



KIV/MBKZ - Semestrální práce
Správa skladového systému

Jan Kohlíček - A17N0075P

kohl@students.zcu.cz

12. dubna 2018

Obsah

1	Zadání	1
2	Programátorská dokumentace	2
2.1	Grafické rozhraní	2
2.2	Komunikace	3
2.3	Lokalizace	3
3	Uživatelská dokumentace	4
3.1	Přihlášení	4
3.2	Menu	5
3.3	Položky	6
3.4	Tagy	7
3.5	Čtečka	8
3.6	Administrace	9
4	Postup nasazení	12
4.1	Server	12
4.1.1	Instalace	12
4.1.2	Konfigurace	12
4.1.3	Spuštění	13
4.2	Mobilní aplikace	13
5	Testování	14
6	Závěr	15

Kapitola 1

Zadání

Úkolem bude rozšířit mobilní aplikaci pro spravu skladového systému.

První rozšíření se bude vázat ke změně autentizace, aktuálně se používá `HTTP Basic auth` a nahradit by ho měl bezpečnější protokol pro autentizaci `OAuth 2.0`.

Další rozšíření bude umožňovat vyhledávat položky skladu, uživatele a zařízení. Také se musí změnit způsob propojení tagu se skladovou položkou, aktuálně se používá `Spinner`, pro velké množství skladových položek velmi nevhodné.

Poslední rozšíření bude v duchu oprav chyb a drobných úprav pro lepší uživatelskou přívětivost.

Kapitola 2

Programátorská dokumentace

Vývoj mobilní aplikace probíhal nativně v Javě pro Android 6.0. Pro zrychlení vývoje jsem použil knihovnu **Butter Knife**. Její hlavní účel je eliminovat používání metody `findViewById` pomocí **property injection** `@BindView`. `@BindView` do jeho parametru stačí zadat ID požadované komponenty a knihovna **Butter Knife** automaticky inicializuje atribut. Tímto způsobem jsem dosáhl větší přehlednosti v kódu.

Aby aplikace mohla správně fungovat potřebuje tato práva:

- `INTERNET` - umožňuje aplikacím otevřít síťové sokety.
- `ACCESS_NETWORK_STATE` - umožňuje přístup k informacím o síti.
- `NFC` - umožňuje provádět I/O operace přes NFC.

2.1 Grafické rozhraní

Do grafického rozhraní jsem zavedl několik vylepšení. Na procházení velkého množství dat byla použita elegantní technika **infinite scroll**, bez nutnosti čekání se načte další stránka. Princip je jednoduchý, zatímco uživatel skroluje, další obsah je automaticky načítán. Pro rychlé nalezení konkrétních dat, se používá standardní vyhledávání v **Toolbaru**, které zajišťuje komponenta **SearchView**.

Tlačítka jako Uložit, Přihlásit, atd. jsou umístěna v komponentě **Toolbar**, což je z důvodu dlouhých formulářů, aby nemusel uživatel skrolovat. Takto je tlačítko stále k dispozici.

Pro znázornění loadingu byly použity komponenty `ProgressDialog` a `ProgressBar`. `ProgressBar` se zobrazí při čekání na data od serveru. `ProgressDialog` při odesílání dat nejde zrušit kvůli tomu, aby se zabránilo přerušení nebo opětovné operaci. Dialog se zavře, pokud se operace dokončí nebo skončí chybou.

Pro zobrazení dat v přehledech byla použita komponenta `CardView` pro možnost zobrazit velké množství informací o jednom záznamu. Tímto způsobem jsem se vyhnul prokliku na detaily.

2.2 Komunikace

Komunikaci s REST API serveru mi zjednodušuje knihovna `Retrofit`. Stačí vytvořit Java rozhraní se všemi možnými dotazy na server a `Retrofit` si je sám implementuje. Požadavky lze volat jako Java metody a přijatá data formátu JSON se konvertují na JAVA objekty pomocí knihovny `Gson`. Pro editaci HTTP hlavičky a práci se SSL certifikáty jsem použil knihovnu `OkHttp`.

2.3 Lokalizace

Mobilní aplikace je plně přeložená do dvou jazyků češtiny a angličtiny. Jazyk se vybírá automaticky podle nastavení v systému a pokud se nenajde požadovaný jazyk, tak se zvolí angličtina.

Na straně serveru se pracuje s časem bez časových zón. Aby čas nebyl irrelevantní k uživateli, musí se podle časové zóny, kterou má systém `Android` nastavenou, přepočítat čas. Toto přináší výhodu, ať je uživatel odkudkoliv, vždy se mu zobrazí správný čas.

Kapitola 3

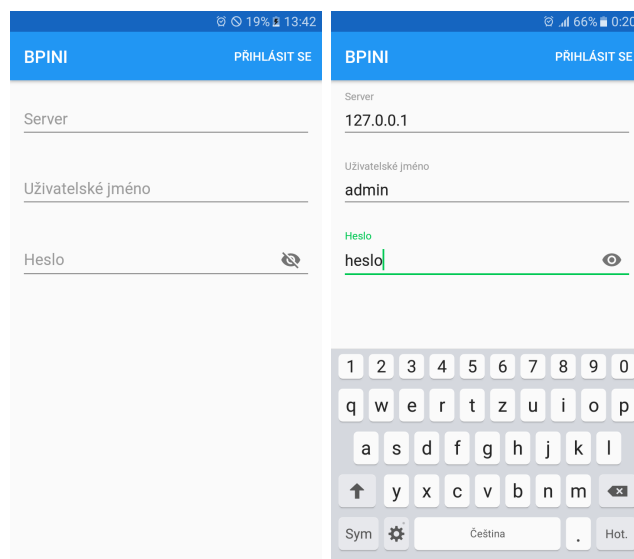
Uživatelská dokumentace

3.1 Přihlášení

Při prvním spuštění aplikace se zobrazí přihlašovací formulář (viz obrázek 3.1a). Zadejte adresu serveru, pak následuje uživatelské jméno a heslo (viz obrázek 3.1b).

Výchozí přihlašovací údaje administrátora systému jsou:

- uživatelské jméno: admin
- heslo: heslo



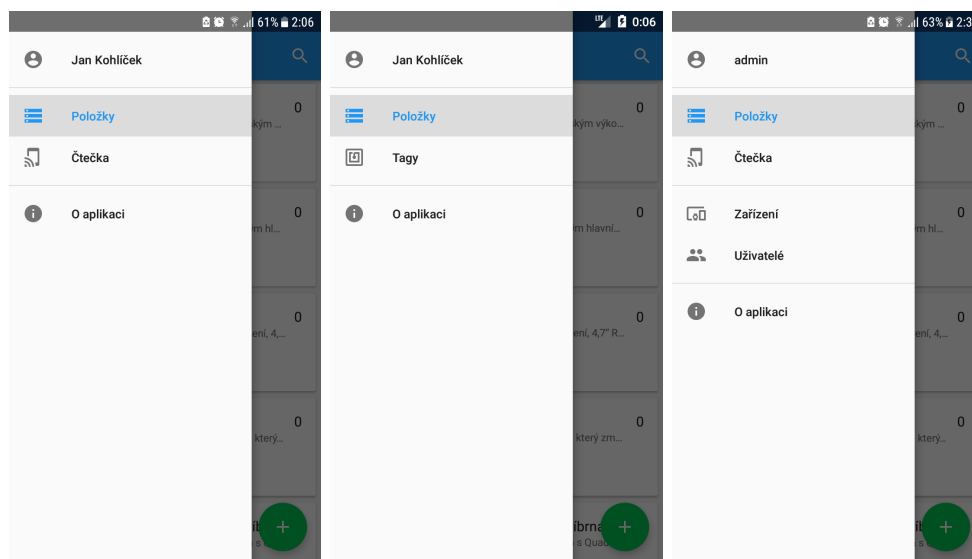
(a) Prázdný

(b) Vyplněný

Obrázek 3.1: Přihlašovací formulář

3.2 Menu

Menu se zobrazí stisknutím hamburger menu nebo vysunutím zpoza okraje (viz obrázky 3.2a, 3.2b a 3.2c). První položka menu otevírá detail přihlášeného uživatele, pak následují „Položky“. Další položkou v menu jsou buď „Tagy“ nebo „Čtečka“, zobrazení záleží na dostupnosti NFC, zařízením s NFC se zobrazí „Čtečka“ ostatním se jinak zobrazí „Tagy“. Poté následují „Zařízení“ a „Uživatelé“, které jsou přístupné jen pro administrátora. Poslední je „O aplikaci“, zobrazí verzi, popis a autora aplikace.

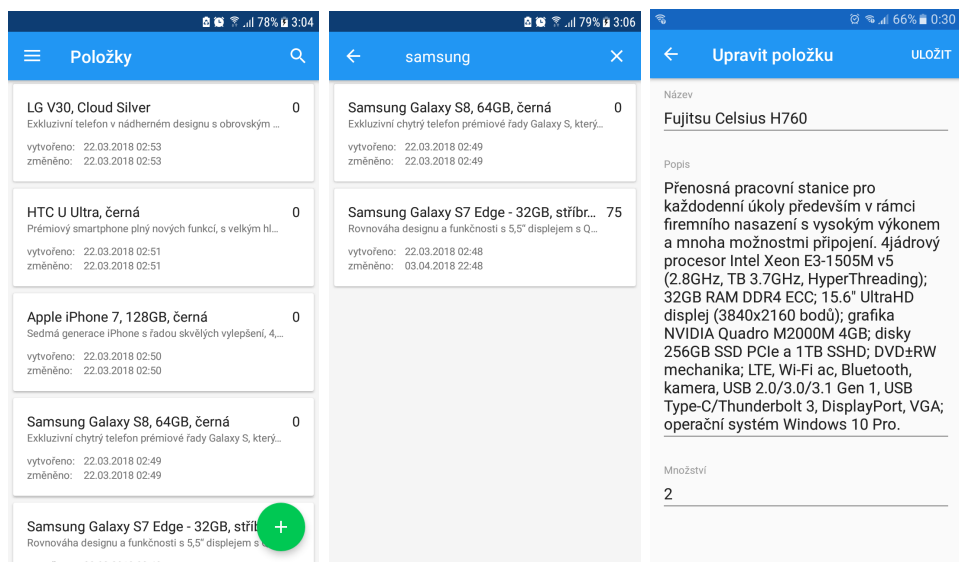


(a) Pro zařízení s NFC (b) Pro zařízení bez NFC (c) Pro administrátora

Obrázek 3.2: Menu aplikace

3.3 Položky

Po spuštění aplikace se zobrazí seznam položek skladu (viz obrázek 3.3a). Položky můžete přidat pomocí zeleného plus v pravém dolním rohu a kliknutím na danou položku editovat. Položku lze smazat jen tehdy, když se její počet rovná nule.



(a) Seznam položek

(b) Vyhledání položky

(c) Úprava položky

Obrázek 3.3: Položky

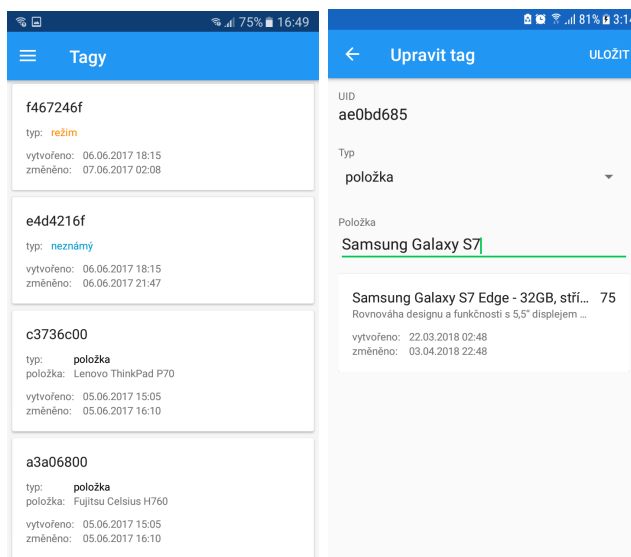
3.4 Tagy

Tagy nelze přidávat ručně, jen pomocí čtečky. Kliknutím na tag se zobrazí editace (viz obrázek ??), která nabízí změnu typu. Tag může být tří typů:

neznámý - tag nemá nastavenou žádnou funkci

režim - tag umožňuje čtečce RFID přepínat režimy přidat/odebrat položku

položka - tag reprezentuje položku ve skladu



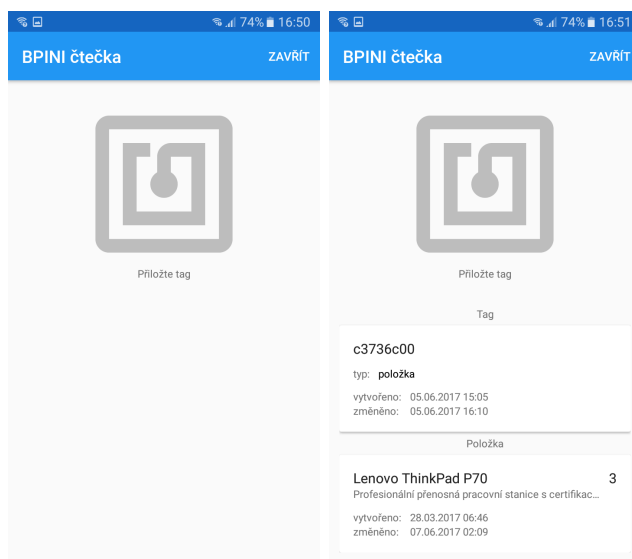
(a) Seznam tagů

(b) Úprava tagu

Obrázek 3.4: Tagy

3.5 Čtečka

Čtečka čekající na přiložení tagu (viz obrázek 3.5a). Po přiložení tagu se načtou detailní informace (viz obrázek 3.5b), ale pokud není v systému zaevidován, pak je vytvořen tag typu „neznámý“.



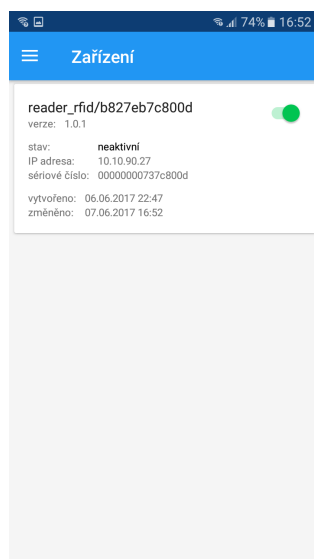
(a) Připravená čtečka

(b) Načtený tag

Obrázek 3.5: Čtečka

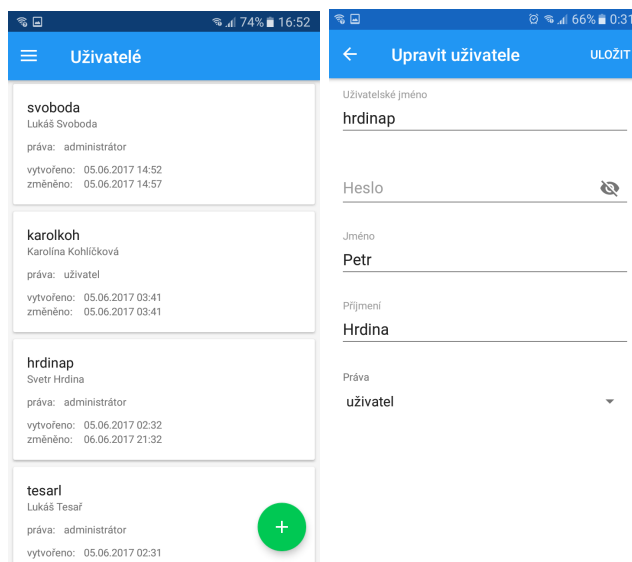
3.6 Administrace

Seznam zařízení se ukazuje jen administrátorům (viz obrázek 3.6). Zařízení se evidují automaticky při jakémkoli pokusu o připojení k serveru. Aby se zařízení připojilo, je nutné, aby konkrétnímu zařízení byl povolen přístup. Ten se mění pomocí přepínače.



Obrázek 3.6: Zařízení

Jen administrátor má přístup ke správě uživatelů (viz obrázek 3.7a). Dostat se na vytvoření nového uživatele je možné přes zelené plus v pravém dolním rohu a kliknutím na uživatele editovat. Uživatele s rolí „administrátor“ může vytvářet a editovat jen výchozí administrátor.



(a) Seznam uživatelů (b) Úprava uživatele

Obrázek 3.7: Uživatelé

Kapitola 4

Postup nasazení

Projekt naleznete ke stažení na:
<https://github.com/kohlicekjan/BPINI>

4.1 Server

4.1.1 Instalace

Pro spuštění serveru je potřeba nainstalovat `Node.js`. Na oficiálních stránkách je k dispozici podrobný postup:

<https://nodejs.org/en/download/package-manager/>

Pro nainstalování potřebných modulů, spusťte ve složce `/src/Server/` tento příkaz:

```
npm install
```

Nainstalujte také databázi `MongoDB`, postup naleznete zde:
<https://docs.mongodb.com/manual/installation/>

4.1.2 Konfigurace

Ve složce projektu `/src/Server/config` jsou konfigurační soubory. V souboru `default.js` zadejte do `uri` adresu spuštěné databáze s názvem databáze, kterou chcete vytvořit. Dále také můžete nastavit adresu a porty serveru. Ukázka konfigurace:

```
{
  host: '127.0.0.1',
  port: {
    http: 80,
    https: 443,
    mqtt: 1883
  },
  mongodb: {
    uri: 'mongodb://127.0.0.1:27017/warehouse'
  }
}
```

4.1.3 Spuštění

Server spustíte tímto příkazem:

```
node server.js
```

4.2 Mobilní aplikace

Aplikace je určena pro Android 6.0 a vyšší. Soubor `BPINI-1.6.0.apk` nahrajte do mobilního zařízení a spusťte instalaci. Při instalaci bude potřeba dočasně povolit instalaci z neznámých zdrojů. Po dokončení najdete aplikaci v menu mezi ostatními aplikacemi.

Kapitola 5

Testování

Aplikace byla testovaná na zařízení Samsung A3 (2017) s Androidem 7.0.0

Kapitola 6

Závěr

V rámci semestrální práce jsem také musel upravit `REST API`, což zkomplikovalo a zdrželo vývoj jednoduchých rozšíření.

Změna protokolu pro autentizaci na `OAuth 2.0` se mi nepovedlo, alespoň jsem přešel na šifrovanou komunikaci, která je dnes standardem.

Vyhledání je plně funkční, jen škoda chybějícího `fulltextu`, to ale budu řešit na straně `REST API`.

Změna propojení mezi položkou skladu a tagem se více měně povedlo. Použil jsem `AutoCompleteTextView` na našepatávač názvu skladové položky, pokud existuje jen jedna schoda zobrazí se detail skladové položky a umožní se uložení. Možný problém může stat u skladových položek se stejným názvem, tento problém zatím nevím, jak budu řešit.

Dále jsem opravil chybu s automatickým vracením zpět při neočekávané chybě od severu. Další chybou bylo možnost spustit „Čtečka“, aniž by byl uživatel přihlášený.

Také jsem přidal automatické odhlášení uživatele, při spuštění aplikace s vyšší vezí než když se přihlašoval. Touhle funčností se předchází chybám z nekonzistence dat mezi verzema.

Mám v planu plno dalších rozšířeních, ale nejdřív musím dodělat podporu na straně `REST API`.