Masarykova univerzita
Fakulta informatiky

Rozšíření multiplatformní mobilní aplikace Kapesní skautIS o agendu správy majetku

Bakalářská práce

Michaela Ostruszková

Brno, jaro 2020

Masarykova univerzita
Fakulta informatiky

Rozšíření multiplatformní mobilní aplikace Kapesní skautIS o agendu správy majetku

Bakalářská práce

Michaela Ostruszková

Brno, jaro 2020

Na tomto místě se v tištěné práci nachází oficiální podepsané zadání práce a prohlášení autora školního díla.

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracovala samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používala nebo z nich čerpala, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Michaela Ostruszková

Vedoucí práce: RNDr. Vít Rusňák, Ph.D.

Poděkování

Na tomto místě bych ráda poděkovala RNDr. Vítu Rusňákovi, Ph.D. za odborné vedení a cenné rady týkající se této bakalářské práce. Dále bych chtěla poděkovat mému snoubenci za jeho obrovskou podporu, pomoc a trpělivost v průběhu celého studia.

Shrnutí

Junák – český skaut, z.s. vyvíjí dnes již poměrně rozsáhlý interní informační systém skautIS. Část jeho funkcí je postupně implementována v rámci mobilní aplikace *Kapesní skautIS* pro snazší správu vybraných agend v mobilním zařízení. Tato práce se zabývá rozšířením mobilní aplikace o agendu správy majetku. Nejprve byla provedena analýza současné funkcionality a rozhraní agendy pro správu majetku ve skautISu. Na základě této analýzy pak byl proveden návrh a implementace rozšíření aplikace *Kapesní skautIS* o nové funkce, které umožňují nejen správu majetku jednotek, ale i jeho rezervaci či inventarizaci. V rámci neformálního uživatelského testování byla získána zpětná vazba a podněty pro další rozšíření.

Klíčová slova

skaut, skautIS, multiplatformní mobilní aplikace, Android, iOS, informační systém, inventarizace, NativeScript, Angular

Obsah

1 Úvod **1**

2 Multiplatformní mobilní aplikace **3**
2.1 *Vývoj mobilních aplikací* 3
2.2 *NativeScript* 4
2.2.1 NativeScript a TypeScript 5
2.2.2 NativeScript a Vue.js 6
2.2.3 NativeScript a Angular 7

3 Informační systém skautIS a analýza agendy majetku **9**
3.1 *Informační systém skautIS* 9
3.2 *Webové služby* 11
3.3 *Testovací rozhraní* 12
3.4 *Poskytované agendy* 12
3.4.1 Analýza agendy majetku 14

4 Kapesní skautIS a existující aplikace pro správu majetku **21**
4.1 *Kapesní skautIS* 21
4.1.1 Struktura aplikace 21
4.1.2 Vzhled aplikace 22
4.1.3 Implementované části a agendy 23
4.2 *Mobilní aplikace pro správu majetku* 25

5 Návrh agendy majetku **26**
5.1 *Systémové požadavky* 26
5.1.1 Funkční požadavky 26
5.1.2 Nefunkční požadavky 27
5.2 *Návrh sekce majetku* 27
5.3 *Návrh sekce skladů* 29
5.4 *Návrh sekce inventarizace* 29
5.4.1 Návrh databáze 31
5.5 *Návrh sekce rezervací majetku* 32

6 Implementace agendy majetku **34**
6.1 *Obecné principy* 34
6.2 *Majetek* 35

6.3 *Sklady* 36
6.4 *Inventarizace* 37
6.4.1 Inventarizace pomocí skenování kódů 38
6.4.2 Offline inventarizace s využitím databáze SQLite 39
6.5 *Rezervované, vydané a půjčené položky* 41
6.6 *Provádění rezervace* 43

7 Testování a nasazení aplikace **44**

8 Závěr **46**

Bibliografie **47**

A Elektronické přílohy v IS MU **49**

B Návrhy obrazovek agendy majetku **50**

1 Úvod

SkautIS[1] je rozsáhlý informační systém vyvíjený pro skautskou organizaci Junák – český skaut, z. s. Poskytuje funkcionalitu pro evidenci a správu jednotek, registraci členů, správu čísel ve skautské telefonní síti (STS) a v neposlední řadě i evidenci majetku a jeho inventarizaci.

Systém je vyvíjen na zakázku firmou Skeleton Software od roku 2009. Přístup k datům poskytuje skrze webové služby přes aplikační rozhraní (*Application Programming Interface*, dále API). Díky funkcím, které API poskytuje, je možné propojit data z informačního systému s vlastní aplikací či webovou stránkou. Pro přihlášení nebo registraci uživatele je využíván jednotný přihlašovací systém (tzv. *single sign-on*), díky kterému stačí pouze jedny přihlašovací údaje pro přihlášení do ostatních aplikací využívajících skautská data. Systém je díky tomu snadno rozšiřitelný.

V minulých letech došlo k několika pokusům poskytnout vybrané služby skautISu v mobilní aplikaci [10, 11, 2]. Tyto aplikace však byly využívány jen krátce a kvůli absenci průběžných aktualizací a opravě chyb došlo k jejich zastarání. V současné době je vyvíjena multiplatformní mobilní aplikace *Kapesní skautIS*[2]. Ta poskytuje základní funkcionalitu, která zahrnuje informace o jednotce, v níž je přihlášený uživatel registrován, vyhledávání osob či správu čísel ve skautské telefonní síti[3]. Aplikace byla letos nasazena do Google Play[4] a App Store[5] ve verzi 1.0.0. Je i nadále vyvíjena a rozšiřována o nové agendy.

Tato práce analyzuje agendu majetku v systému skautIS a identifikuje funkcionalitu, kterou je smysluplné implementovat do mobilní aplikace. Praktickým výstupem pak je rozšíření stávající aplikace o agendu správy majetku a skladů, možnosti rezervace majetku a přidání procesu inventarizace. Práce navazuje na již existující bakalářskou práci Bc. Jana Crhy [2], který vytvořil multiplatformní mobilní apli-

1. `https://is.skaut.cz/`
2. Aplikace vznikla jako diplomová práce Mgr. Petra Domkaře, viz [4].
3. Skautská telefonní síť (STS) je obdoba podnikových tarifů pro členy Junáka.
4. Aplikace Kapesní skautIS pro zařízení s OS Android: `https://play.google.com/store/apps/details?id=cz.skaut.kapesniskautis&hl=cs`.
5. Aplikace Kapesní skautIS pro zařízení s iOS: `https://apps.apple.com/us/app/kapesn%C3%AD-skautis/id1486102589`.

kaci pro správu majetku a skladů. Jelikož od doby vydání aplikace došlo k zásadním úpravám poskytovaného API, aplikace již není zcela funkční. Navíc je vzhledem k právě vyvíjenému *Kapesnímu skautISu* vhodné, aby se funkcionalita jednotlivých agend koncentrovala pouze v jedné aplikaci. Po prvotní analýze se ukázalo, že bude snazší agendu implementovat znovu jako rozšíření, než se snažit o integraci původního kódu. Aplikace Bc. Jana Crhy navíc neposkytuje tolik funkcionality, jak se pro agendu majetku v *Kapesním skautISu* očekává (např. možnost úpravy položky či správu provedených rezervací).

Ve druhé kapitole jsou popsány dostupné možnosti pro vývoj mobilní aplikace a podrobně je představeno programování v aplikačním rámci NativeScript. SkautIS a agenda majetku jsou popsány v kapitole 3. Následující, čtvrtá, kapitola představuje strukturu aplikace *Kapesní skautIS* a analýzu existující aplikace pro správu majetku. Kapitola 5 se zabývá výběrem vhodných funkcí a návrhem rozšíření aplikace *Kapesní skautIS* o tyto funkce. Popis implementace tohoto rozšíření a jeho testování jsou rozebrány v kapitolách 6 a 7. Závěrečná kapitola práci shrnuje a nastiňuje další možnosti rozšíření agendy v budoucnu.

2 Multiplatformní mobilní aplikace

Tato kapitola popisuje způsoby, kterými lze vyvíjet multiplatformní mobilní aplikace. Podrobněji je představen aplikační rámec pro tvorbu multiplatformních mobilních aplikací NativeScript a jeho propojení s jazykem TypeScript a aplikačními rámci Vue.js a Angular.

2.1 Vývoj mobilních aplikací

V dnešní době na trhu dominují dva mobilní operační systémy. Jsou jimi Android od společnosti Google a iOS od společnosti Apple. Existuje řada jazyků, technologií či aplikačních rámců, které je možné pro vývoj mobilních aplikací použít. Pro vývoj pro Android je možné použít hlavně programovací jazyky Java nebo Kotlin[1], pro iOS jsou to například Objective-C[2] nebo Swift[3]. V těchto jazycích vznikají takzvané nativní aplikace, což jsou aplikace vyvíjené pro jednu konkrétní platformu s použitím nástrojů a knihoven týkajících se dané platformy. Chceme-li poskytnout aplikaci pro obě zmíněné platformy, stává se jejich tvorba složitější – vyvíjí se dvě aplikace pro obě platformy současně, obsahem jsou však shodné. Je potřeba udržet stejnou strukturu a fungování aplikace s co nejmenšími rozdíly. Kromě nativního vývoje však existují další možnosti, jak psát mobilní aplikace. Zejména se jedná o tzv. hybridní aplikace, progresivní webové aplikace (tzv. *Progressive Web Applications* (PWA)), či multiplatformní aplikace [8].

Hybridní aplikace kombinuje využití nativních a webových technologií. Vývoj probíhá pro obě platformy najednou, a to za použití HTML, JavaScriptu či jeho nástaveb, případně s využitím pomocných rámců. Využívá vykreslovací jádro (tzv. *engine*) webového prohlížeče na vykreslení komponent. Při překladu dochází k přeložení do nativní formy obou platforem. Toho je dosaženo tak, že komponenty se zaobalí do tzv. *WebView*, což je jedna ze standardních komponent mobilních aplikací, která slouží pro zobrazování webových stránek. Hlavní nevýhodou je nižší výkon takových aplikací ve srovnání s apli-

1. `https://kotlinlang.org/`
2. `https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html`
3. `https://swift.org/`

kacemi nativními. Příkladem hybridního rámce je například Apache Cordova[4] od společnosti Apache.

Progresivní webové aplikace, stejně jako hybridní, jsou kombinací funkcí nativního vývoje a přístupnosti webových stránek. PWA jsou určeny pro spuštění ve webovém prohlížeči. Využívají technologie HTML, JavaScript a CSS. Na rozdíl od hybridních aplikací není nutné PWA instalovat. Jedná se o převedení a přizpůsobení webových stránek do mobilních zařízení. Proto k jejich spuštění na jakémkoli zařízení stačí URL adresa stránky a webový prohlížeč. Stránky si uchovávají část dat přímo v zařízení, proto je možné část aplikace používat i bez připojení k internetu. Díky možnostem značkovacího jazyka HTML5 aplikace umožňuje přístup například k fotoaparátu, úložišti zařízení nebo čtečce otisku prstů. Je možné ji přidat na domovskou stránku a také dokáže v zařízení vytvářet notifikace.

Další možností je multiplatformní aplikace (z angl. *cross-platform application*). Při vývoji se opět využívá jen jeden programovací jazyk, zdrojový kód je následně zcela přeložen do nativního kódu dané platformy. Tento způsob vývoje má proto blíže k nativní aplikaci, jejich výkon je srovnatelný. Pro multiplatformní vývoj se nejčastěji volí technologie Xamarin[5], Flutter[6] nebo frameworky React Native[7] či NativeScript[8]. V dalším textu se zaměříme právě na poslední zmiňovaný, neboť v něm je naprogramována i mobilní aplikace Kapesní skautIS.

2.2 NativeScript

NativeScript je open-source framework umožňující vývoj multiplatformních mobilních aplikací. Představila jej firma Telerik v roce 2014, poprvé byl vydán v březnu 2015. NativeScript sestává z několika hlavních částí: *Runtimes, Core Modules, CLI* a *Plugins* [3]. *Runtimes* umožňuje volání API pro zisk nativního aplikačního rámce od Androidu a iOS za pomocí JavaScriptového kódu. K tomu je využito virtuálního JavaScript stroje (JavaScript Virtual Machine). *Core Mo-*

4. `https://cordova.apache.org/`
5. `https://dotnet.microsoft.com/apps/xamarin`
6. `https://flutter.dev/`
7. `https://reactnative.dev/`
8. `https://www.nativescript.org/`

dules poskytují abstrakci potřebnou ke zpřístupnění základních nativních prvků platformy. Navíc poskytují způsob zápisu grafického rozhraní, propojení dat s uživatelským rozhraním (tzv. *data-binding*)[9] a navigace v kódu vycházejícího ze značkovacího jazyka XML. Další částí je rozhraní příkazového řádku umožňující vytvářet a spouštět aplikace využívající NativeScript. Příkaz sestává z několika částí: `tns <Command> [Command Parameters] [--command <Options>]`. Mezi základní příkazy patří `tns create`, který vytvoří novou aplikaci. Dále jsou to například příkazy `tns build <Platform>` pro sestavení aplikace do vybrané platformy nebo `tns debug <Platform>` pro spuštění v režimu debug. Tvorbu aplikace ulehčují rozšiřující moduly, tzv. *Plugins*. Jsou to stavební bloky (nebo také zásuvné moduly) poskytující rozšířenou funkcionalitu. Příkladem jsou již zmíněné Core Modules. Zásuvný modul je nutné do aplikace nainstalovat příkazem `tns plugin add <Plugin>` přes NativeScript CLI. Všechny části zachycuje diagram na obrázku 2.1.

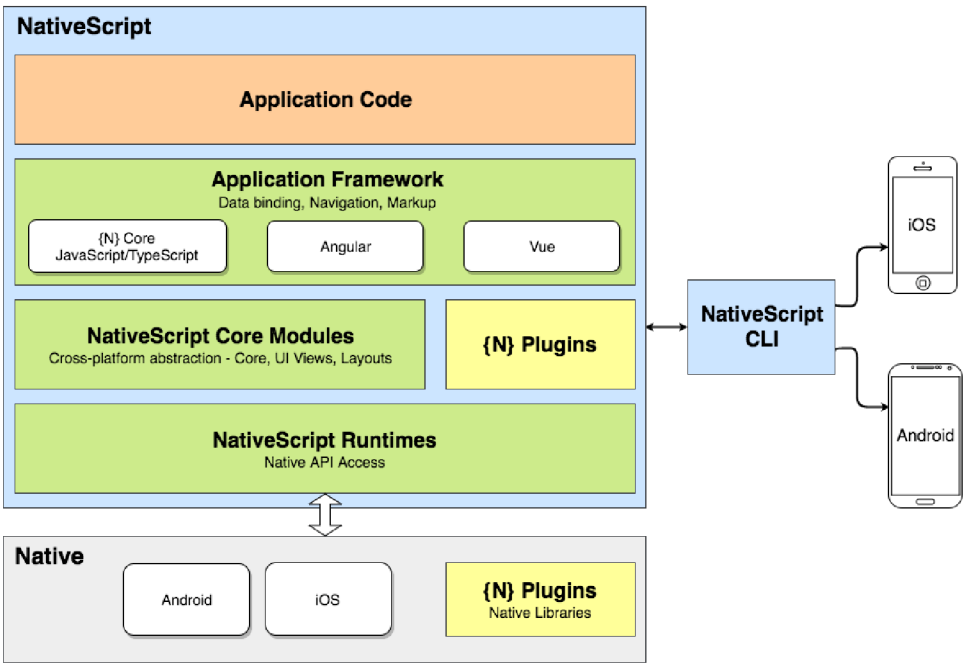
Pro tvorbu aplikace v NativeScriptu je využíván jazyk JavaScript. Pro snadnější vývoj je však možné použít také TypeScript, Angular či Vue.js.

2.2.1 NativeScript a TypeScript

JavaScript (JS) je sice jazyk, který umožňuje objektově orientované programování (dále jen OOP), není však tzv. *class-based*. To znamená, že a priori nepracuje s konceptem tříd. Klíčové slovo `Class`, které bylo zavedeno v roce 2015, představuje pouze syntaktickou zkratku pro vytvoření objektu z prototypu `Object` tak, jak ho známe například z jazyka Java či C++. Všechny třídy jsou tedy pouze rozšířením prototypu `Object`. Ten obsahuje atribut `__proto__`, ve kterém se uchovávají všechny atributy a metody prototypu včetně konstruktoru a dalších nezbytných funkcí pro správný chod objektu. Po vytvoření nové třídy se k atributu `__proto__` přidají i atributy a metody patřící této nové třídě.

Aby bylo programování více přívětivější vývojářům, kteří jsou na koncept tříd zvyklí, podporuje NativeScript možnost psát kód i v ja-

9. Data-binding je proces, který stanovuje spojení mezi uživatelským rozhraním a daty, které zobrazuje.



Obrázek 2.1: Diagram popisující princip fungování aplikačního rámce NativeScript. Převzato z [3].

zyku TypeScript[10] (dále jen TS), což je typově orientovaná nádstavba nad jazykem JavaScript. Během překladu dochází k transpilaci TS kódu s koncovkou .ts do JS souboru s příponou .js. Proto je každý JS soubor také platným TS souborem. Jeho silnou stránkou je přínos typové kontroly, kterou JS běžně neposkytuje. Kromě tříd přináší také rozhraní, výčtové typy nebo zkrácenou syntaxi pro anonymní funkce.

2.2.2 NativeScript a Vue.js

Vue.js je aplikační rámec jazyka JavaScript s otevřeným zdrojovým kódem určený pro tvorbu uživatelského rozhraní a jednostránkových aplikací. I když byl Vue.js na trh uveden teprve v roce 2014, jeho popularita mezi vývojáři rychle roste. Oblíbil si jej i Igor Randjelovic, který započal se snahou začlenit Vue.js do aplikačního rámce NativeScript.

10. `https://www.typescriptlang.org/`

Ukázalo se, že aplikace tvořené s Vue jsou druhým nejčastěji vytvářeným typem aplikací, což vedlo k vytvoření oficiální podpory. K té došlo na počátku roku 2019 a vznikl tak nový nástroj NativeScript-Vue [12].

Silnou stránkou NativeScript-Vue je, že se zaměřuje pouze na vrstvu zobrazení a snadno integruje s jinými knihovnami nebo stávajícími projekty (využívajícími například Angular). Je také vhodnou volbou, pokud je potřeba co nejmenších nároků na úložiště nebo paměť RAM. Podporuje také tzv. oboustranný data-binding, což je technika propojení mezi daty a UI vrstvou. To znamená, že změna v datové vrstvě vyvolá změnu v UI vrstvě a naopak. NativeScript-Vue toto propojení poskytuje přes `v-model`. Jde o syntaktickou zkratku zastupující propojený atribut i událost, která je vyvolána při změně jeho hodnoty. Po změně atributu dojde k aktualizaci dat (skrze událost) a dojde ke změně zobrazení v UI vrstvě ve všech komponentách, které jsou s atributem propojeny [5]. Příkladem je komponenta `TextField`, kde `v-model` zastupuje atribut `text` a událost `textChange`. Díky tomu je možné namísto:

```
<TextField :text="value" @textChange="setValue">
</TextField>
```

psát pouze:

```
<TextField v-model="value"></TextField>
```

NativeScript-Vue bohužel nepodporuje možnost routování[11] tak, jak je poskytuje samotný Vue.js. Nabízí však řadu alternativ, jednou z nich je například manuální routování [1]. Mezi nevýhody také patří nedostatek tutoriálů či příkladů řešení kvůli relativně nedávnému propojení aplikačních rámců.

2.2.3 NativeScript a Angular

Angular je open-source framework využívající TS určený pro tvorbu webových aplikací. Vychází ze svého předchůdce, kterým je framework AngularJS[12], oba vyvíjí společnost Google. Angular svou funkcionalitu NativeScriptu poskytuje od vydání verze NS 2.0.

11. Routování, nebo také směrování, je v mobilní aplikaci způsob přecházení mezi jednotlivými stránkami aplikace na základě předdefinovaných cest.
12. `https://angularjs.org/`

Jelikož Angular nevyužívá engine webového prohlížeče, je velmi snadné jej propojit s NativeScriptem. To je jeho obrovskou výhodou – sdílení kódu mezi webovou a mobilní aplikací. Pro vývojáře znajícího Angular je proto velmi snadné vyvíjet mobilní aplikaci skrze NS a Angular. Stejně jako Vue, i Angular poskytuje možnost oboustranného propojení dat s UI vrstvou skrze tzv. `[(ngModel)]`.

NativeScript využívá výhradně XML elementy, což znamená nemožnost použití HTML značek, jinak běžně používaných při vývoji aplikací s použitím aplikačního rámce Angular. Nevýhodou je také fakt, že Angular je náročný na operační paměť i velikost úložiště.

3 Informační systém skautIS a analýza agendy majetku

Tato kapitola popisuje informační systém skautIS včetně představení principu uživatelských účtů a rolí. Dále jsou popsány aplikační rozhraní a poskytované webové služby. Je představena testovací verze skautISu a také poskytované testovací rozhraní. Následně jsou rozebrány jednotlivé agendy a nakonec následuje podrobnější analýza agendy pro správu majetku.

3.1 Informační systém skautIS

SkautIS je rozsáhlý informační systém určený (nejen) pro členy organizace Junák – český skaut. Je vyvíjen společností Skeleton Software. Aplikace používá technologie .NET Framework 4 a ASP.NET 4, databázové služby poskytuje MS SQL Server 2008 R2. SkautIS má také vlastní aplikační rozhraní (dále jen API), které je realizováno webovými službami popsanými v sekci 3.2. Celý systém je rozdělen do několika vrstev, díky čemuž je jeho vývoj značně bezpečnější. Prezentační vrstva přistupuje k datům skrze API, proto je snadné aplikaci dále upravovat a rozvíjet. Systém také poskytuje jednotné přihlašovací rozhraní pro snadné připojení ostatních skautských aplikací [9].

Do skautISu se může zaregistrovat každý bez ohledu na věk či členství v organizaci [13]. Po registraci získává uživatel takzvaný **Základní uživatelský účet**. S tímto účtem se lze přihlásit i na další skautské weby (např. Skautská křižovatka[1], Buď připraven[2], weby junákem pořádaných akcí apod.). Jelikož spousta implementovaných agend vyžaduje řádné členství, je tento typ účtu značně omezený. Po propojení uživatelského účtu se skutečnými údaji z registrace získává uživatel přístup k dalším funkcím. Propojení se provádí na základě ověření údajů vedoucím jednotky a spárováním s využitím automaticky generovaného bezpečnostního kódu, který je unikátní pro každého uživatele. Propojením dochází k ověření, že daný uživatelský

1. Skautská křižovatka je informační portál pro členy Junáka. Stránky webu: `https://krizovatka.skaut.cz/`.
2. `https://www.budpripraven.cz/`

Obrázek 3.1: Dostupné funkce pro uživatele v roli vedoucího/admina jednotky (nahoře) a člena jednotky (dole) se liší.

účet patří reálné osobě s aktivním členstvím v organizaci. Pro vedoucí organizačních jednotek (středisek a výše), jejich zástupce a předsedy a členy revizních komisí je nabízen ještě další typ účtu, a to v podobě **Ověřeného uživatelského účtu**. Vzniká na základě úředně ověřeného dokumentu, jež prokazuje propojení uživatelského účtu s onou osobou.

Každá agenda ve skautISu je přístupná pouze uživateli, který vlastní příslušnou **roli**. Ta mu může být přidělena vedoucím, pokud má člen zřízený propojený uživatelský účet. Na základě těchto rolí se uživateli mění dostupné funkce uživatelského rozhraní (příklad na obrázku 3.1). Rolí může mít každý uživatel libovolné množství. Mezi základní role patří: Vedoucí/admin jednotky, Aktivní činovník

jednotky, Člen jednotky, Hospodář jednotky, Člen revizní komise, či Pověřená osoba STS. Pro tuto bakalářskou práci je důležitá již zmíněná role **Vedoucí/admin jednotky** a role **SKLADNÍK**. Každá role má specifikováno, které údaje může číst nebo měnit.

3.2 Webové služby

SkautIS má své vlastní API (dostupné na `https://ws.skautis.cz/`), přes které je možné komunikovat s jeho webovými službami. Kompletní přehled dostupných webových služeb je na adrese `https://test-is.skaut.cz/JunakWebservice/`.

Webové služby jsou rozdělené do kategorií, které rámcově odpovídají jednotlivým agendám skautISu. Ty zahrnují například práci s uživateli (kategorie **UserManagement**), či správu organizačních jednotek a registrovaných členů (kategorie **OrganizationUnit**). Pro tuto bakalářskou práci je nejdůležitější kategorie **Material**, která poskytuje služby pro správu vybavení a skladů.

Každá webová služba obsahuje několik funkcí. Ty jsou aplikací volány pomocí protokolu Simple Object Access Protocol (SOAP). Jedná se o protokol pro výměnu strukturovaných informací při implementaci komunikace s webovými službami, který je sestaven s využitím značkovacího jazyka XML. Dotazy jsou na server zasílány metodou POST využívající protokol HTTP. Zprávy mají jednotnou strukturu – po deklaraci hlavičky XML následuje obálka (`SOAP:Envelope`), která označuje, že daná XML zpráva je typu SOAP. Dále je vnořené tělo dokumentu (`SOAP:Body`), které obsahuje dotaz a odpověď. Dotazy i odpovědi se zapisují jako elementy, například funkce `PurchaseTypeAll` pro zjištění všech typů možností pořízení položky bude mít následující strukturu:

```
<PurchaseTypeAll xmlns="https://is.skaut.cz/">
  <purchaseTypeAllInput>
    <ID_Login>guid</ID_Login>
    <ID>string</ID>
    <DisplayName>string</DisplayName>
  </purchaseTypeAllInput>
</PurchaseTypeAll>
```

První element označuje název volané funkce. Následuje obal označující, že se jedná o dotaz (koncovka `Input`). Poté jsou již vkládány jednotlivé atributy, kde místo typu atributu je vložena konkrétní hodnota, tzn. pro ID je to například `<ID>152</ID>`. Odpověď je tvořena podobným formátem, akorát k názvu funkce je přidána koncovka `Response` a obal namísto `Input` na konci elementu obsahuje slovo `Result`. Pokud odpověď vrací pole, je každý prvek pole obalen dalším atributem s koncovým `Output`.

Popisy jednotlivých funkcí jsou bohužel mnohdy nedostatečné, většinou se jedná o velmi krátkou větu. Blíže nejsou popsány ani jednotlivé atributy a je třeba si domýšlet, co má daný atribut označovat.

3.3 Testovací rozhraní

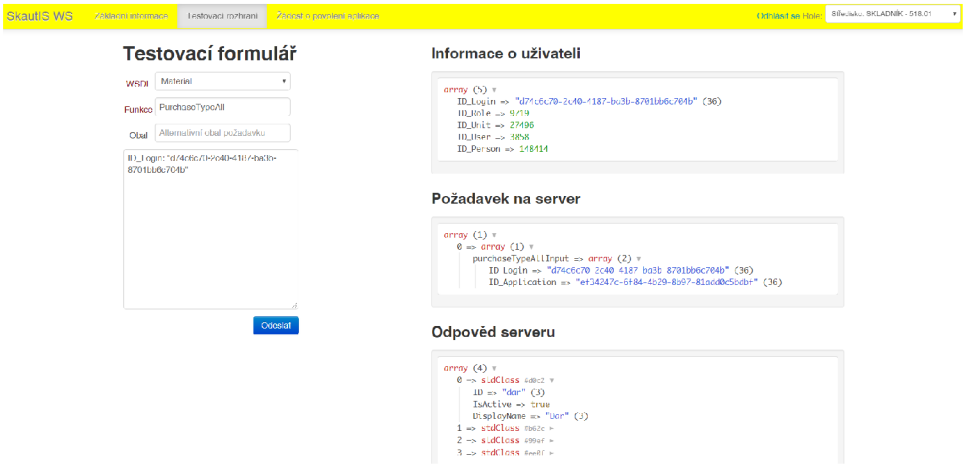
Pro seznámení se s funkcionalitou informačního systému skautIS a vyzkoušení všeho, co nabízí, existuje *Testovací skautIS*. Je shodný s reálným skautISem s tím rozdílem, že obsahuje smyšlená data. Od pravého skautISu je odlišitelný díky žlutému pozadí na všech stránkách. Pro přihlášení jsou dostupné testovací účty (viz `https://napoveda.skaut.cz/skautis/testovaci`). I v testovacím skautISu si lze vytvořit vlastní účet a přiřadit role, se kterými chce uživatel pracovat.

K vyzkoušení jednotlivých funkcí webových služeb je dostupné také testovací rozhraní (`https://ws.skautis.cz/testovani`). Na této stránce je možné si vyzkoušet, jaká data je nutné serveru poskytnout a jaká jsou naopak přijímána zpět. Ve formuláři se vybere webová služba, jedna z jejích funkcí, volitelně se přidává obal (některé funkce vyžadují obalující atribut) a v poslední kolonce se nastavují jednotlivé atributy s hodnotami. Příklad volání funkce `PurchaseTypeAll` a její odpovědi je zobrazen na obrázku 3.2.

3.4 Poskytované agendy

SkautIS je rozdělen do několika agend. Ty se uživateli zpřístupní, pokud má přiřazenu příslušnou související roli.

Mezi hlavní agendy patří sekce **Osoba**. V té jsou k dispozici informace o vlastní osobě, dále může uživatel vyhledat osobu, která je členem v jeho nebo v podřízené jednotce. Lze využít také vyhle-



Obrázek 3.2: Příklad volání funkce a její odpovědi v Testovacím rozhraní.

dávání v rámci **Skautského adresáře**, nalezené informace však smí být použity pouze pro účely spojené se skautingem, například pro kontaktování ohledně vzájemné spolupráce.

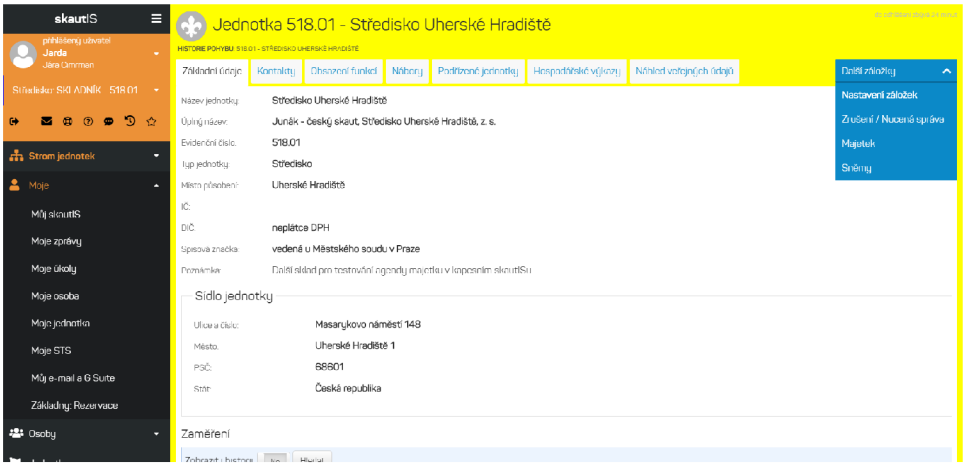
Další hlavní agendou je **Jednotka**. V ní má uživatel přístup k informacím své jednotky (v závislosti na roli vidí více nebo méně informací, s nejvyššími privilegii může informace i upravovat). Mezi ně patří např. základní informace, informace o členech a obsazení funkcí, informace o podřízených jednotkách, správa telefonních čísel v STS, správa rolí jednotky nebo správa majetku jednotky. Dále může uživatel vyhledávat základní informace o jiných jednotkách.

Agenda **Akce** poskytuje možnost evidovat nejrůznější typy akcí. Ty se dělí na Vzdělávací akce, Tábory, Sněmy a další. Uživatel se může k těmto akcím přihlásit, případně je i vytvářet, pokud vlastní příslušnou roli.

Dalšími hlavními agendami jsou například **Dotace** umožňující spravovat dotace jednotek, dále **Služby** poskytující správu vlastních čísel v STS, možnost rezervace majetku, či rezervaci základen. Následuje agenda **Kvalita** umožňující provádět/číst hodnocení kvality a nakonec agenda **Exporty** poskytující rozhraní pro vytváření šablon a export dat formou tabulkových výpisů.

3.4.1 Analýza agendy majetku

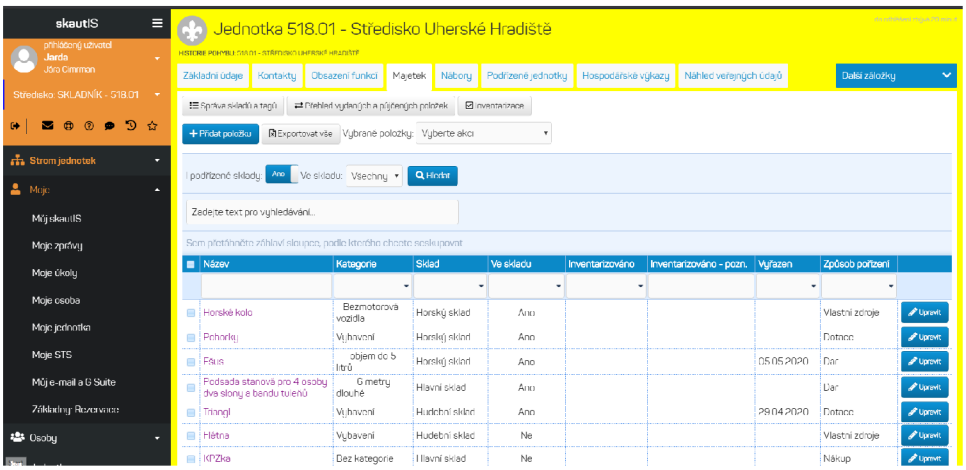
Každá organizační jednotka má zpravidla vlastní majetek. Jde např. o vybavení kluboven, táborové náčiní apod. SkautIS poskytuje služby, které umožňují majetek evidovat a spravovat, a to včetně procesu každoroční inventarizace. Do své jednotky se uživatel dostane přes *Moje* → *Moje Jednotka* nebo *Jednotky* → *Moje Jednotka*. Pokud uživatel vlastní roli Vedoucí/admin nebo SKLADNÍK, na stránce jednotky se nahoře mezi záložkami zobrazí karta Majetek (může být schovaná napravo v sekci Další záložky, viz obrázek 3.3). Po kliknutí na tuto kartu se načte stránka pro přehled majetku jednotky, viz obrázek 3.4.



Obrázek 3.3: Dostupné karty v sekci Jednotky z pohledu role SKLADNÍKA. Karta majetku je schovaná v boční roletce Další záložky.

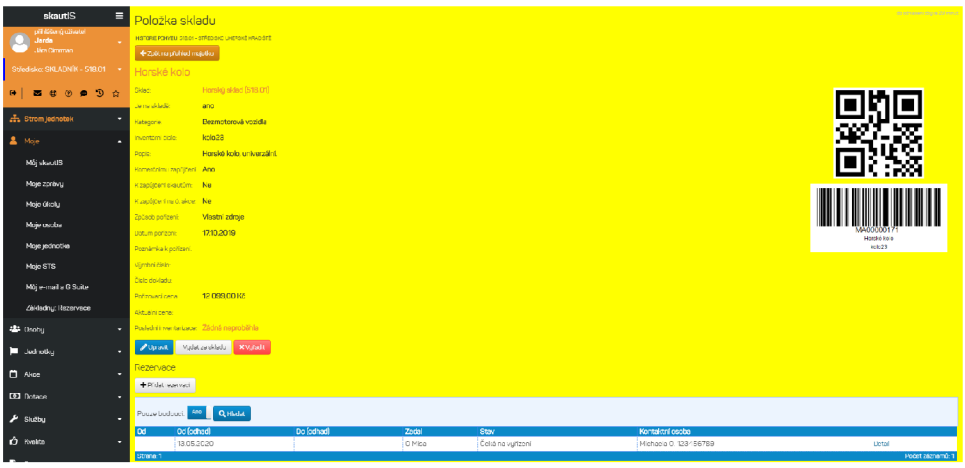
Majetek jednotky

Jestliže jednotka vlastní alespoň jeden sklad, zobrazí se seznam majetku jednotky. Ten lze po kliknutí na vybranou hlavičku tabulky seřadit, například podle názvu, kategorie či skladu. Majetek je také možné filtrovat: podle jména, zda se má zobrazovat majetek i z podřízených skladů nebo jestli se daná položka zrovna ve skladu nachází. Položky lze hromadně označit a provádět s nimi akce, která jsou v rozevírací roletce nad seznamem (například přidání položky do inventury).



Obrázek 3.4: Stránka pro správu majetku v informačním systému skautIS.

Po kliknutí na konkrétní položku se zobrazí stránka s jejím detailem (viz obrázek 3.5). Ta nabízí kromě informací o položce také možnost její úpravy či vyřazení majetku. Pokud je na položku vytvořena alespoň jedna rezervace, objeví se zde jejich přehled a tlačítko umožňující položku vydat.



Obrázek 3.5: Stránka s detailem položky v IS skautIS.

Během analýzy funkcionality za použití testovacího rozhraní bylo voláním funkce `WarehouseItemUpdate` zjištěno, že po odeslání pouze jednoho atributu v dotazu se ve skautISu po kliknutí na detail dané položky objeví chyba. V testovacím rozhraní se položka vypíše neúplná. Způsobují to chybějící atributy typu string, které se z nepochopitelných důvodů při jejich nevyplnění z objektu odstraní. Proto je nutné v dotazu posílat všechny atributy bez ohledu na změnu jejich hodnot.

Sklady jednotky

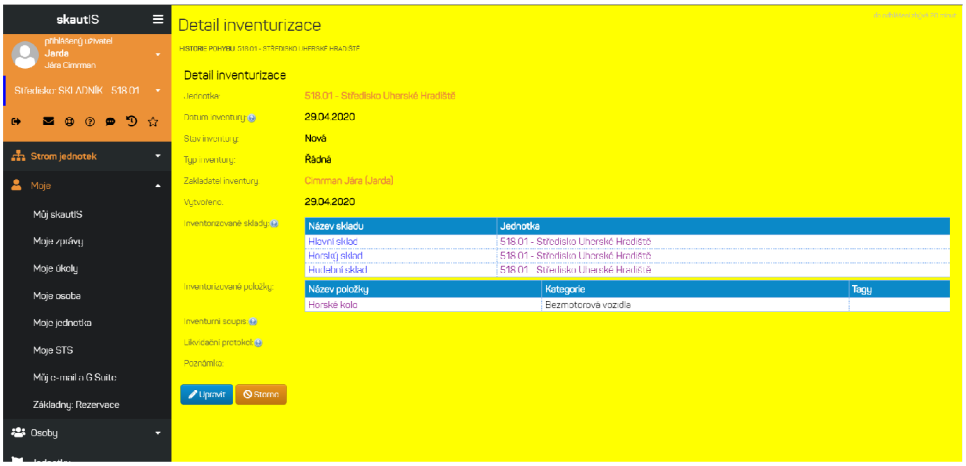
V horní části stránky přehledu majetku je umístěno tlačítko Správa skladů a uživatelsky definovaných štítků. Na stránce Správa skladů se nachází seznam všech skladů jednotky, včetně podřízených (viz obrázek 3.6). Sklady je opět možné seřadit kliknutím na vybraný sloupec v tabulce. Napravo se nachází tlačítko upravit, které přesměrovává na stránku s možností úpravy skladu. Vedle něj je tlačítko umožňující nastavit daný sklad jako výchozí. Po kliknutí na vybraný sklad dojde k přesměrování na předchozí stránku s přehledem majetku, nikoliv k zobrazení jeho detailu, jak by uživatel mohl očekávat. Na horní části stránky se nachází tlačítko umožňující přidat nový sklad, pod seznamem je sekce zaměřená na správu štítků (*tagů*) sloužících pro přehlednější práci s majetkem.

Obrázek 3.6: Stránka s přehledem skladů v IS skautIS.

Inventarizace jednotky

Vedle přehledu vydaných a půjčených položek je tlačítko Inventarizace, které odkazuje na stránku se seznamem realizovaných či aktuálních inventarizací. Na horní části stránky je popis, jak se má inventarizace provádět. Pro podrobnější nápovědu je pod popisem tlačítko odkazující na rozsáhleji popsaný postup provádění inventury. Následuje seznam ukončených, zrušených a aktivních inventur. V každé jednotce smí být aktivní nejvýše jedna inventura, nemusí však probíhat ve všech skladech. Po kliknutí na datum inventury se zobrazí její detail, kde je kromě základních informací uveden také seznam inventarizovaných skladů a soupis již inventarizovaných položek (viz obrázek 3.7). Samotná inventarizace probíhá přidáváním položek do inventury na stránce přehledu majetku (viz podsekce 3.4.1).

Po přidání inventarizovaných položek je nutné inventuru uzavřít nahráním likvidačního a inventárního soupisu. Aktuální inventarizaci je samozřejmě možné i zrušit.



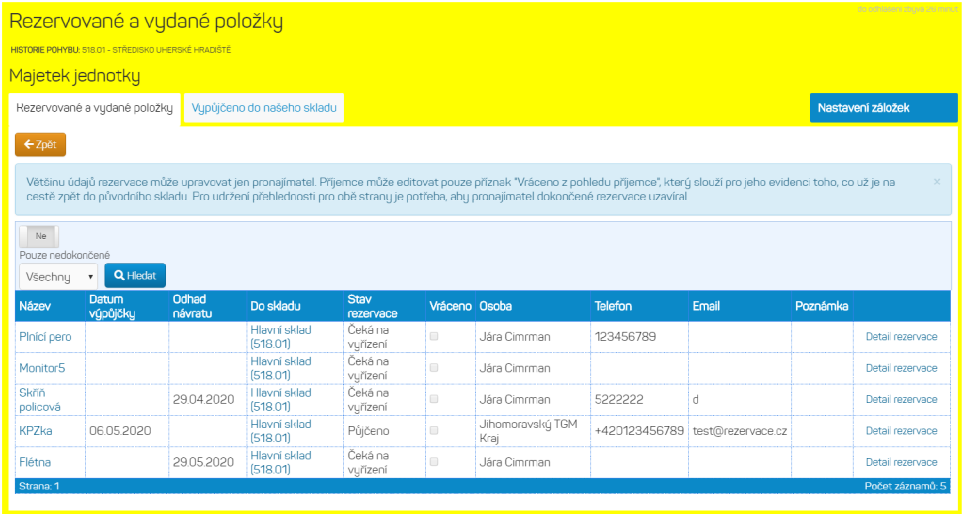
Obrázek 3.7: Stránka s detailem prováděné inventury v IS skautIS.

Při analýze agendy bylo objeveno několik chyb či nelogických postupů. Např. při vytvoření nové inventarizace se automaticky přidají všechny položky do soupisu již inventarizovaných položek. Po následné inventarizaci jedné či více položek výběrem ze seznamu na

předchozí stránce se seznam vyprázdní a přidají se (již správně) pouze právě inventarizované položky.

Správa rezervací jednotky

Vedle tlačítka Správy skladů se nachází tlačítko Přehled vydaných a půjčených položek vedoucí na stránku správy rezervací majetku. Tato stránka je rozdělena na dvě záložky. Tou první je sekce Rezervované a vydané položky, která slouží pro správu položek vypůjčených nebo rezervovaných jinými jednotkami. Druhá záložka se jmenuje Vypůjčeno do našeho skladu a podle názvu má uchovávat seznam položek vypůjčených uživatelem do jeho skladu. Bohužel je tato záložka nefunkční – po kliknutí naběhne stránka s chybovou hláškou, která uživateli oznamuje, že v aplikaci nastala chyba.



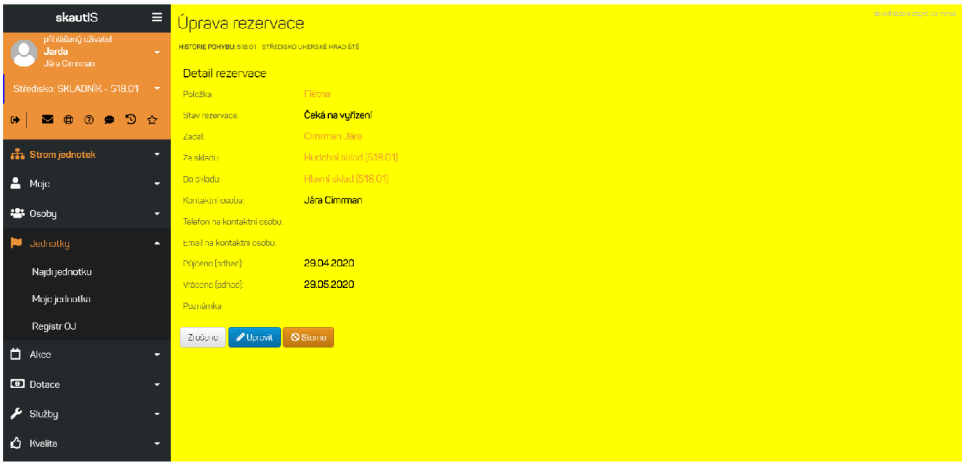
Obrázek 3.8: Seznam půjčených a rezervovaných položek.

Analýzou dané agendy za použití testovacího rozhraní bylo zjištěno, že položky, které by měly patřit do druhé záložky, se zobrazují v záložce vydaného majetku jednotky. Příklad na obrázku 3.8: KPZka patří právě přihlášenému uživateli, který ji vypůjčil jiné osobě, kdežto zbylé položky označují jeho dosud nevyřízené rezervace. Navíc, v seznamu vydaných položek se nezobrazují položky čekající na vydání.

Jejich soupis skautIS bohužel neposkytuje, o každé rezervaci je třeba rozhodnout v detailu dané položky.

Po kliknutí na libovolnou položku nastane jeden ze dvou případů: pokud je položka majetkem jednotky přihlášeného uživatele, otevře se její detail. Pokud však jde o položku, kterou si uživatel chce vypůjčit a z jeho skladu nepochází, naběhne chybová hláška s oznámením, že k detailu dané položky nemá uživatel přístup.

Na konci tabulky vydaných a půjčených položek se nachází tlačítko Detail rezervace. V případě, že se jedná o rezervaci vytvořenou uživatelem, zobrazí se její detail (obrázek 3.9). Jde-li o rezervaci vytvořenou jiným uživatelem, objeví se stránka oznamující nedostatečná práva pro čtení údajů této rezervace. Z toho důvodu nelze získat více informací o rezervaci a jejím příjemci. Dále ze seznamu nelze vyčíst, které položky jsou vypůjčené do uživatelova skladu a které jsou z něj naopak půjčovány. Jedním ze sloupců v tabulce se seznamem půjčeného majetku je sloupec Vráceno, který označuje, jestli byla položka vrácena z pohledu příjemce. Nad seznamem se píše, že tento příznak smí upravit pouze příjemce, nikoliv pronajímatel. Jelikož jsou položky, které si uživatel vypůjčil, obsažené v první záložce místo druhé, není možné tento příznak upravit. Proto ve skautISu momentálně není možné půjčené položky vracet.



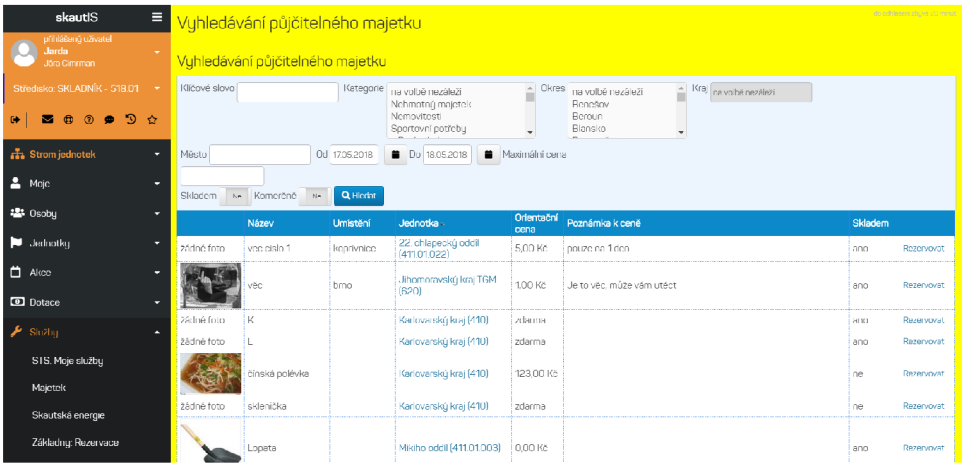
Obrázek 3.9: Detail rezervace vytvořené uživatelem.

Zpracování a postupy v této sekci jsou nepřehledné a většinou nelogické. Pro bližší porozumění systému rezervací a jejich přehledu je nezbytné zkoušení v testovacím rozhraní.

Rezervace majetku

Na levém panelu se v sekci Služby nachází odkaz Majetek odkazující na stránku s možností vytvářet rezervace položek. Jak je vidět na obrázku 3.10, na stránce se nachází seznam rezervovatelného majetku. Horní část stránky je určená k filtraci položek. Po kliknutí na tlačítko rezervovat, jež se nachází na konci řádku každé položky, dojde k přesměrování na rezervační formulář. Po vyplnění údajů a potvrzení dojde k vytvoření rezervace na danou položku.

Ve skautISu je sekce rezervovatelného majetku přístupná všem rolím. Nicméně žádosti o rezervace mohou vytvářet pouze uživatelé s oprávněním skladník a vedoucí/admin jednotky, kteří navíc musí ještě ve své jednotce mít založen alespoň jeden sklad.



Obrázek 3.10: Stránka se seznamem položek možných k zapůjčení.

4 Kapesní skautIS a existující aplikace pro správu majetku

V této kapitole je popsáno, jak je aplikace Kapesní skautIS navrhnuta a strukturována. Dále je nastíněno její uživatelské rozhraní a funkcionalita, kterou poskytuje. Na závěr je představena analýza již existující aplikace pro správu majetku, ze které tato bakalářská práce vychází.

4.1 Kapesní skautIS

Multiplatformní mobilní aplikace *Kapesní skautIS* vznikla v roce 2019 jako praktický výstup diplomové práce Mgr. Petra Domkaře [4]. Její vznik inicializovala snaha o zpřístupnění nejdůležitější funkcionality skautISu v mobilní aplikaci. Pro její vývoj byl vybrán framework NativeScript (společně s využitím rámce Angular) díky jeho schopnosti vyvíjet najednou na obě majoritní platformy současnosti – Android a iOS.

4.1.1 Struktura aplikace

Pro snadnou rozšiřitelnost využívá aplikace modulů, které poskytuje framework Angular. Aplikace je rozvržena do tří hlavních modulů.

Prvním z nich je **modul jádra** (Core module). Ten udržuje všechny služby, které se používají napříč celou aplikací. Obsahuje například entity potřebné k přihlášení do aplikace, volání webových služeb nebo správu stavu aplikace pro hlavní komponentu, kterou nalezneme v souboru `App.component.ts`.

Další částí jsou **sdílené moduly** (Shared modules), které uchovávají komponenty uživatelského rozhraní, jež mohou být použity ve více modulech. Typickým příkladem je navigační tlačítko s obrázkem umístěné na horní straně obrazovky. Dále poskytuje takzvané roury (v angličtině *pipes*) sloužící k propojení jednoho výstupu přes libovolnou transformační funkci na výstup jiný. Například roura pro telefonní číslo transformuje devítimístné číslo do čitelnějšího formátu s národní předvolbou +420 a mezerami mezi trojčíslími.

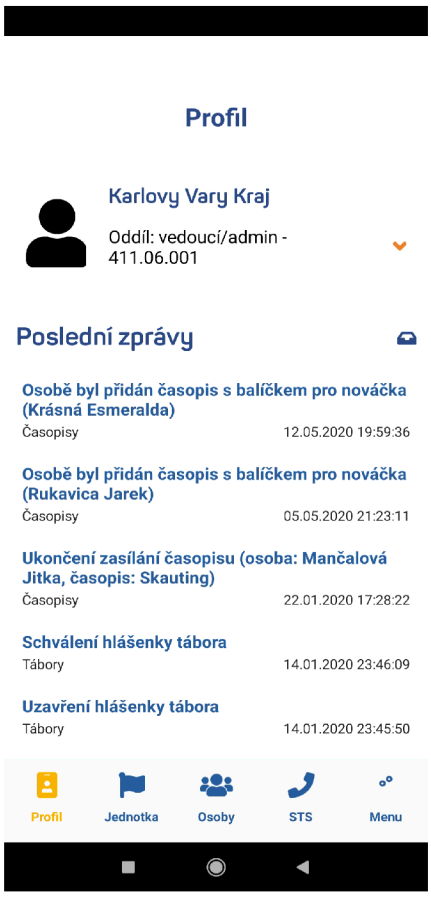
Posledním hlavním modulem je takzvaný **modul agend** (Features module). Ten obsahuje jednotlivé moduly aplikace. Výhodou rozdělení aplikace do modulů je jejich načítání až v případě, kdy je modul potřeba. Jde o návrhový vzor známý jako líné načítání (z angl. *lazy-loading*). Každá agenda se skládá z následujících částí:

- hlavní modul dané agendy importující všechny komponenty agendy,
- navigační modul uchovávající navigaci mezi stránkami,
- složka `Components` obsahující komponenty částí obrazovek,
- složka `Containers`, ve které se nacházejí jednotlivé stránky agendy,
- složka `Entities`, která obsahuje definice tříd používajících se pro volání webových služeb,
- složka `State-Mgmt`, která ztvárňuje možnost správy stavu dané agendy,
- složka `Services` poskytující specifické služby pro daný modul.

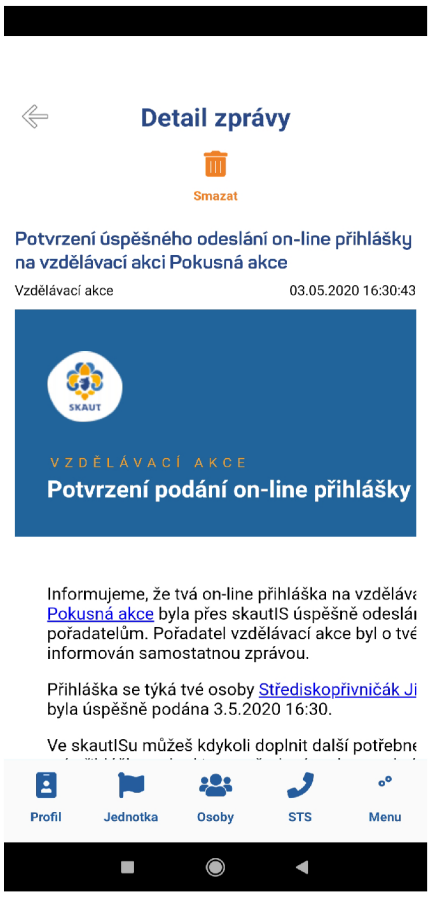
4.1.2 Vzhled aplikace

Aplikace od svého vzniku již několikrát změnila svůj design, který se stále vyvíjí. V současné podobě obsahuje dolní navigační panel, který se zobrazuje na všech obrazovkách. Nabízí 5 tlačítek: Profil, Jednotka, Osoby, STS a Menu. Poslední zmíněná položka otevírá postranní navigační panel. Ten lze také vysunout tažením z levé strany obrazovky doprava. Obsahuje jméno uživatele, roletku s možností přepínání role, dále implementované agendy (viz podsekce 4.1.3), tlačítko s odkazem na stránku O aplikaci a nakonec možnost odhlášení.

Aplikace využívá barev z grafického manuálu Junáka. Základem je bílé pozadí a využívány jsou především odstíny modré a oranžové, které vychází z odstínů použitých v logu organizace. Navigační tlačítka jsou většinou modré, ostatní akční tlačítka oranžové. Je-li na stránce potřeba vykonat nějakou akci (odkázat se na jinou stránku, něco smazat či upravit atp.), používá se horní navigační panel s tlačítkem obsahujícím oranžový text a obrázek. Pokud některá z hlavních agend umístěná na spodním navigačním panelu odkazuje na jinou stránku, která z tohoto panelu není dostupná jedním kliknutím, objeví se v horní části obrazovky akční panel s názvem stránky a tlačítkem zpět.



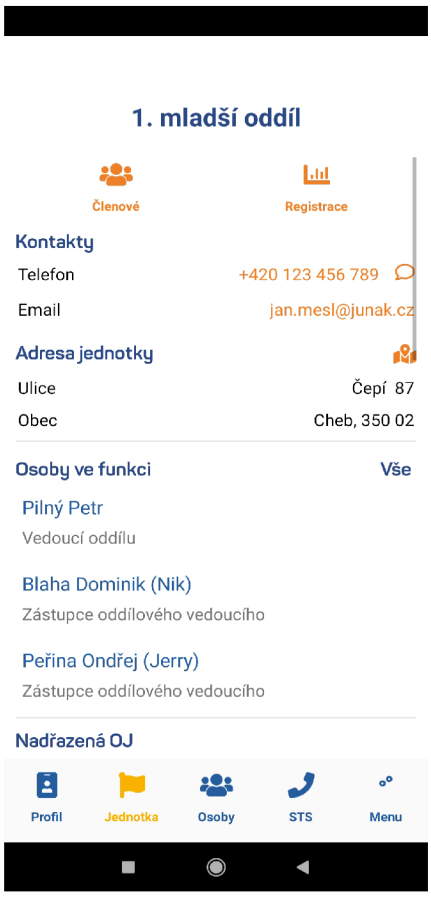
Obrázek 4.1: Stránka uživatelova profilu.



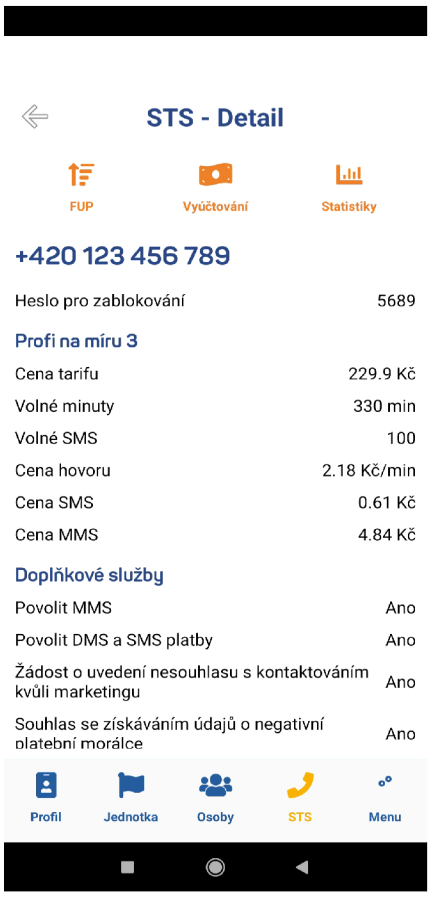
Obrázek 4.2: Detail zprávy v Kapesním skautISu.

4.1.3 Implementované části a agendy

Aplikace Kapesní skautIS zatím poskytuje jen několik agend webového skautISu. První stránka, která se po přihlášení do aplikace zobrazí, je *Profil* uživatele (obrázek 4.1). Ten obsahuje profilový obrázek, jméno uživatele a jeho roli, kterou lze díky rozevírací roletce měnit. Po kliknutí na obrázek se aplikace přesměruje na detail dané osoby. V něm se zobrazí kontakty na uživatele, jeho adresa a další informace (v závislosti na roli přihlášeného uživatele). Pod obrázkem a jménem se zobrazuje část z agendy *Zprávy*. Na stránce profilu je vidět posledních 5 nepřečtených zpráv. Do agendy Zprávy se uživatel dostane skrz



Obrázek 4.3: Detail jednotky uživatele.



Obrázek 4.4: Detail čerpání tarifu v Kapesním skautISu.

postranní panel. Ta mu umožňuje prohlížet a mazat vybrané zprávy. Detail zprávy zobrazuje obrázek 4.2. Další implementovanou agendou je *Jednotka* (obrázek 4.3). Obsahuje základní informace vybrané jednotky, informace o jejích členech a statistiky registrace. Umožňuje zobrazit související jednotky (nadřazenou a podřazené). Zobrazují se zde také osoby ve funkci, po kliknutí na libovolnou z nich se otevře detail osoby. Agenda *Osoby* umožňuje vyhledávání osob. Po kliknutí na nalezenou osobu se otevře její detail. Lze hledat jak v jednotkách, tak ve skautském adresáři. Poslední implementovanou agendou je *STS*, která umožňuje spravovat svá čísla ve skautské telefonní síti, zob-

razovat jejich detail (viz 4.4), statistiky využívání, faktury či možnost navýšení datového limitu.

4.2 Mobilní aplikace pro správu majetku

V roce 2018 byla v rámci bakalářské práce Bc. Jana Crhy [2] vyvinuta multiplatformní mobilní aplikace umožňující spravovat skautský majetek. Aplikace byla vyvinuta za pomocí aplikačních rámců NativeScript a Angular.

Po přihlášení dojde k přesměrování na stránku s možností výběru jednotky nebo provádění rezervací. Po vybrání jednotky aplikace uživatele přesměruje na stránku se seznamem skladů. Kliknutím na libovolný sklad dojde k přesměrování na stránku s přehledem majetku skladu s možností jeho inventarizace. Stránka obsahuje dvě záložky umožňující přepínaní mezi inventarizovanými a neinventarizovanými položkami. Aplikace poskytuje možnost offline inventarizace díky uložení položek do lokální databáze. Dále je možné inventarizovat položky skenováním QR nebo čárových kódů. Na stránce s volbou jednotky se na dolním okraji nachází tlačítko odkazující na stránky s možností rezervace položek. Stránka obsahuje seznam položek možných k zapůjčení. Po výběru položky se zobrazí vyskakovací okno potřebné k potvrzení rezervace.

Aplikace nebyla po svém dokončení dále udržována a vzhledem k rozsáhlým změnám provedených ve webových službách již není plně funkční.

5 Návrh agendy majetku

Tato kapitola je věnována návrhu jednotlivých částí agendy. V první části jsou představeny funkční a nefunkční požadavky kladené na novou agendu. Následně je na základě těchto požadavků znázorněn diagram případu užití. Pro návrh obrazovek byla použita aplikace MockFlow[1]. Přehled jednotlivých návrhů stránek se nachází v příloze B. Postupně je na základě systémových požadavků představen návrh všech částí agendy majetku.

5.1 Systémové požadavky

Rozšíření Kapesního skautISu o agendu majetku vychází z již existující aplikace popsané v sekci 4.2. Vzhledem k nedávným změnám aplikačního rozhraní skautISu je potřeba většinu její funkcionality navrhnout znovu. Navíc je vhodné funkcionalitu rozšířit o další funkce, které původní aplikace neimplementuje. V následujících podkapitolách jsou rozepsány požadavky potřebné k implementaci dané aplikace.

5.1.1 Funkční požadavky

Funkční požadavky představují souhrn funkcionality, která je po aplikaci vyžadována. Přehled všech funkčních požadavků agendy majetku zobrazuje následující seznam:

- Uživatel může pracovat s agendou majetku, pokud vlastní přístupná práva vedoucí/admin nebo SKLADNÍK.
- Uživatel může prohlížet majetek své jednotky.
- Uživatel může vyhledávat v majetku své jednotky.
- Uživatel může zobrazit vyřazené položky ze svého skladu.
- Uživatel může prohlížet detail položek své jednotky.
- Uživatel může upravit položku své jednotky.
- Uživateli se zobrazí rezervace položky v jejím detailu.
- Uživatel může vydávat položky ze své jednotky.
- Uživatel může vyřadit položku ze svého skladu.
- Uživatel může prohlížet sklady své jednotky.

1. `https://www.mockflow.com/`

- Uživatel může vyhledávat mezi sklady své jednotky.
- Uživatel může zobrazit detail skladu své jednotky.
- Uživateli se zobrazí v detailu skladu jeho majetek.
- Uživatel může založit novou inventuru v případě, že neprobíhá žádná aktivní.
- Uživatel může provádět inventuru svého majetku.
- Aplikace umožňuje provádět inventuru offline.
- Uživatel může provádět inventuru skenováním QR nebo čárových kódů.
- Uživatel může zobrazovat majetek čekající k inventarizování, čekající na potvrzení a již inventarizovaný.
- Uživatel může prohlížet majetek nabízený k rezervaci.
- Uživatel může majetek určený k rezervaci vyhledávat a filtrovat.
- Uživatel si může majetek rezervovat.
- Uživatel může spravovat (upravovat, rušit) své rezervace.

Na základě funkčních požadavků byl vytvořen diagram případů užití (obrázek 5.1).

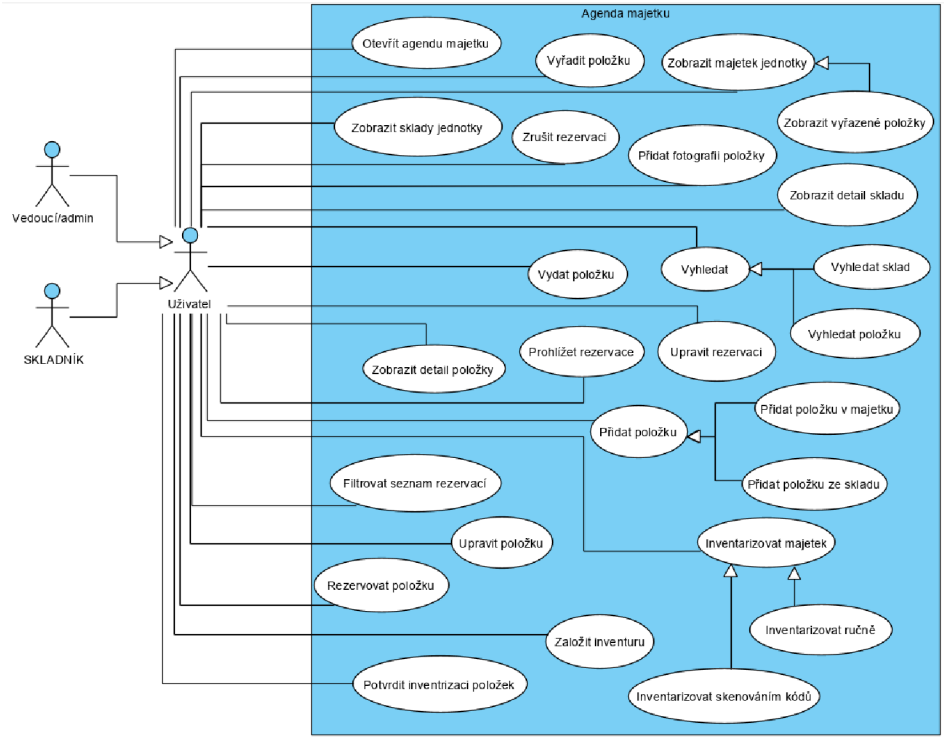
5.1.2 Nefunkční požadavky

Omezení kladené na daný systém (zahrnující například rychlost, zabezpečení, spolehlivost či výkon) popisují tzv. nefunkční požadavky. Následující seznam poskytuje souhrn nefunkčních požadavků na agendu majetku:

- Agenda majetku je vyvinuta za použití aplikačních rámců NativeScript a Angular.
- Agenda je součástí multiplatformní aplikace Kapesní skautIS.
- Rozšíření aplikace o agendu majetku je přidáno v rámci nového modulu pro zachování modularity a snadné správy aplikace.
- Agenda vyžaduje (až na část inventarizace) připojení k internetu kvůli volání webových služeb.

5.2 Návrh sekce majetku

V rozcestníku je sekce dostupná hned pod prvním odkazem s názvem **můj majetek**. Hlavní komponentou této části je seznam majetku

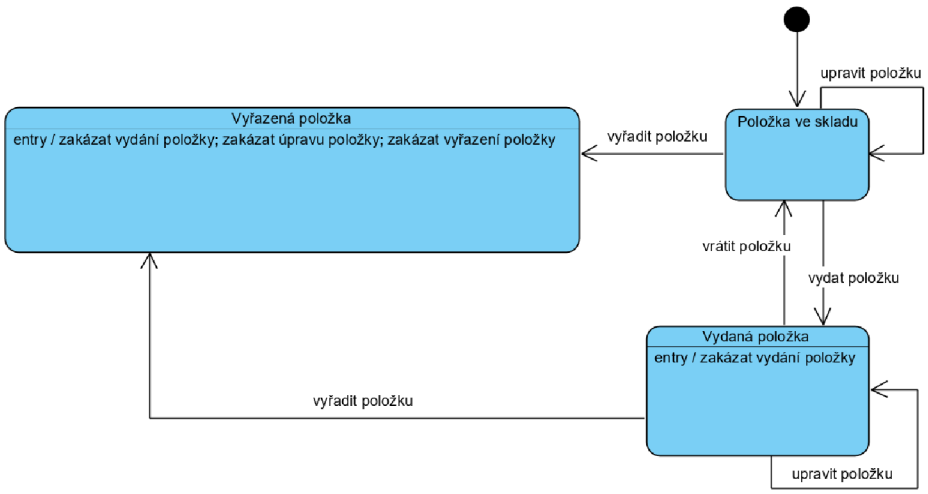


Obrázek 5.1: Diagram případů užití znázorňující systémové požadavky.

jednotky společně s možností zobrazení detailu položky. Tabulka se seznamem majetku obsahuje dva sloupce: první s fotografií a druhý s názvem položky. V případě, že je k majetku přiřazena fotografie, dojde po jejím kliknutí na zvětšení obrázku. Kliknutím na zbytek řádku dojde k přesměrování na detail položky.

Detail položky obsahuje podrobné informace o položce. Stavy, kterých může položka nabývat, znázorňuje diagram na obrázku 5.2. Stránka nabízí možnost úpravy položky (včetně úpravy fotografie), vyřazení položky nebo vydání položky. To je možné pouze v případě, že na položku existuje alespoň jedna čekající rezervace. Seznam rezervací je zobrazen ve spodní části detailu položky.

V detailu položky je mimo jiné uveden sklad, ve kterém se položka nachází. Název skladu je vytvořen ve formě odkazu, tzn. po jeho kliknutí dojde k přesměrování na stránku s detailem daného skladu.



Obrázek 5.2: Stavový diagram sekce majetku.

Pro snadnější filtrování položek obsahuje stránka s přehledem majetku možnost vyhledávání podle názvu. Dále nabízí možnost zobrazení vyřazených položek.

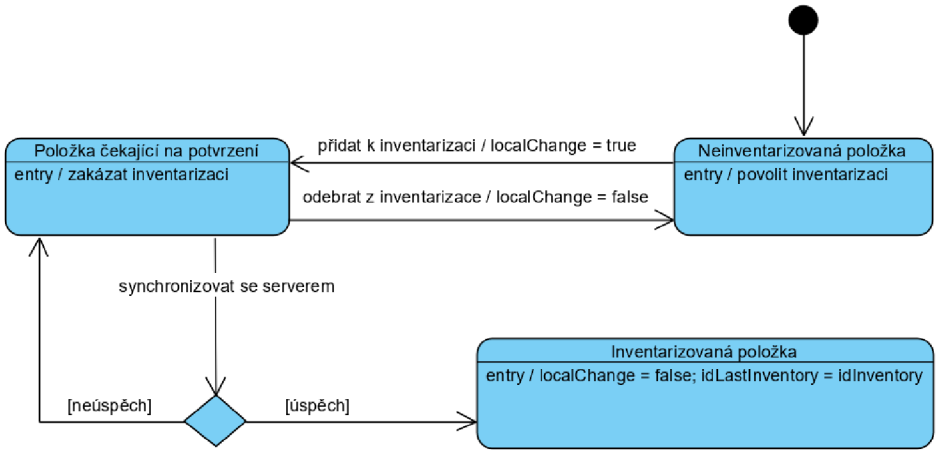
5.3 Návrh sekce skladů

Druhé tlačítko v rozcestníku nese název **moje sklady**. Po kliknutí dojde k zobrazení stránky podobné stránce majetku: na horní části je vyhledávací pole, pod ním seznam skladů jednotky. Po kliknutí na daný sklad se objeví jeho detail s možností úpravy.

Kromě popisných informací je v detailu skladu uveden jeho nadřazený sklad (v případě, že nějaký má), jednotka, které sklad patří a seznam majetku skladu. Všechny z vyjmenovaných umožňují po kliknutí přejít na stránku s jejich detailem.

5.4 Návrh sekce inventarizace

Proces inventarizace je ve skautISu trochu matoucí, proto bylo cílem tuto sekci co nejvíce zjednodušit. Stavy položek procházejících procesem inventarizace jsou popsány na obrázku 5.3. Po kliknutí v rozcest-



Obrázek 5.3: Stavový diagram procesu inventarizace.

níku na stránku **inventarizace** je zobrazen seznam skladů dané jednotky. Po výběru skladu se ověří, zda v něm probíhá aktivní inventura. Pokud ne, v horním panelu se zpřístupní tlačítko s možností založení nové inventury. Pokud již inventura založena je, dojde k přesměrování na stránku se třemi záložkami: Neinventarizováno, K potvrzení a Inventarizováno.

Záložka *Neinventarizováno* obsahuje seznam položek daného skladu, které ještě nebyly přidány k inventarizaci. Na horním panelu pod záložkami se nachází tlačítko Skenování kódů sloužící pro skenování QR a čárových kódů položek ve skladu. Pokud se uživatel rozhodne inventuru provádět ručně, může k tomu použít tlačítko na konci řádku v seznamu položek.

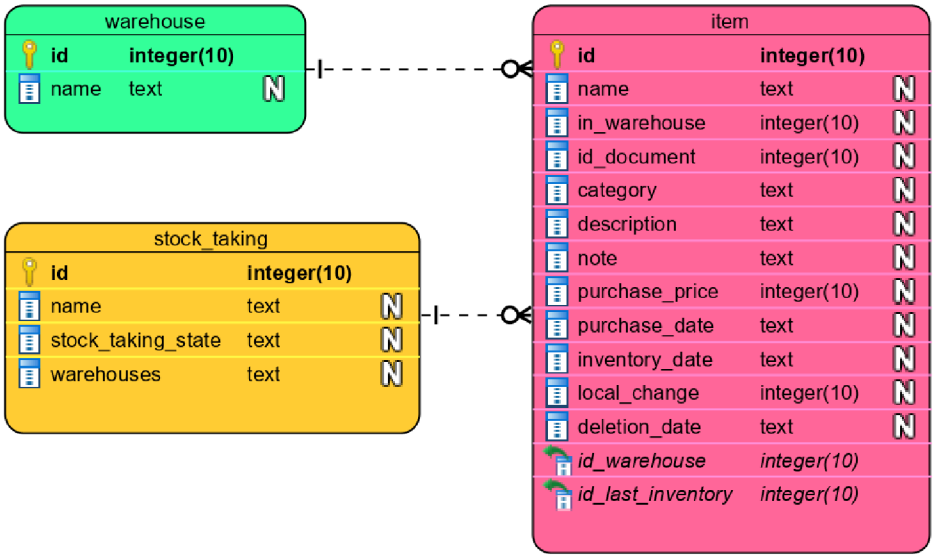
Ve skautISu dochází k inventarizaci položek jejich přidáním do právě prováděné inventury. Vzhledem k možnosti provádět inventuru v Kapesním skautISu offline bylo nutné kromě neinventarizovaných a již inventarizovaných položek přidat jednu sekci navíc, a to pro položky přidané k inventarizaci, které zatím nebyly synchronizovány se serverem. Všechny položky, které jsou přidány jak ručně, tak skenováním, se proto zobrazí na druhé záložce se jménem *K potvrzení*.

V případě, že si uživatel inventarizaci položky před potvrzením inventarizace rozmyslí, je na konci řádku položky v záložce K potvrzení křížek, který ji vrátí zpět do skupiny neinventarizovaných položek. Pro

potvrzení inventarizace slouží tlačítko Potvrdit nacházející se v horním panelu druhé záložky. V poslední záložce *Inventarizováno* je dostupný přehled položek již potvrzených, tzn. synchronizovaných se serverem.

Jakmile je inventura provedena a potvrzena, je nutné ji uzavřít nahráním likvidačního a inventurního soupisu ve webovém rozhraní skautISu. Tato funkcionalita nebyla do mobilní aplikace implementována z důvodu nutnosti formuláře vytisknout, podepsat a posléze opět naskenovat. Proto není smysluplné zahrnovat tuto část do mobilní aplikace, která má sloužit primárně k řešení fyzické inventury majetku.

5.4.1 Návrh databáze



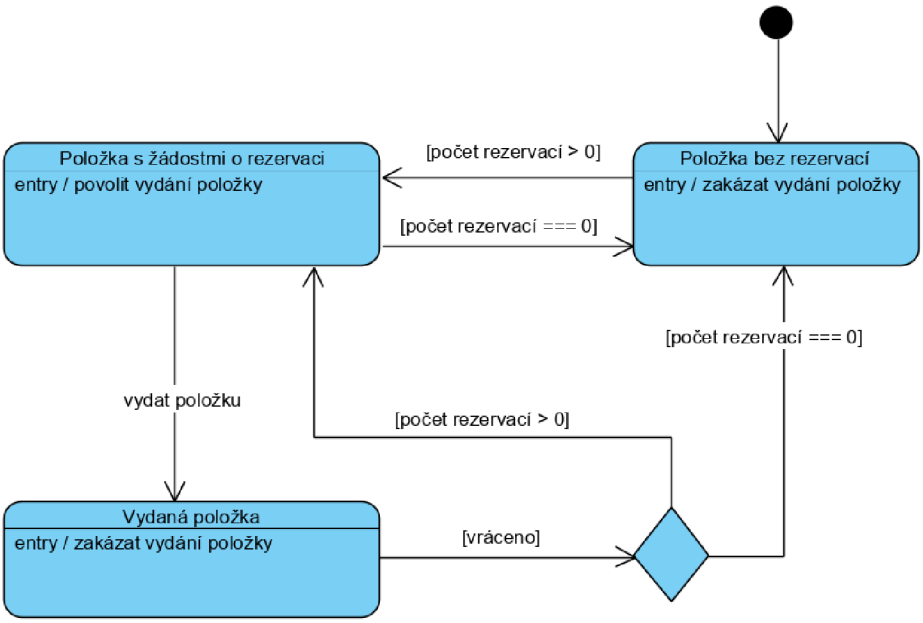
Obrázek 5.4: ERD diagram návrhu databáze pro umožnění procesu offline inventarizace.

Pro uchování dat lokálně v zařízení je zapotřebí použití databáze. Jak je vidět na obrázku 5.4, návrh databáze sestává ze tří tabulek: **warehouse** (sklad), **stock_taking** (inventura) a **item** (položka). Tabulka warehouse obsahuje identifikátor, který je používán pro načtení položek konkrétního skladu. Atribut `name` je ukládán z důvodu propojení

s tabulkou stock_taking, neboť informace o skladech dané inventury zasílá webová služba v podobě jednoho atributu – `warehouses`. Jeho hodnotou je pole názvů skladů převedené na typ řetězce. Tento atribut bohužel porušuje první normální formu (1NF), což znamená, že některý z atributů není atomický (obsahuje více hodnot). Vzhledem k uchovávání dat pouze pro účel načtení položek a pro jednodušší a rychlejší práci s databází nebyla přidána propojující tabulka řešící porušení 1NF. Atributy v tabulce item slouží pouze k popisu položky.

5.5 Návrh sekce rezervací majetku

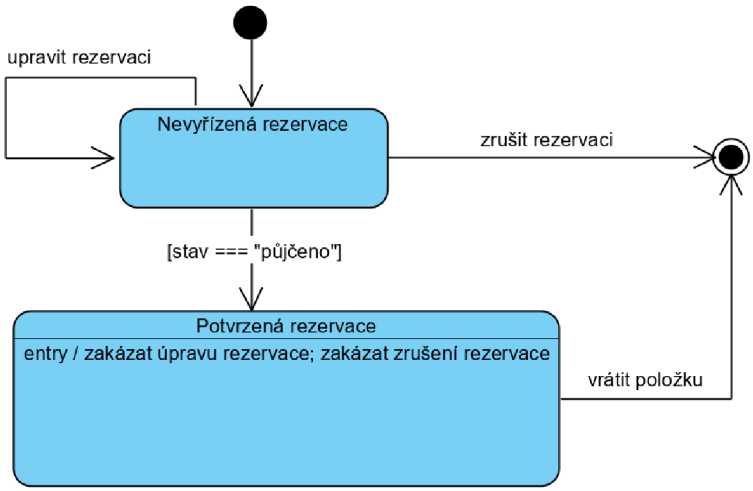
Proces rezervace implementovaný ve skautISu není vždy zcela logický a implementace obsahuje několik chyb (viz sekce 3.4.1). To ovlivnilo i výslednou podobu této agendy v mobilní aplikaci.



Obrázek 5.5: Stavový diagram položky v průběhu rezervace ze strany pronajímatele.

Správa rezervací je dostupná z rozcestníku pomocí tlačítka **přehled rezervací**. Po kliknutí se zobrazí stránka se dvěma záložkami: *Vydané*

položky (stavy položky na obrázku 5.5) a *Vypůjčeno do našeho skladu* (stavy rezervace na obrázku 5.6).



Obrázek 5.6: Stavový diagram rezervace majetku z pohledu příjemce.

V první záložce jsou zobrazeny položky, které uživatel půjčil jiné jednotce. Pro co nejjednodušší práci s rezervacemi měla stránka obsahovat také přehled všech položek, které čekají na vyřízení rezervace ze strany pronajímatele. Bohužel, webové služby skautISu neposkytují hromadný přehled těchto položek. Seznam rezervací ke každé položce je proto dostupný pouze v jejím detailu. Druhá záložka naopak ukazuje položky vypůjčené do uživatelova skladu z jiné jednotky a položky s dosud nevyřízenou rezervací. Nepotvrzené rezervace je možné upravit nebo zrušit.

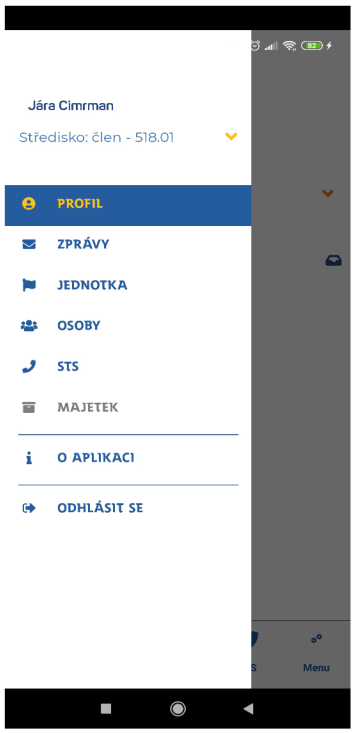
Kromě správy rezervací je v aplikaci umožněno majetek rezervovat. K tomu slouží v rozcestníku odkaz **rezervovat**. Stránka nabízí seznam všech položek, které mají v detailu uvedenou možnost půjčení. Na horní části obrazovky je dostupné filtrování položek v podobě vyhledávání podle jména, města nebo podle maximální ceny. Po kliknutí na položku se rozbalí její detail. Napravo je umístěno tlačítko rezervovat, které uživatele přesměruje na stránku, kde provede rezervaci. Po vytvoření tato rezervace přibude na stránce *přehledu rezervací* v záložce *Vypůjčeno do našeho skladu*.

6 Implementace agendy majetku

Tato kapitola se zabývá implementací funkcí popsaných v předešlé kapitole do aplikace Kapesní skautIS. Nejprve jsou popsány obecné principy použité při implementaci. Dále jsou představeny jednotlivé části aplikace včetně změn oproti návrhům vzniklých většinou na základě nefunkčních či chybějících volání webových služeb.

6.1 Obecné principy

Z důvodu zachování modularity (viz podsekce 4.1.1) se téměř všechny zdrojové kódy implementace agendy majetku nachází v `app/pages/material`. Ve výjimečných případech bylo nutné zasahovat do hlavních komponent. Jedním z nich bylo přidání volání webové služby pro zjištění všech rolí uživatele. Díky tomu mohlo dojít ke zpřístupnění agendy **MAJETEK** pro role vedoucí/admin a SKLADNÍK, ostatním rolím není agenda dostupná, viz obrázek 6.1. Dalším zásahem bylo přidání povolení[1] pro zjištění stavu internetového připojení kvůli potvrzení inventarizace.



Obrázek 6.1: Agenda majetku není členovi dostupná.

Veškerá volání webových služeb jsou realizována třídou `WebApiService`, která je umístěna v modulu jádra. Třída obsahuje funkci `call` sloužící pro volání HTTP metody POST. Jedním z parametrů metody je dotaz ve formátu XML. Ten je vytvořen pomocí metody `getEnvelope`. Data, která metoda POST vrací, jsou převedena z formátu XML do

1. Povolení udávající, k čemu má aplikace v zařízení přístup, se pro Android nastavují v souboru: `app\App_Resources\Android\src\main\AndroidManifest.xml`

instance určené entity a obalena typem *Observable*, který reprezentuje asynchronní tok dat. Lze si jej představit jako pole, do kterého hodnoty přicházejí asynchronně. Pro čtení *Observables* je k dispozici metoda `subscribe`, která umožňuje reagovat na změny stavů dané *Observable* [6].

Při volání některých funkcí webových služeb bylo zjištěno, že metoda `getEnvelope` nepočítá s atributem typu pole a kódování v hlavičce neodpovídá požadovanému UTF-8[2], které bylo pro implementaci agendy majetku nezbytné. Po nahlášení problémů autor aplikace zmíněné nedostatky obratem vyřešil.

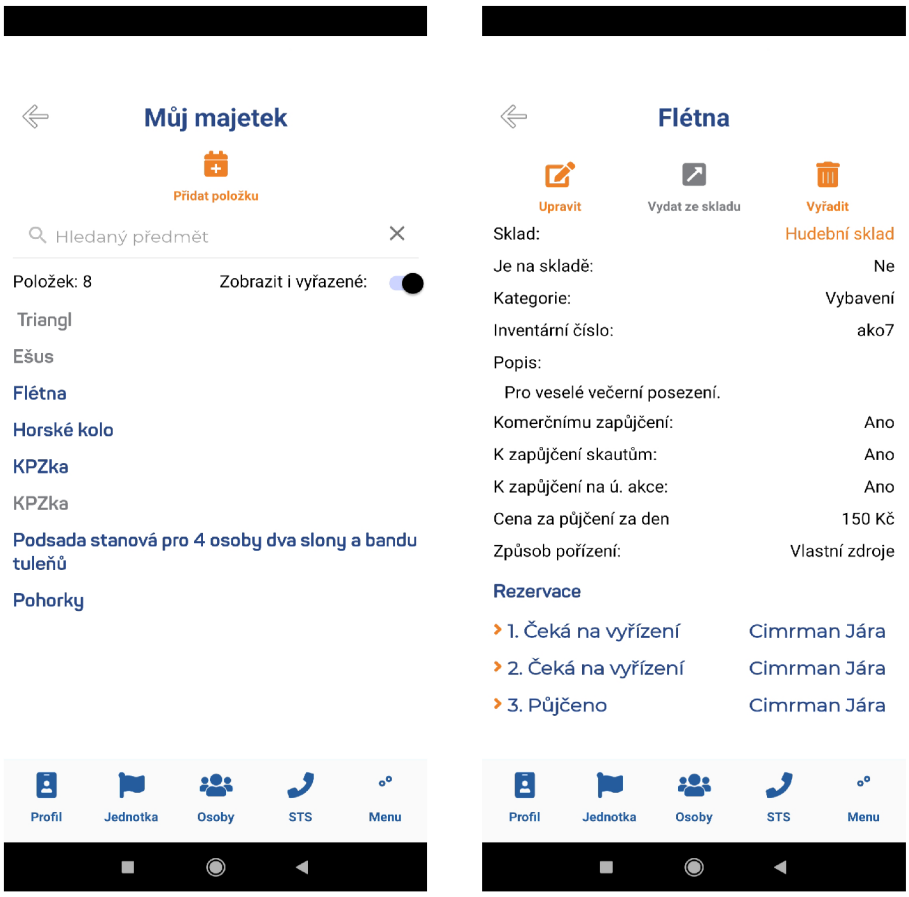
Na několika místech agendy bylo potřeba umožnit vepsání data (např. při vytvoření rezervace). Pro tento účel byla vybrána vestavěná komponenta `Date Picker`. Její zobrazení však zabírá velkou část obrazovky. To vedlo k vytvoření komponenty `ModalDatePickerComponent`, která je realizována formou modálního dialogu a slouží pro výběr data.

6.2 Majetek

Návrh agendy majetku obsahoval možnost zobrazení fotografie položky a možnost její změny pořízením či nahráním nové přímo z telefonu. Tato funkcionalita nakonec implementována nebyla z důvodu změn ve webovém rozhraní. Dříve byl obrázek poskytován ve formě *base64*, nyní je to *ID_Dokumentu*, které označuje uložení obrázku na externím úložišti Google Drive. Pro přístup k souboru však neexistuje volání webové služby. Tato chybějící funkcionalita byla nahlášena vývojářům skautISu. Potřebná webová služba do odevzdání práce nebyla implementována. Po domluvě s vedoucím práce tak aplikace neposkytuje zobrazení fotografií jednotlivých položek ani možnost jejich pořízení. Byla však alespoň implementována funkcionalita pro pořízení fotografie, resp. její zobrazení v budoucnu.

Po testování, které je popsáno v kapitole 7, byla do této sekce přidána možnost vložení nového majetku. Odkazuje na něj tlačítko Přidat položku na stránce s přehledem majetku. Další změnou, která vzešla z testování, je stále dostupné tlačítko s možností vydání položky. V pří-

2. Kódování UTF-8 obsahuje i znaky s diakritikou na rozdíl od kódování ASCII.



Obrázek 6.2: Seznam majetku jednotky včetně zobrazení vyřazených položek (vlevo) a detail položky (vpravo).

padě, že na položku není vypsaná žádná rezervace nebo je položka již vypůjčená, je toto tlačítko pouze označeno jako nedostupné.

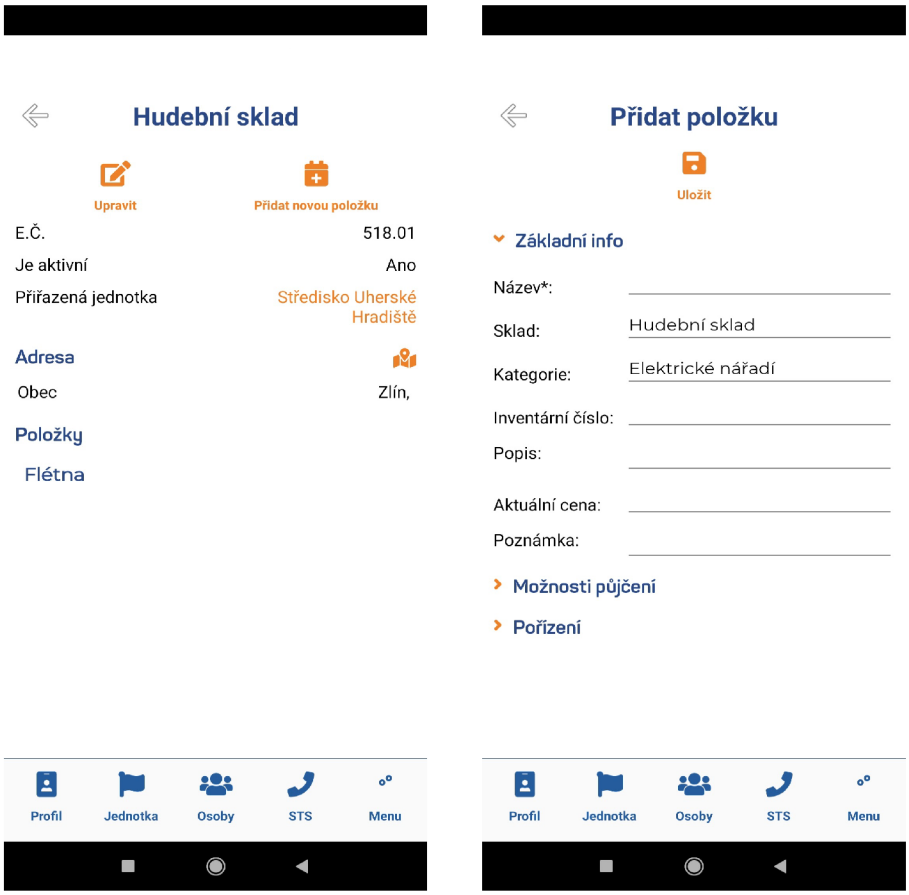
Seznam majetku a detail položky se nachází na obrázku 6.2.

6.3 Sklady

Implementace skladů proběhla podle návrhu. Díky výstupům z testování a přidání možnosti vložit novou položku byla dodatečně přidána možnost vytvoření nového majetku přímo v detailu skladu. Rozdílem

mezi vytvořením položky v seznamu majetku a přímo ze skladu je pouze v předvyplnění vybraného skladu.

Detail skladu a vytvoření nové položky do skladu je ilustrováno na obrázku 6.3.

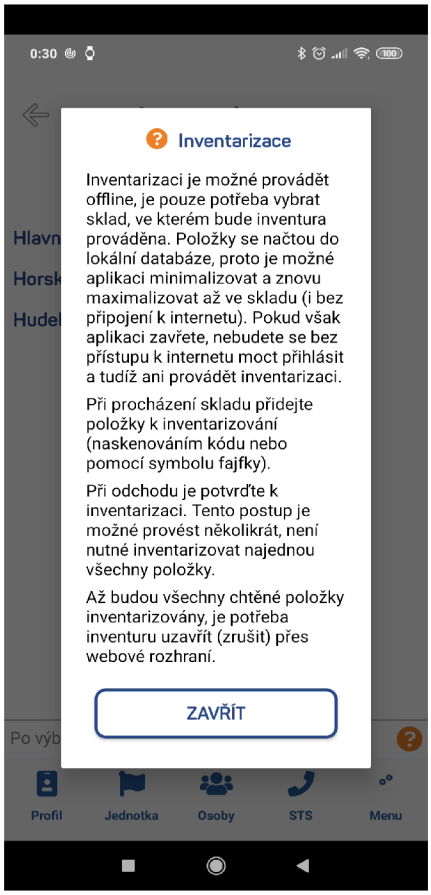


Obrázek 6.3: Detail skladu (vlevo) a vytvoření nové položky do skladu (vpravo).

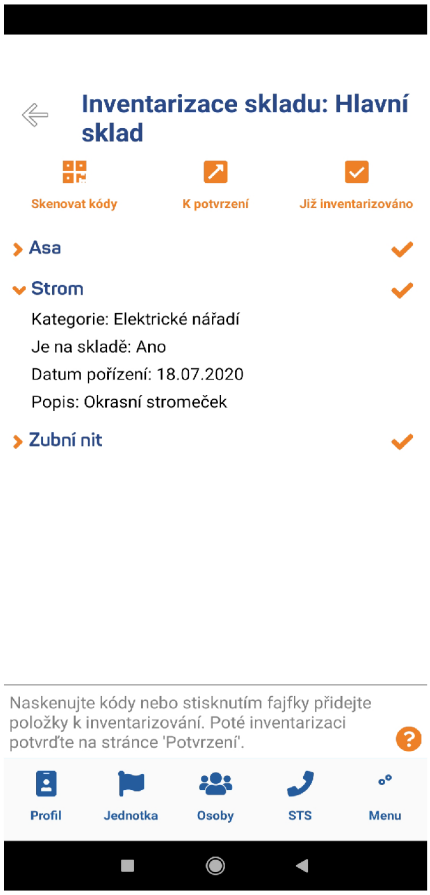
6.4 Inventarizace

Během implementace této části agendy byl změněn původní návrh záložek vzhledem k nesouladu s návrhem zbytku aplikace. Místo toho bylo využito odkazovacích tlačítek na horním panelu. Z výsledků

testování (viz kapitola 7) bylo patrné, že inventarizace je stále poněkud složitějším procesem. Z toho důvodu byla do vybraných stránek inventarizace přidána modální okna (viz obrázek 6.4) s nápovědou, jak inventarizaci provádět.



Obrázek 6.4: Modální okno popisující proces inventarizace.



Obrázek 6.5: Seznam neinventarizovaných položek.

6.4.1 Inventarizace pomocí skenování kódů

Aplikace umožňuje přidávat položky k inventarizaci pomocí skenování QR a čárových kódů. K tomu byl využit zásuvný modul (viz

sekce 2.2) *nativescript-barcodescanner*[3]. Pro jeho použití je potřeba povolit aplikaci přístup k fotoaparátu. Po naskenování kódu dojde k zavření kamery a vyskočí okno oznamující úspěšné/neúspěšné přidání položky k inventuře.

6.4.2 Offline inventarizace s využitím databáze SQLite

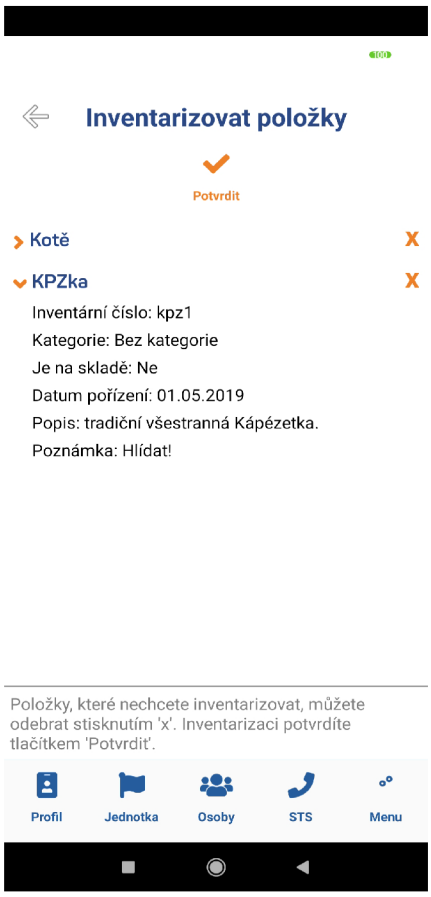
Jedním z funkčních požadavků uvedených v podsekci 5.1.1 bylo poskytnutí možnosti provádění inventarizace bez připojení k internetu. Nejpřívětivějším řešením bylo použití databáze. Do projektu bylo nainstalováno rozšíření *nativescript-sqlite*[4]. Jedná se o relační databázi, která ukládá data do tabulek lokálně v mobilním zařízení. Jak název napovídá, pro práci s databází se používá jazyk SQL. Při implementaci databáze bylo převážně čerpáno z článku [7].

Při vstupu do sekce inventarizace se aktualizují informace o inventarizacích jednotky do databáze. Následně, po výběru skladu, je na server odeslán požadavek, který zjistí, zda ve skladu probíhá aktivní inventura. Pokud ano, ze serveru se do databáze uloží položky daného skladu a jeho základní informace. Následující stránky již fungují bez nutnosti připojovat se na server a číst z něj data.

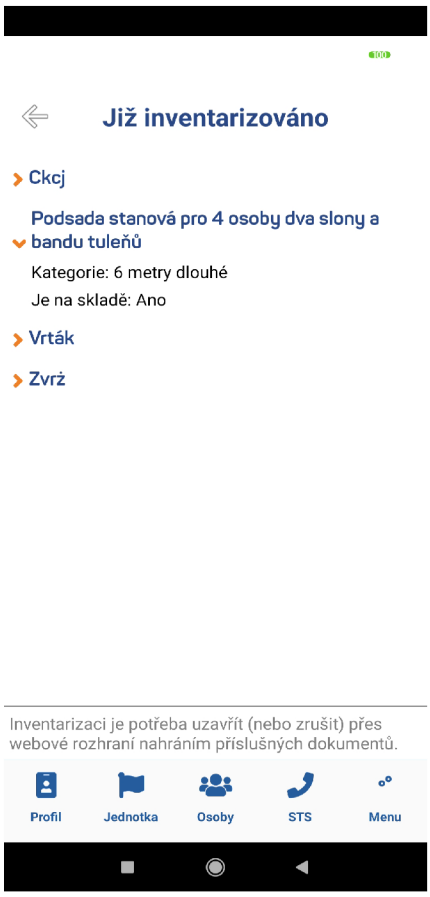
Při proklikávání mezi jednotlivými stránkami inventarizace se načtou vždy všechny nevyřazené položky skladu a jejich filtrování probíhá pouze při zobrazení na stránce. Je to z toho důvodu, aby při přidání/odebrání položky k inventarizování nemuselo vždy po zápisu docházet i ke čtení z databáze. Pro rozeznání, jestli je položka neinventarizovaná, připravená k inventarizování či inventarizovaná je použito dvou proměnných – číslo *idLastInventory* a příznak *localChange*. Položkám, které již byly inventarizovány dříve, je nastaveno *idLastInventory* na **id** aktivní inventury. Při přidání položky k inventarizování je nastaveno *localChange* na TRUE. Po potvrzení inventarizace je příznak nastaven zpět na FALSE a *idLastInventory* je nastaveno na **id** dané inventury. Jednotlivé stránky inventarizace zachycují obrázky 6.5, 6.6 a 6.7.

Pokud se uživatel rozhodne odejít ze sekce inventarizace a nebude připojen k internetu, objeví se dialog oznamující tuto skutečnost. Je to

3. `https://market.nativescript.org/plugins/nativescript-barcodescanner`
4. `https://market.nativescript.org/plugins/nativescript-sqlite`



Obrázek 6.6: Položky čekající na synchronizaci se serverem.

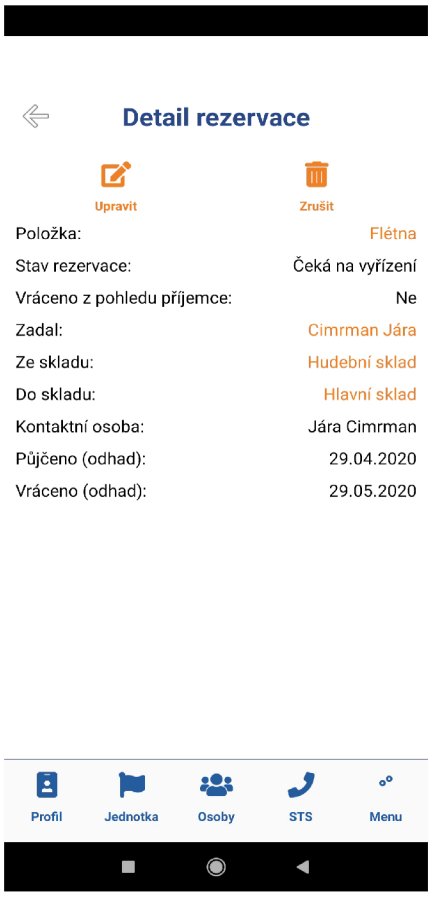


Obrázek 6.7: Seznam inventarizovaných položek.

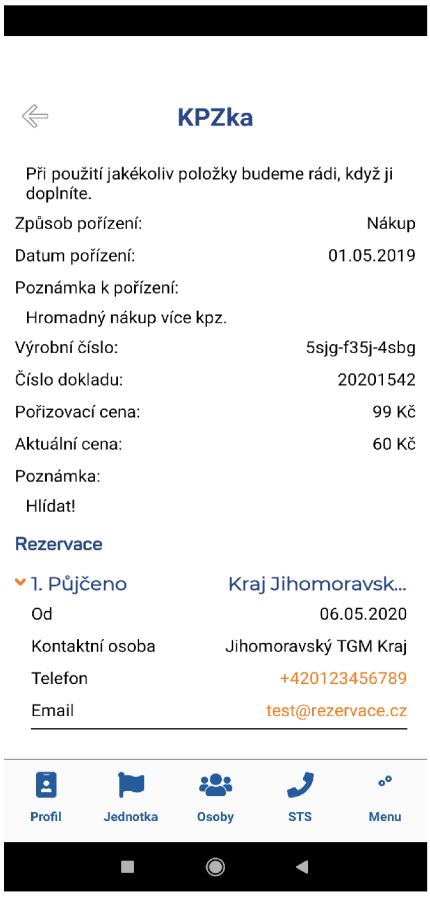
z důvodu nemožnosti pokračovat v offline inventarizaci, jelikož aplikace bude vyžadovat stažení aktuálních dat ze serveru k připravení lokální databáze. Pro potvrzení inventarizace aplikace opět zkontroluje, zda je uživatel připojený k internetu. Pokud ano, odešle data na server. Pokud ne, oznámí uživateli, ať akci opakuje, až bude připojení k internetu dostupné.

6.5 Rezervované, vydané a půjčené položky

Tato sekce byla rozdělena do dvou samostatných stránek na základě zachování jednotného vzhledu aplikace, který příliš nepodporuje přepínací záložky. V rozcestníku tak vznikly dva odkazy, **Přehled rezervací/výpůjček** a **Vydané položky**. Obě stránky obsahují pro lepší pochopení modální okna s nápovědou.



Obrázek 6.8: Detail čekající rezervace ze strany příjemce.

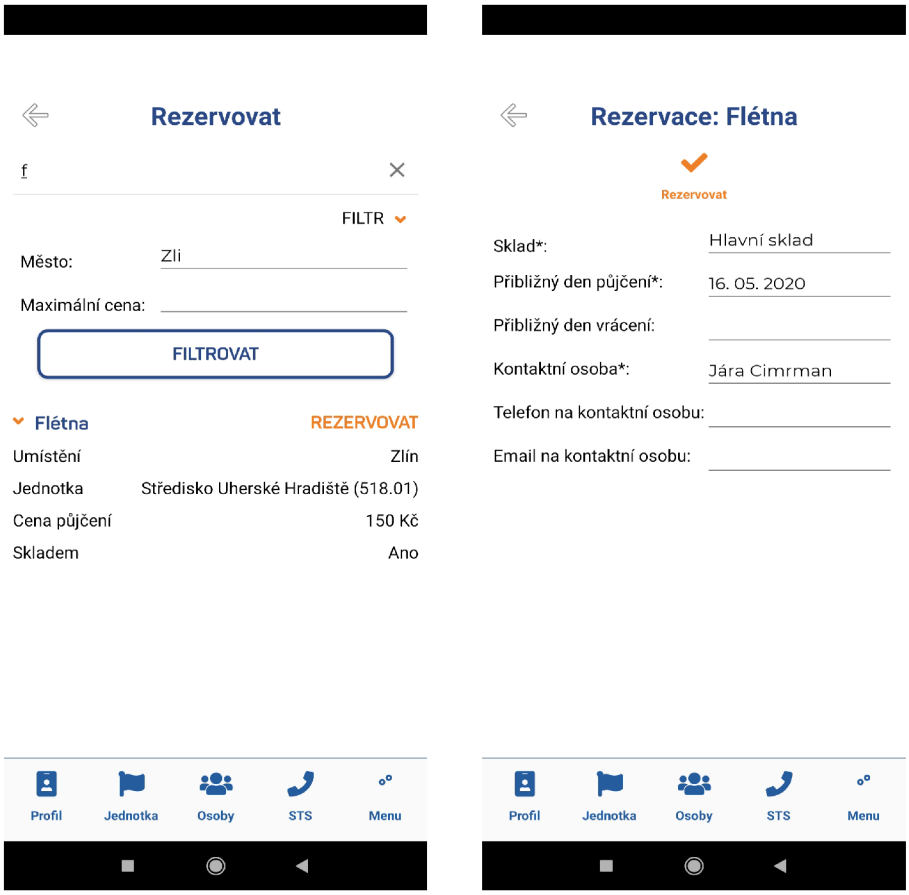


Obrázek 6.9: Detail vydané položky ze strany pronajímatele.

První stránka je určena pro položky, které si uživatel chce vypůjčit do jeho skladu. Původně byly seřazeny podle stavu rezervace, tzn. nahoře se zobrazovaly dosud nevyřízené. Na základě výsledků z testování byly přidány další možnosti řazení, a to podle názvu a podle data.

Detail čekající rezervace s možností její úpravy či zrušení je zobrazen na obrázku 6.8.

Druhá stránka obsahuje seznam položek, které uživatel vydal ze svého skladu. Stejně jako u rezervací, i zde byla po testování doplněna možnost řazení podle názvu položky a data. Řazení podle stavu není nutné, jelikož se jedná pouze o vypůjčené položky. Po kliknutí na danou položku dojde k přesměrování do jejího detailu, kde je k nalezení detail rezervace (obrázek 6.9).



Obrázek 6.10: Filtrování dostupných položek k rezervování (vlevo) a vytvoření nové rezervace(vpravo).

6.6 Provádění rezervace

Jak bylo řečeno v části 3.4.1, rezervace majetku je dostupná pouze uživatelům v roli skladníka nebo vedoucího/admina. Z toho důvodu nebylo nutné tuto funkcionalitu zpřístupnit všem členům a proto mohla zůstat v agendě majetku.

Stránka s nabízeným vybavením obsahuje filtr vyhledávání. Jeho použití funguje lokálně v rámci aplikace, aby nemuselo po každém novém hledání docházet ke stahování dat ze serveru, což je výhodné zejména při připojení k internetu skrze mobilní data. Vyhledávání položek s použitím filtru a vytváření nové rezervace je zobrazeno na obrázku 6.10.

Pro rezervaci majetku je jedním z povinných polí uvedení skladu, do kterého bude položka vypůjčena. V případě, že uživatelova jednotka nevlastní žádný sklad, vyskočí po kliknutí na tlačítko REZERVOVAT zpráva oznamující tuto skutečnost.

7 Testování a nasazení aplikace

Aplikace rozšířená o agendu majetku bude nasazena do obchodů App Store a Google Play po dokončení této práce. Z toho důvodu bylo provedeno neformální uživatelské testování agendy majetku pouze v rámci vybrané skupiny testerů na platformě Android. V této kapitole je popsán průběh a výsledky testování, následně jsou shrnuty změny v aplikaci provedené na základě těchto výstupů.

Uživatelského testování se zúčastnili 3 členové Infoodboru Junáka, což je poradní skupina pro informační technologie v Junáku. Pro účely testování byl vytvořen soubor s příponou `.apk`, který je možné nainstalovat na mobilní zařízení bez nutnosti umístit jej do některého z výše zmíněných aplikačních obchodů. Soubor byl vytvořen přes příkazový řádek NativeScriptu příkazem `tns build android --bundle --env.uglify --env.aot`. Příkaz vytváří pouze nepodepsanou verzi souboru, kterou nelze nahrát do obchodu Google Play.

Testování proběhlo formou plnění zadaných úkolů (viz příloha A). Odpovědi byly sbírány pomocí formuláře Google. Výstupy testování byly převážně dvojího typu: chyby v aplikaci a chybějící nebo nedostatečná funkcionalita. Nalezené chyby, jako například validace e-mailové adresy, úprava některých popisků, překlepy apod. byly opraveny.

Požadovanou funkcionalitou, která byla později implementována, byla možnost přidání nové položky, kterou aplikace původně neumožňovala. Dále byla přidána možnost vyhledávání bez diakritiky. Došlo také ke zviditelnění tlačítka pro vydání položky i v případě, že na položku není vystavena žádná rezervace. Jedním z návrhů bylo odlišení nejnovějších rezervací od těch předchozích. Výsledkem je rozšíření možností řazení rezervací, kdy kromě řazení dle stavu přibyla také možnost řazení dle data či názvu.

Kromě výše zmíněných úprav byly v testování požadavky, které nebylo možné splnit. Většinou se jednalo o funkcionalitu, kterou neposkytuje samotný skautIS a jeho webové služby, například nemožnost přejít na podrobnější detaily příchozí rezervace nebo chybějící možnost vrácení položky. Dalšími názory bylo ne příliš vhodné umístění spodního navigačního panelu či šipky zpět v horním navigačním panelu. Tyto změny by však zasahovaly do celkového návrhu aplikace.

Dalším požadavkem bylo vytvoření notifikace v případě, že uživatel zapomene potvrdit inventarizaci. Tato funkce bude implementována při další iteraci vývoje aplikace.

I přes snahu co nejvíce zjednodušit proces inventarizace a zpřehlednit agendu správy rezervací bylo zjištěno, že tyto části jsou stále ne zcela intuitivní. Z toho důvodu byly přidány nápovědy ve formě modálních oken obsahující popis dané sekce.

8 Závěr

Cílem této bakalářské práce bylo seznámit se s možnostmi multiplatformního vývoje mobilních aplikací, a to především v aplikačním rámci NativeScript. Následně byla provedena analýza informačního systému skautIS včetně podrobného seznámení s agendou majetku. Byla nastudována aplikace *Kapesní skautIS*, která se snaží zpřístupnit vybrané agendy skautISu do mobilního zařízení. Praktickým výstupem je rozšíření aplikace *Kapesní skautIS* o agendu správy majetku včetně popisu jejího návrhu a vývoje. Nakonec bylo provedeno uživatelské testování nové agendy na platformě Android.

Agenda majetku poskytuje možnost správy skladů a jejich majetku. Majetek je možné upravovat, ale také vyřazovat či přidávat nový. Aplikace umožňuje provádění inventarizace skladů, a to bez nutnosti připojení k internetu. Inventarizaci je možné provádět formou skenování QR a čárových kódů. Agenda také obsahuje možnost rezervace položek a následnou správu těchto rezervací.

Aplikace rozšířená o novou agendu prošla uživatelským testováním, jehož výstupy přispěly nejen ke zdokonalení a úpravám aplikace, ale také přinesly podněty k možnému rozšíření agendy v budoucnu. Aplikace bude v co nejbližší době nasazena do testovacích kanálů obchodů Google Play a Apple Store.

Mezi náměty na další vývoj patří možnost přidání fotografie v detailu majetku, která bohužel v době vypracování práce nebyla možná z důvodů chybějících webových služeb na straně API systému skautIS. Dalším rozšířením by mohlo být přidání funkcionality pro vracení položek, jakmile to v systému skautIS bude možné. Užitečná by byla možnost provádění inventarizace rovnou v seznamu položek či vytvoření systému notifikací pro oznámení nedokončené inventury. V neposlední řadě by mohlo dojít k přidání možnosti zvolit výchozí sklad nebo k rozšíření možností pro vkládání adresy skladu, a to skrze našeptávání adresy, zadání GPS souřadnic či přímo zaznačením skladu na mapě.

Bibliografie

[1] Raymond Camden. *Getting Your Route On with NativeScript-Vue: Episode One*. 23. čvc 2019. URL: https://www.nativescript.org/blog/getting-your-route-on-with-nativescript-vue-episode-one (cit. 13. 05. 2020).

[2] Jan Crha. „Mobilní aplikace pro správu majetku“. Bakalářská práce. Masarykova univerzita, Fakulta informatiky, Brno, 2018. URL: https://is.muni.cz/auth/th/ibkmp/.

[3] NativeScript Documentation. *Technical Overview – NativeScript Docs*. URL: https://docs.nativescript.org/core-concepts/technical-overview (cit. 26. 04. 2020).

[4] Petr Domkař. „Multiplatformní mobilní aplikace“. Diplomová práce. Masarykova univerzita, Fakulta informatiky, Brno, 2018. URL: https://is.muni.cz/auth/th/u09n4/.

[5] Vue.js Guide. *Form Input Bindings — Vue.js*. URL: https://vuejs.org/v2/guide/forms.html (cit. 20. 05. 2020).

[6] Michal Haták. *Observables jednoduše – co to je a k čemu je to dobré*. 2017. URL: https://www.inizio.cz/blog/observables/ (cit. 18. 05. 2020).

[7] Kumaran. *NativeScript Groceries app using SQLite*. 2018. URL: https://medium.com/@kumarandena/nativescript-groceries-app-using-sqlite-1018df947601 (cit. 10. 05. 2020).

[8] Priyesh Patel. *PWA vs Hybrid App vs Native: Choosing the Right Mobile App*. 4. pros. 2018. URL: https://blog.bitsrc.io/4-ways-to-build-your-mobile-app-make-the-right-choice-efe079c7c817 (cit. 17. 05. 2020).

[9] Skeleton Software s.r.o. *skautIS*. 2020. URL: https://www.skeleton.cz/reference/skautis (cit. 04. 05. 2020).

[10] Pavel Synek. „Mobilní aplikace pro inventarizaci majetku“. Bakalářská práce. Masarykova univerzita, Fakulta informatiky, Brno, 2015. URL: https://is.muni.cz/auth/th/s9o35/.

[11] Michal Šimík. „SkautIS pro iOS“. Bakalářská práce. Masarykova univerzita, Fakulta informatiky, Brno, 2015. URL: https://is.muni.cz/auth/th/o67jr/.

[12] Emil Tabakov. *NativeScript 5.2 Comes with Official Support for Vue.js*. 2019. URL: https://www.nativescript.org/blog/nativescript-5.2-comes-with-official-support-for-vue (cit. 26. 04. 2020).

[13] David Tureček. *Uživatelský účet* [*Nápověda*]. 2019. URL: https://napoveda.skaut.cz/skautis/informacni-system/uzivatel (cit. 04. 05. 2020).

A Elektronické přílohy v IS MU

Archiv bakalářské práce je uložen na stránce `https://is.muni.cz/auth/th/j36qv/`. V archivu závěrečné práce se nachází:

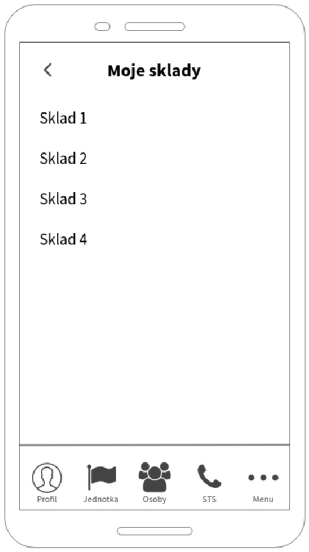
- zdrojový kód aplikace: *kapesni_skautis-bc-Ostruszkova.zip*,
- soubor určený k instalaci testovací verze aplikace pro systém Android: *kapesni-skautIS-with-asset-management-agenda.apk*,
- text práce ve formátu PDF: *text-prace-latex.pdf*,
- zdrojové soubory textu práce psané v programu LaTeX včetně všech použitých obrázků: *text-prace-latex-zdroj.zip*,
- dokument k testování aplikace: *testing.pdf*.

B Návrhy obrazovek agendy majetku

Rozcestník agendy majetku

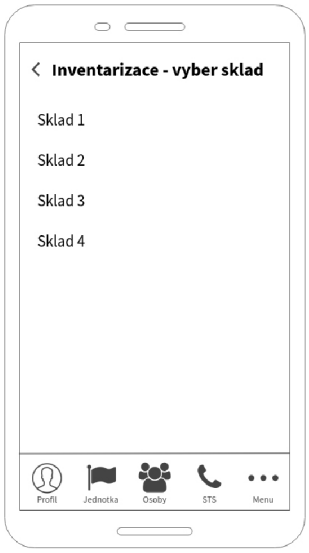
Seznam majetku

Detail položky

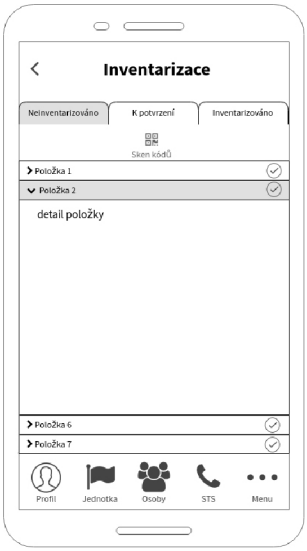


Seznam skladů

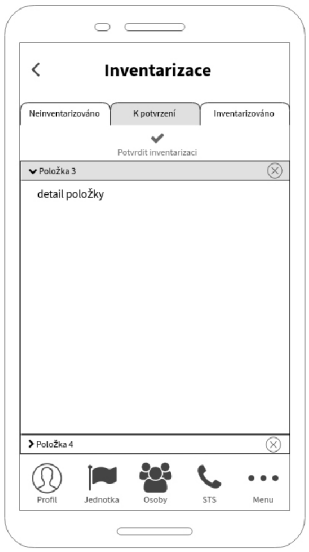
Detail skladu



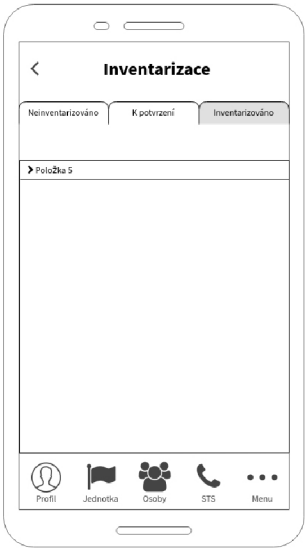
Sklady k inventarizaci



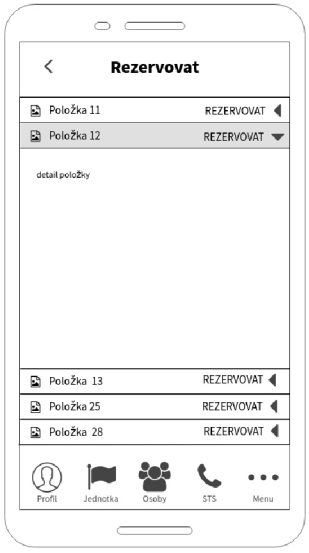
Neinventarizované položky



Položky čekající na potvrzení inventarizace



Inventarizované položky



Rezervovatelný majetek

Vydané položky

Vypůjčené položky