dávky vyúčtování cest“ by byla tabulka provedených výkonů nahrazena tabulkou návštěv.

Výběr služby nad výběrem podstránek je součástí základní komponenty tvořící stránku a ta vybranou službu předává vstupními parametry všem 5 podstránkám. Možnosti výběru služeb jsou stejně jako u přípravy vyúčtování získány selektorem z globálního úložiště, kde jsou služby již přednačtené.

Podstránka „Vyhledání dokladu“ uživateli umožňuje vyhledat doklady a jejich výkony, nebo návštěvy pomocí čtyř parametrů. Zdravotní pojišťovny jsou načteny automaticky při inicializaci komponenty a možnosti typů dávky jsou dány napevno. Jakmile komponenta detekuje, že uživatel vybral zdravotní pojišťovnu, typ dávky i službu v nadřazené komponentě, tak s těmito parametry zavolá koncový bod `getBillingBatches` pro získání odpovídajících dávek. Poté, co jsou dávky obdrženy, tak z nich komponenta vygeneruje možnosti pro výběr dávky. V tuto chvíli stále nejsou načteny žádné doklady ani výkony, či návštěvy, protože by jich bylo příliš mnoho. Až jakmile uživatel výběrem specifikuje také číslo dávky, tak jsou načteny doklady a pod ně spadající provedené výkony, nebo návštěvy. Po načtení těchto dvou seznamů se zobrazí obě tabulky a také se vygenerují možnosti pro poslední uživatelský vstup – „Číslo dokladu“. V případě jeho vyplnění zbude v zobrazených tabulkách pouze řádek s vybraným dokladem a řádky s výkony, či návštěvami spadajícími pod tento vybraný dokument.

Uživatel může vybrat libovolný počet záznamů v obou tabulkách a nastavit je na odmítnuté. Při změně stavu vyúčtování provedených výkonů a návštěv se v serverové části kromě aktualizace stavu u odpovídajících záznamů ještě ověřuje stav příslušného dokladu. Doklad samotný totiž nemusí být uživatelem explicitně odmítnut, ale pokud jsou odmítnuty všechny výkony, nebo návštěvy, které pod něj spadají, tak se musí i doklad nastavit

na odmítnutý. Mimo základní stavy „Vyúčtovaný“, „Odmítnutý“ a „Stornovaný“ může doklad také nabývat stavů „Částečně odmítnutý“, „Částečně stornovaný“ nebo „Částečně odmítnutý i stornovaný“. Při explicitním odmítnutí dokladu jsou automaticky odmítnuty i všechny jeho výkony, nebo návštěvy.

Podstránky „Odmítnuté návštěvy“ a „Stornované návštěvy“ tvoří jedna komponenta, protože obě obsahují téměř stejnou tabulku návštěv. Jako vstupní parametr této komponenty je informace o tom, kterou z těchto tabulek má zobrazovat. Podle toho se pak liší parametry načítání návštěv, názvy tlačítek na změny stavů a zobrazení sloupce „Důvod stornování“, který se zobrazuje pouze v tabulce stornovaných návštěv.

Podstránky „Odmítnuté výkony“ a „Stornované výkony“ jsou také tvořeny jednou komponentou, pro kterou platí stejné vlastnosti, jako pro předchozí komponentu.

Nevyúčtované práce

Stránku „Nevyúčtované práce“ tvoří komponenta `BillingUnbilledWorkPage` a nachází se na url adrese `/unbilled`. Součástí této komponenty je výběr služby a výběr datového rozmezí. Možnosti do výběru služby jsou získávány z globálního úložiště pomocí selektoru a datové rozmezí je stejné, jako na stránce přípravy vyúčtování. Tyto dva uživatelské vstupy tvoří parametry pro načítání návštěv. Dalším implicitním parametrem je, že dané návštěvy mají prázdný stav vyúčtování.

Samotné načítání a zobrazení návštěv však není součástí hlavní komponenty `BillingUnbilledWorkPage`, ale podkomponenty `BillingUnbilledWork`, která je také použita na stránce přípravy vyúčtování pro zobrazení nevyúčtovaných prací v předchozích obdobích. Tato komponenta vstupními parametry přijímá službu a datové rozmezí, pro které má nevyúčtované práce načíst. Načtené práce zobrazuje v tabulce, umožňuje navigaci na jejich detaily a také jejich odstranění.

Vytvořená vyúčtování

Navigaci na stránku „Vytvořená vyúčtování“ předchází výběr zdravotní pojišťovny v levém menu, pro niž se vytvořená vyúčtování mají na této stránce zobrazit. Stránku tvoří komponenta `BillingCreatedBillingsPage` a nachází se na url adrese `/created/:kód_pojišťovny`, kde `:kód_pojišťovny` je nahrazen kódem vybrané pojišťovny.

Rok/měsíc	Číslo faktury	Druh pojistného vztahu	Datum vystavení	Datum splatnosti	Počet dávek	Cena za dávku	Fakturovaná částka	KDAVKA	FDAVKA	Nastavit jako Nezpracované vyúčtování (smazat soubory KDAVKA a FDAVKA)
2024/02, 2024/03	240001, 240002	Veřejné zdravotní pojištění	10.4.2024	10.5.2024	4	7972,00 Kč	7972,00 Kč	↓	↓	⊗
2024/04	240003	Veřejné zdravotní pojištění	3.5.2024	3.6.2024	2	5246,00 Kč	5246,00 Kč	↓	↓	⊗

Obrázek 6.6: Vzhled webové stránky pro vytvořená vyúčtování

Na obrázku 6.6 je ukázka vzhledu stránky. Stránka „Vytvořená vyúčtování“ jako většina stránek patřící k vyúčtování obsahuje filtrování pomocí výběru služby. Stránka obsahuje

3 podstránky, jejichž obsah je ale na rozdíl od předchozích komponent definován hlavní komponentou `BillingCreatedBillingsPage`. Komponenta při jakékoliv změně kódu pojišťovny v url adrese nebo výběru služby načítá odpovídající dávky obou typů zvlášť a zobrazuje je v základních tabulkách bez speciálních funkcí. Podstránka „Faktury“ je speciální v tom, že obsahuje faktury sloučené podle vyúčtování. Každý řádek této podstránky tak představuje jedno vytvořené vyúčtování a obsahuje data ze všech faktur daného vyúčtování současně. Na obrázku 6.6 je toto naznačeno v prvním řádku tabulky, který současně zobrazuje 2 faktury – textová data obou faktur jsou uvedena za sebou, číselná data jsou sečtena do jednoho čísla.

Uživatel zde může znovu stáhnout soubory KDAVKA a FDAVKA do lokálního úložiště. Url adresy pro stažení obou souborů jsou načteny zároveň s vytvořenými vyúčtováními, takže klientská část pouze spustí stahování po kliknutí na ikonu stažení. Dále může uživatel nastavit vyúčtování jako nezpracované. Klientská část v takovém případě zavolá koncový bod `deleteBilling` s identifikátorem daného vyúčtování a serverová část všem zahrnutým fakturám vymaže cizí klíč odkazující na dané „Vyúčtování“ a danou entitu „Vyúčtování“ spolu s připojenými soubory KDAVKA a FDAVKA smaže. Všem zahrnutým výkonům a návštěvám se nastaví stav vyúčtování na stornovaný.

Vpravo nahoře na stránce se nachází tlačítko odkazující na stránku nezpracovaných vyúčtování. Tu tvoří komponenta `BillingUnprocessedBillingsPage` a načítá všechny faktury a jejich dávky, kterým bylo odstraněno nadřazené „Vyúčtování“, jako bylo popsáno v předchozím odstavci. Na této stránce může uživatel zobrazené faktury a dávky zahrnout do nového vyúčtování. Pokud se tak rozhodne, tak se spustí stejný proces, jako při vytvoření vyúčtování, včetně generování nových souborů KDAVKA a FDAVKA popsán v podsekcí [Serverová část vytvoření vyúčtování a generování souborů](#).

Detail faktury, detail dávky a detail dokladu

Stránky pro detaily faktury, dávky a dokladu zobrazují všechny informace o dané entitě. Detail faktury obsahuje mimo základních údajů seznam dávek, detail dávky obdobně obsahuje seznam dokladů a detail dokladu obsahuje buď seznam provedených výkonů, nebo návštěv. Na všech těchto stránkách je možné upravovat některé základní údaje daných entit, ale pouze pokud není faktura/dávka/doklad součástí zpracovaného vyúčtování, protože v takovém případě jsou ze všech základních údajů již sestaveny soubory KDAVKA a FDAVKA a po úpravě entity by obsah souborů neodpovídal skutečným datům. Jestliže ale není faktura, dávka či doklad součástí zpracovaného vyúčtování, je možné některé jejich základní údaje instantně upravovat – změny se ukládají okamžitě po úpravě vstupních polí.

6.3 Testování

Pro účely testování byla databáze naplněna umělými daty. K naplnění databáze slouží balíček `fixtures`, který při jednorázovém spuštění postupně naplní databázi uživateli, jejich právy, organizacemi, službami atd. Organizace, typy služeb a služby mají pevně nastavené názvy, odbornosti a vzájemné propojení (která služba patří pod kterou organizaci a který typ služby). Zbytek dat je generován pomocí JavaScriptové knihovny `faker.js`, která na základě zadaných parametrů generuje náhodná ale smysluplná jména, čísla, názvy apod. Pro testování funkcionality vykazování byly upraveny zdrojové kódy pro naplnění databáze návštěvami tak, aby se v provedených výkonech vyskytla alespoň jednou každá z chyb vyúčtování, které se kontrolují při přípravě vyúčtování. Dále byla databáze naplněna sku-

tečnými zdravotními výkony převzatými z číselníku od VZP. Jednotlivé zdravotní výkony tak obsahují čas, počet bodů, cenu a další informace, díky kterým je možné vyúčtování vytvořit.

Pro ověření funkčnosti a uživatelské přívětivosti aplikace bylo provedeno testování několika klíčových scénářů zahrnujících přípravu vyúčtování, včetně ověření detekce cíleně zavedených chyb z předchozího odstavce, a dále vytvoření vyúčtování, včetně vygenerování a stažení souborů KDAVKA a FDAVKA. Také byla ověřena funkčnost odmítání a stornování dokladů, nastavení vyúčtování na nezpracované a jeho znovuzpracování.

Do vyvíjeného systému má také přístup jeden z cílových uživatelů, který se podílel na specifikaci požadavků. Všechny procesy zmíněné v předchozím odstavci také otestoval a potvrdil jejich základní funkčnost. Z jeho testování však vzešlo několik požadavků na různé úpravy a opravy, které ale už nebylo možné do této práce zahrnout. Například byla odhalena chyba v přípravě vyúčtování, kdy systém v některých chvílích nesprávně zobrazí tlačítko pro přípravu vyúčtování, přestože se v databázi nenachází žádné záznamy, které by bylo možné do vyúčtování zahrnout, a následně po kliknutí na toto tlačítko se objeví chyba vzešlá ze serverové části. Příkladem menších méně podstatných úprav může být, že by si systém měl pamatovat vybrané hodnoty, pomocí kterých se na stránce „Odmítnuté doklady“ vyhledávají doklady, provedené výkony a návštěvy.

Správnost formátu generovaných souborů KDAVKA a FDAVKA byla otestována programem VZPKontrol, který pro tyto účely poskytuje VZP. Program je volně ke stažení pouze pro smluvní partnery VZP a k využití v této práci byl získán díky kontaktům na jednoho z takových partnerů. Program VZPKontrol umožňuje prohlížet a kontrolovat soubory KDAVKA a FDAVKA, zda jejich obsahy splňují pravidla formátování. Po spuštění kontroly program zavolá aplikační rozhraní VZP a ihned přijde odpověď ve formátu pdf. Obsah vráceného pdf souboru zobrazuje obrázek 6.7.

Z obrázku 6.7 lze vyčíst, že kontrolovaný soubor by byl zdravotní pojišťovnou odmítnut. Dále lze z výpisu zjistit, že soubor obsahuje 2 žádosti o vyúčtování (faktury) a žádné žádosti o předběžnou úhradu. Zbytek souboru tvoří seznam všech chyb, které byly v kontrolovaném souboru nalezeny.

PARAMETRY:

STAV SOUBORU: ODMÍTNUT

PŘIDĚLENÉ UNIKÁTNÍ JMÉNO: <nezadano>

IDENTIFIKACE SOUBORU

VELIKOST SOUBORU: 609 (bytes)

IČZ: 80892000

SOUBOR ZE DNE: 14.05.2024 12:02

POČET ŽÁDOSTÍ: 2

ŽÁDOSTI O VYÚČTOVÁNÍ:

OBDOBÍ	ČÍSLO FAKTURY	DRUH POJIŠTĚNÍ	TYP FAKTURY	ČÁSTKA
03/2024	0002024001	1		8 822,80 Kč
04/2024	0002024002	1		8 822,80 Kč

ŽÁDOSTI O PŘEDBĚŽNOU ÚHRADU:

SEZNAM CHYB:

ČÍSLO VĚTY/ TYP VĚTY	CHYBA V: VĚTA (V) POLOŽKA (P)	ČÍSLO CHYBY	ČÍSLO POLOŽKY	NÁZEV POLOŽKY	ZAČÁTEK POLOŽKY V ASCII
POPIS CHYBY / OPIS DAT					
1/F	P	10101	4	FICO	13
Nevyplněná povinná znaková položka/" "					
1/F	P	10101	10	FBNKOD	34
Nevyplněná povinná znaková položka/" "					
1/F	P	10101	13	FBNUCET	54
Nevyplněná povinná znaková položka/" "					
3/D	P	15029	5	DUHR	13
Povolené hodnoty jsou pouze: 1,2,3,4/"P"					
4/D	P	15029	5	DUHR	13
Povolené hodnoty jsou pouze: 1,2,3,4/"P"					
6/D	P	15029	5	DUHR	13
Povolené hodnoty jsou pouze: 1,2,3,4/"P"					
7/D	P	15029	5	DUHR	13
Povolené hodnoty jsou pouze: 1,2,3,4/"P"					

Obrázek 6.7: Výsledek kontroly souboru FDAVKA programem VZPKontrol.

Kapitola 7

Závěr

Cílem této práce bylo analyzovat, navrhnout a implementovat vykazování zdravotní péče zdravotním pojišťovnám, které mělo být součástí webového informačního systému XCarol. Všechny 3 kroky byly úspěšně splněny a výsledkem práce jsou funkční procesy pro vytvoření vyúčtování a jeho následné spravování.

Na začátku práce proběhlo seznámení s moderními přístupy ve vývoji webových informačních systémů a seznámení s problematikou vykazování zdravotní péče v České republice, včetně rozboru dokumentů od Veřejné zdravotní pojišťovny České republiky, které popisují formáty výsledných generovaných souborů. Následně proběhly analýzy základní kostry systému XCarol a funkcionality vykazování, která měla být na základní kostře systému vystavěna. Podle analýzy byl vytvořen návrh funkcionality vykazování zdravotní péče a podle tohoto návrhu byla tato funkcionalita implementována.

Výsledkem práce je celkem 9 hlavních webových stránek umožňujících vytvoření vyúčtování, vygenerování souborů KDAVKA a FDAVKA pro odeslání vyúčtování zdravotním pojišťovnám, odmítání, stornování a znovu vyúčtování dokladů, provedených zdravotních výkonů a cest za pacienty, zobrazení přehledů vytvořených vyúčtování a jejich složek, rušení vytvořených vyúčtování a jejich opětovné vytvoření.

Ve vytvořené funkcionalitě lze nalézt spoustu možností k vylepšení. Nejpřínosnějším budoucím vylepšením by bylo automatické odesílání vytvořených vyúčtování na aplikační rozhraní zdravotních pojišťoven a následné zprostředkování výsledků vyúčtování uživatelům systému. Dalším z budoucích vylepšení může například být, že u detekovaných chyb ve tvořeném vyúčtování by mohlo být možné upravovat související návštěvy přímo na stránce přípravy vyúčtování, aniž by bylo potřeba přecházet na detail upravované návštěvy. Na stejné stránce by také mohla zůstat vybraná stejná chyba, pokud se v nově načtených chybách stále nachází – momentálně se při novém načtení chyb aktuálně vybraná chyba odznačí. Dále by se na podstránce „Vyhledání dokladu“ na stránce odmítnutých dokladů mohly zapamatovávat výběry pro vyhledávání, pokud uživatel přejde na jinou podstránku a zpět.

Literatura

- [1] *Obecné zásady komunikace přes B2B* [online]. Všeobecná zdravotní pojišťovna ČR, červen 2013 [cit. 2024-04-06]. Dostupné z: <https://media.vzpstatic.cz/media/Default/dokumenty/ekk/b2b-manualy/komunikace-b2b.pdf>.
- [2] *Pravidla pro vyhodnocování dokladů ve VZP ČR* [online]. Ústředí Všeobecné zdravotní pojišťovny ČR, leden 2018 [cit. 2024-01-03]. Dostupné z: https://media.vzpstatic.cz/media/Default/dokumenty/vyuctovani/pravidla_v62_12.pdf.
- [3] *Datové rozhraní - individuální doklady* [online]. Ústředí Všeobecné zdravotní pojišťovny ČR, září 2023 [cit. 2023-12-18]. Dostupné z: <https://media.vzpstatic.cz/media/Default/dokumenty/vyuctovani/datove-rozhrani-individualnich-dokladu-k-1.10.2023.pdf>.
- [4] *Datové rozhraní - číselníky* [online]. Ústředí Všeobecné zdravotní pojišťovny ČR, září 2023 [cit. 2023-12-18]. Dostupné z: <https://media.vzpstatic.cz/media/Default/dokumenty/vyuctovani/datove-rozhrani-ciselniku-k-1.10.2023.pdf>.
- [5] OMÁČKOVÁ, R. *Vykazování zdravotní péče* [online]. Ústav zdravotnických informací a statistiky ČR, 2016 [cit. 2023-12-18]. Dostupné z: <https://drg.uzis.cz/res/file/skoleni-koderi/20161013-prezentace.pdf>.
- [6] ZWASS, V. *Information system* [online]. Encyclopedia Britannica [cit. 2023-12-09]. Dostupné z: <https://www.britannica.com/topic/information-system>.
- [7] ČESKO. *§ 13 zákona č. 48/1997 Sb., o veřejném zdravotním pojištění a o změně a doplnění některých souvisejících zákonů - znění od 1. 7. 2023*. 1997. In: *Zákony pro lidi.cz* [online]. © AION CS 2010–2023 [cit. 10. 12. 2023]. Dostupné z: <https://www.zakonyprolidi.cz/cs/1997-48#p13>.