



Mobilní aplikace pro RFID inventarizaci knihovny

Bakalářská práce

Studijní program:

B0613A140005 Informační technologie

Studijní obor:

Aplikovaná informatika

Autor práce:

Kristýna Kohoutová

Vedoucí práce:

Ing. Jan Kolaja, Ph.D.

Ústav nových technologií a aplikované informatiky





Zadání bakalářské práce

Mobilní aplikace pro RFID inventarizaci knihovny

Jméno a příjmení: **Kristýna Kohoutová**
Osobní číslo: M19000023
Studijní program: B0613A140005 Informační technologie
Specializace: Aplikovaná informatika
Zadávací katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: 2021/2022

Zásady pro vypracování:

1. Vytvořte mobilní aplikaci, která bude komunikovat s ruční RFID čtečkou Alien.
2. Vyhodnoťte možnosti ukládání dat v iOS a vyberte vhodný databázový nástroj tak, aby data byla vždy přístupná (i když je zařízení offline).
3. Navrhněte a realizujte databázi knih, tak aby bylo možné knihy ukládat, hledat, třídit, mazat.
4. Aplikace necht' umožňuje vyhledávání zvolené knihy v prostoru (v knihovně) pomocí zjišťování RSSI.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
30-40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] JAN EKHOLM, Mikko Lähteenmäki. Sample NURAPI applications for iOS [online]. Dostupné z: https://github.com/NordicID/nur_sample_ios
- [2] DENNIS HOWARD. Beginners Guide to Programming iOS 14 Using SwiftUI and Xcode: A Simple Step by Step Guide to Everything You need to Know about Coding iOS 14 on Swift 5.2. 2020-11.
- [3] GARY ELMER. Programming iOS 14 Using Swift UI: Get Started With Swift 5 and Xcode. 2020-10.

Vedoucí práce:

Ing. Jan Kolaja, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce:

12. října 2021

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2021

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědoma toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

11. května 2022

Kristýna Kohoutová

Poděkování

Chtěla bych poděkovat panu Ing. Janu Kolajovi Ph.D. za vedení při tvorbě bakalářské práce, za konzultace a za poskytnutí cenných rad a námětů. Dále bych také chtěla poděkovat své rodině.

Abstrakt

Bakalářská práce v úvodu popisuje technologii RFID s jejími klíčovými prvky. Poté se zaměřuje na samotnou aplikaci pro operační systém iOS, jejíž vývoj byl hlavním cílem práce. Je zde popsán proces vývoje aplikace, využívající technologie RFID. Hlavním úkolem aplikace je umožnit uživateli načítat RFID tagy s využitím čtečky ALR-S350-EMA, uložení tagů do databáze a následně hledání tagů v prostoru. Aplikace byla vyvíjena pro hledání knih v knihovně. Do každé knihy je vložen RFID tag. Informace o knize s daným tagem se ukládají do databáze a poté si uživatel zvolí knihu, kterou chce nalézt v knihovně a aplikace jej k dané knize nasměruje. Výhodou využívání technologie RFID je, že tagy dokážou na rozdíl od čárových kódů komunikovat z větší vzdálenosti bez nutnosti přímé viditelnosti. Není tak nutné otevírat knihu pro její identifikaci, což by bylo při použití čárových kódů nezbytné. Aplikace komunikuje za pomoci technologie Bluetooth s ruční UHF RFID čtečkou a pomocí připojení k internetu využívá cloudovou databázi pro ukládání dat.

Klíčová slova:

RFID, ruční UHF RFID čtečka, NurApi, Alien Technology, iOS aplikace

Abstract

The bachelor thesis describes RFID technology with its key elements in the introduction. It then focuses on the iOS application itself, the development of which was the main goal of the thesis. The process of developing an application using RFID technology is described. The main purpose of the application is to allow the user to read RFID tags using the ALR-S350-EMA reader, store the tags in a database and then search for the tags in space. The application was developed for searching books in a library. An RFID tag is embedded in each book. The information about the book with the tag is stored in the database and then the user selects the book he wants to find in the library and the application directs him to the book. The advantage of using RFID technology is that the tags can communicate from a greater distance without the need for direct visibility, unlike barcodes. There is no need to open the book to identify it, which would be necessary when using barcodes. The application communicates using Bluetooth technology with a handheld UHF RFID reader and uses a cloud database for data storage via an internet connection.

Key words:

RFID, handheld UHF RFID reader, NurApi, Alien Technology, iOS application

Obsah

1	Úvod	11
2	Technologie RFID	12
2.1	Obecně o RFID	12
2.2	Existující mobilní RFID aplikace	17
2.3	Alternativní produkt AirTag od společnosti Apple	18
2.3.1	Ekonomický aspekt UHF RFID vs Apple Airtag	18
2.4	Využití RFID pro inventarizaci knih	19
3	Programování pro iOS, Android a knihovna NurApi	21
3.1	Vývoj aplikace pro iOS	21
3.2	Rozdíly v knihovně NurApi pro Android	22
3.3	Knihovna NurApi	22
4	RFID aplikace	24
4.1	Návrh aplikace a zvolená databáze	25
4.1.1	Databáze	25
4.1.2	Propojení aplikace s Firebase	26
4.1.3	Firebase autentizace - Firebase Authentication	27
4.1.4	Firebase databáze - Firebase Firestore	27
4.1.5	Návrh databáze	28
4.1.6	Návrh základní funkcionality aplikace	28
4.1.7	Vyhledávání knihy uložené v databázi	29
4.2	Finální řešení aplikace a poznatky při vývoji	29
4.2.1	Finální aplikace	29
4.2.2	Poznatky při vývoji aplikace	34
4.2.3	Rozšíření vyhledávání i pro QR a čárové kódy	34
4.2.4	Potencionální úpravy pro zveřejnění aplikace	35
4.3	Aplikace v praxi	35
4.3.1	Použití aplikace s RFID čtečkou a tagy	35
4.3.2	Testování	37
4.3.3	Kritické zhodnocení aplikace, možná vylepšení	40
5	Závěr	43
	Použitá literatura	44

Seznam obrázků

2.1	RFID tagy [5]	15
2.2	RFID čtečka	16
2.3	Porovnání systémů označování knih	19
4.1	Princip systému inventarizace knihovny	24
4.2	Dostupné tagy	31
4.3	Informace o tagu	31
4.4	Knihy v databázi	32
4.5	Hledání tagu	32
4.6	Návod k aplikaci	33
4.7	Přidávání žánru	33
4.8	Vyhledávání v prostoru	39

Seznam zkratek

- API - Application Programming Interface
- EIRP - Equivalent Isotropically Radiated Power
- EPC - Electronic Product Code
- HF - High Frequency
- ID - Identification
- iOS - Operační systém společnosti Apple pro mobilní zařízení
- LF - Low Frequency
- REST - Representational State Transfer
- RFID - Radio Frequency Identification
- RF - Radio Frequency
- RSSI - Received Signal Strength Indication
- SHF - Super High Frequency
- SMA - Simple Moving Average
- UHF - Ultra High Frequency

1 Úvod

Technologie RFID se postupně začleňuje do běžného každodenního života. Využita může být v automobilovém průmyslu, při sledování různých druhů zboží ve skladech nebo je možné využít tuto technologii pro lokalizační systémy v reálném čase. Technologie RFID může být použita například i ve veřejných nebo školních knihovnách. Školní knihovna vkládá do svých knih i RFID tagy, tedy nepoužívá pouze čárové kódy, jak tomu bylo dříve. Výhoda této technologie spočívá ve schopnosti RFID tagu vysílat informace čtečce na určitou vzdálenost. Není tak nutné knihy otevírat a skenovat čárové kódy uvnitř knih. Schopnost vysílat informace tedy zrychluje a zefektivňuje práci s objekty opatřenými těmito RFID tagy.

Cílem práce je vytvoření aplikace, která komunikuje s RFID čtečkou za pomoci API metod pro operační systém iOS poskytnutých výrobcem čtečky. Aplikace má zobrazit uživateli seznam knih s RFID tagy, které jsou v dosahu čtečky. Ke každému tagu se ukládají informace o knize, do které je tag vložen. Informace popisující knihu jsou název knihy, jméno a příjmení autora, žánr a případná poznámka ke knize. Základní myšlenkou aplikace je ulehčení hledání zvolené knihy označené RFID tagem v prostoru knihovny. Uživatel musí mít možnost vybrat si, kterou knihu potřebuje nalézt a poté aplikace začne ve spolupráci se čtečkou hledat RFID tag přiřazený knize. Uživateli potom aplikace zobrazuje relativní vzdálenost od knihy podle známé hry „Samá voda, přehořívá a hoří“.

V práci je popsána technologie RFID s jejími klíčovými částmi, tedy mimo jiné UHF RFID čtečka a UHF RFID tag. Je navrženo obecné řešení aplikace společně s požadavky na funkcionalitu. Popsána je realizace aplikace za pomoci obrázků demonstřujících vzhled a fungování aplikace. Ke konci jsou popsány možnosti, jak aplikaci dále vylepšit a také její testování.

2 Technologie RFID

2.1 Obecně o RFID

Zkratka RFID vznikla ze slov Radio Frequency Identification, představuje označení způsobu indentifikace objektů. Pro indentifikaci objektů, jsou nutné dva základní prvky, kterými jsou identifikační štítky (tagy) a čtečka. V závislosti na druhu čtečky jsou použity určité druhy antény a kabelu. Klíčovými prvky jsou tag a čtečka, kde anténa a kabel jsou součástí čtecího zařízení. Důležitý je i software, který zpracovává data přijatá čtečkou od RFID tagu. Software, který tato data zpracovává, je popsán v dalších kapitolách této práce.

RFID technologie je vzdáleně připomíná technologii čárových kódů. Nemá tuto technologii nahrazovat, může sloužit pro její doplnění a řešit některé nedostatky, které technologie čárových kódů má. Nabízí i zcela nové možnosti identifikace objektů, které by s čárovými kódy nebyly možné.

Příklady použití RFID

- Inventarizace zboží a správa materiálů v různých odvětvích
- Biometrické pasy a platební karty
- Sledování zvířat
- Lokalizační systémy v reálném čase
- Sledování knihovních materiálů
- Identifikace sportovců na sportovních akcích

Dělení podle frekvence

RFID komunikaci je možné, dle zdrojů [1], [2], rozdělit dle frekvence následovně:

- Nízká frekvence (LF - Low Frequency)
- Vysoká frekvence (HF - High Frequency)
- Ultra vysoká frekvence (UHF - Ultra High Frequency)
- Super vysoká frekvence (SHF - Super High Frequency)

Dle zdroje [3] mají zmíněné frekvence následující vlastnosti.

Nízká frekvence

Nízká frekvence má dosah kolem 10 cm a pracuje v rozsahu 30 - 300 KHz. Tato frekvence je velmi rozšířená. Obvykle bývá využívána pro značení zvířat, ve zdravotnictví a také v průmyslu. Komunikace mezi tagem a čtečkou je možná z velmi krátké vzdálenosti a tagy fungující na této frekvenci jsou pasivní, nemají tedy žádnou vestavenou baterii. Obvykle lze do paměti tagů uložit jen velmi málo informací. Tagy pracující na nízké frekvenci jsou větší a jejich výroba je nákladnější, v porovnání s tagy pracujícími na dalších frekvencích. Výhodou těchto tagů je možnost čtení i pokud jsou připevněny k objektům obsahujícím vodu, ke zvířecím tkáním, ke kovu atd.

Vysoká frekvence

HF, tedy vysoká frekvence pracuje na pevně dané frekvenci, která je 13.56 MHz. Signál je možné zachytit ze vzdálenosti cca 30 cm. Místo běžného rozsahu hodnot je tedy pevně daná hodnota, která je používána po celém světě. Tagy, které tuto frekvenci využívají mají menší rozměry a není tak nákladné je vyrobit v porovnání s tagy, které pracují na nízké frekvenci. Tuto frekvenci využívají například platební karty.

Ultra vysoká frekvence

Ultra vysoká frekvence operuje na dvou různých hodnotách frekvencí. První frekvence, tedy 433 MHz je využívána pro aktivní tagy. Aktivní tagy mají vestavenou baterii a signál je možné zachytit na desítky metrů. Druhá frekvence je v rozsahu 860-960 MHz. Zmíněná frekvence se spíše využívá pro pasivní a polopasivní tagy. Pasivní tagy jsou schopny komunikovat na vzdálenost jednotek až desítek metrů. Na rozdíl od předešlých frekvencí je možné číst najednou více tagů pracujících na této frekvenci, tedy je ošetřeno, aby nedošlo ke kolizi mezi komunikujícími tagy. Výroba tagů je ve srovnání s tagy nízké frekvence levnější a tagy jsou menší, jsou velmi tenké. Protože voda absorbuje vlny ultra vysoké frekvence, není možné používat tagy způsobem jako u nízkých frekvencí, tedy připevnění k předmětům obsahujícím vodu není možné.

Frekvence jsou rozděleny na tzv. kanály. Regionálně se používají různé počty a frekvence těchto kanálů. V Evropě existuje 15 kanálů a maximální vysílaná energie při komunikaci mezi čtečkou a tagem je 4 W EIRP.

Pro identifikaci objektů se využívá standard EPC (Electronic Product Code).

Super vysoká frekvence

Super vysoká frekvence, také označovaná jako mikrovlnná frekvence, je z velké části podobná Ultra vysoké frekvenci. Frekvence využívané v SHF jsou 2,45 GHz a 5,8GHz. Tagy pracující na Super vysoké frekvenci mohou být pasivní, polopasivní a aktivní. Pasivní tagy ovšem nejsou moc rozšířené a kvůli malé poptávce je jejich výroba nákladná. Dosah se liší v závislosti na typu tagu, pasivní tagy mají podobný dosah

jako tagy z Ultra vysoké frekvence. Tagy jsou schopné komunikovat se čtečkou na vzdálenost od jednotek až po stovky metrů. Zároveň jsou tagy také velmi citlivé na kovy a tekutiny.

Díky velkému dosahu aktivních tagů je možné je využít v oblasti sledování pohybujících se objektů v reálném čase. Polopasivní tagy se využívají například pro výběr dálničních poplatků a pro kontrolu vozidel, které jsou ve velké vzdálenosti.

UHF RFID tag

Rozdělení RFID tagů

- Pasivní tagy - nejsou schopné samy od sebe vysílat. Využívají energii z rádiových vln, které vytváří RFID čtečka. Pokud se objeví v okolí čtečky pasivní tag, energie vysílaná ze čtečky je tagem použita k nabití napájecího kondenzátoru tagu. Poté tag využije tuto zpracovanou energii na odeslání odpovědi dané čtečce.
- Aktivní tagy - mají vlastní baterii a díky tomu jsou schopné komunikovat se čtečkou na větší vzdálenost a je možné do nich uložit větší množství dat. Baterie vydrží být nabitá v průměru kolem 3 let a poté se tag musí většinou vyměnit, protože není možná výměna baterie. V porovnání s pasivními tagy je nevýhodou vyšší cena, nutnost výměny po vybití baterie a v některých případech i rozměry tagu.

Informace o tagu

- EPC (Electronic Product Code) představuje 96 - 480 bitové unikátní číslo.
- Tag ID - jedinečný identifikátor přidělený výrobcem, který není možné měnit.

Běžné informace získatelné při čtení

- RSSI (Received Signal Strength Indication) představuje hodnotu přijímané síly signálu od tagu. Hodnota této síly se většinou udává v milliwattech nebo v logaritmické jednotce dBm. Obvyklý rozsah hodnot je dle zdroje [4] -30 až -85 dBm. (V případě této aplikace se rozsah například pohyboval v rozmezí -50 až -75 dBm.)
- Hodnota frekvence, na které tag operuje. Je možné určit, ve které kategorii frekvence se tag nachází, tedy zda tag operuje na nízké, vysoké, ultra vysoké nebo super vysoké frekvenci.
- RFID kanál udává, do kterého RFID kanálu spadá frekvence, na které daný tag operuje. Například v Evropě existuje 15 RFID kanálů.



Obrázek 2.1: RFID tagy [5]

UHF RFID čtečka

RFID čtečky jsou jednou z klíčových součástí RFID systému. Činnost čteček spočívá v odesílání a přijímání rádiových vln. Rádiové vlny jsou nezbytné pro komunikaci s tagy. UHF RFID čtečky je možné rozdělit podle provedení na fixní a mobilní. Jak z názvů již vyplývá, fixní čtečky jsou umístěné pevně na jednom místě (vstupní brána, ve které se sleduje pohyb materiálu) a mobilní čtečky jsou typicky menší zařízení, která umožňují větší flexibilitu pohybu (například inventarizace zboží v obchodě nebo vyhledávání knih v knihovně).

- Fixní čtečky
 - Integrované čtečky
- Mobilní čtečky

Fixní čtečky

Fixní (pevné) čtečky neobsahují integrovanou anténu. Jsou k dispozici porty pro externí připojení antén. Připojení externích antén umožňuje vysokou flexibilitu a umožňuje zákazníkovi výběr antén dle vlastního uvážení. Portů je většinou více a je tak možné připojit větší počet antén v závislosti na potřebě pokrytí dané oblasti. V závislosti na výběru čtečky je možné připojit od jedné do 32 antén.

V některých případech je u fixních čteček možnost volby kabelu. Kabely se rozdělují podle tří základních vlastností - podle typu konektorů, délky a izolační schopnosti.

Integrované čtečky

Integrované čtečky spadají do podmnožiny pevných, tedy fixních čteček. Tyto čtečky zůstávají, stejně jako pevné čtečky, většinu času na jednom místě. Rozdíl oproti klasickým fixním čtečkám spočívá v tom, že mají již zabudovanou anténu i čtecí komponentu v sobě. Většinou je možné přidat ještě externí anténu, pokud je to potřeba. Přidání další antény vytvoří další čtecí bod, který může zlepšit čtecí dosah.

Mobilní čtečky

Mobilní čtečky jsou většinou menší zařízení ve tvaru, který je uzpůsoben pro uchopení do ruky. Umožňují větší flexibilitu pohybu při komunikaci s RFID tagy. Komunikují s chytrým mobilním zařízením nebo počítačem za použití technologie Bluetooth.



Obrázek 2.2: RFID čtečka

Další část RFID systému - Anténa

Anténa je velmi důležitá část RFID systému, která konvertuje signál získaný od čtečky do rádiových vln, které jsou používány pro komunikaci s RFID tagy. Anténa může být v systému RFID buď integrovaná nebo externí.

Tato zařízení přijímají energii od čtečky, z této energie anténa generuje vysokofrekvenční pole a na tagy v okolí čtečky přenáší vysokofrekvenční signál. Schopnost antény vygenerovat vlny určitým směrem a o určitém výkonu se nazývá „zisk antény“. Čím vyšší zisk anténa má, tím je schopna generovat silnější a dále dosahující pole rádiových vln.

Antény se dělí dle několika základních vlastností. Zde je uvedeno základní dělení:

Rozdělení podle polarizace

- anténa s kruhovou polarizací
- lineární anténa

RFID anténa vytváří vlny ve vodorovné nebo svislé rovině, což je popsáno jako polarizace antény. Polarizace antény může mít velký dopad na čtecí dosah systému. Pro maximální dosah čtení je potřebné, aby se shodovala polarizace antény s polarizací RFID tagu.

Rozdělení podle odolnosti

- vnitřní anténa
- venkovní anténa

Další rozdělení antény

Antény se kromě polarity a odolnosti dělí na základě dalších parametrů, například podle rozsahu frekvence, na které pracují, podle rozsahu čtení a nebo podle typu montáže (panelová anténa, portálová anténa, ...). Dle frekvence nejsou v tomto seznamu antény děleny, protože je předpokládáno použití antén jen pro UHF pásmo.

2.2 Existující mobilní RFID aplikace

Aplikací týkajících se RFID technologie existuje již velmi mnoho, jak pro operační systém iOS tak i pro operační systém Android. Existující aplikace nabízejí mnoho funkcí. Je možné s jejich pomocí vyhledávat dostupné tagy v okolí čtečky nebo filtrovat jen ty tagy, které mají silný signál a jsou tedy v bezprostřední blízkosti čtečky.

Přestože je nabídka RFID aplikací široká, pro konkrétní produkt ALR-S350-EMA, který byl k dispozici pro tuto práci se nepodařilo najít žádnou aplikaci přes distribuční službu pro iOS AppStore .

Pro operační systém Android je možné získat v obchodě aplikací (Google Play) demo verzi aplikace pro komunikaci s Alien čtečkou. Zmíněná demo verze je schopna zobrazovat dostupné tagy v okolí čtečky a hledat vybraný tag. Aplikace také disponuje dalšími užitečnými funkcemi týkajícími se například nastavení výkonu čtečky. Pro operační systém iOS je také k dispozici demo verze aplikace pro komunikaci s Alien čtečkou, která ovšem nabízí mnohem méně funkcionalit ve srovnání s verzí pro Android. Nabízí pouze ukázkou spojení se čtečkou a načtení RFID tagu. Problém činí i nedostatečná dokumentace k používání knihovny určené pro komunikaci se čtečkou. Komplikace s tímto spojením budou popsány později v dalších kapitolách.

2.3 Alternativní produkt AirTag od společnosti Apple

Zajímavostí je produkt nazvaný AirTag od společnosti Apple uvedený na trh v roce 2021. Jedná se o kompletní RFID řešení pro identifikaci objektů.

K tomuto produktu vytvořila také aplikaci, která se minimálně částí své funkcionality podobá aplikaci vytvářené v rámci této práce. Umožňuje také vyhledávání tagu stylem samá voda, přihořívá a hoří. Zmíněný tag se v některých oblastech odlišuje od tagů, se kterými komunikuje vyvíjená aplikace. Například zmíněné tagy mají vlastní baterii, jsou větší, a ke komunikaci není potřebná čtečka ale jen mobilní telefon. I přes zmíněné odlišnosti je podstata fungování podobná. Mobilní zařízení s daným tagem komunikuje s použitím technologie Bluetooth. Aplikace pomáhá uživateli nacházet různé objekty ke kterým je daný tag připnutý. Při lokalizaci ztraceného tagu se využívají některé další produkty, které tato firma nabízí tedy iPhone, Macbook a iPad.

2.3.1 Ekonomický aspekt UHF RFID vs Apple Airtag

Tato kapitola porovnává 2 možnosti sestavení inventarizačního systému z ekonomického hlediska s ohledem na pořizovací cenu základního vybavení a následně jeho provoz (počet inventarizovaných knih). Prvky, které jsou nutné pořídit pro funkční systém vyhledávání knih jsou následující.

Systém využívající Apple Airtag

- Airtag - nejlevněji za 583,- Kč
- baterie k výměně - cca 70,- Kč
- Apple iPhone 11 - cca 14.000,- Kč

Baterie u Apple Airtag vydrží nabitá cca 12 měsíců a po této době je nutné baterii vyměnit. Tento aspekt ale není v následujícím porovnání zohledněn. Pro komunikaci je potřebné mobilní zařízení. Nejlevnější model, který komunikaci s tagem podporuje je Apple iPhone 11.

Systém využívající UHF RFID

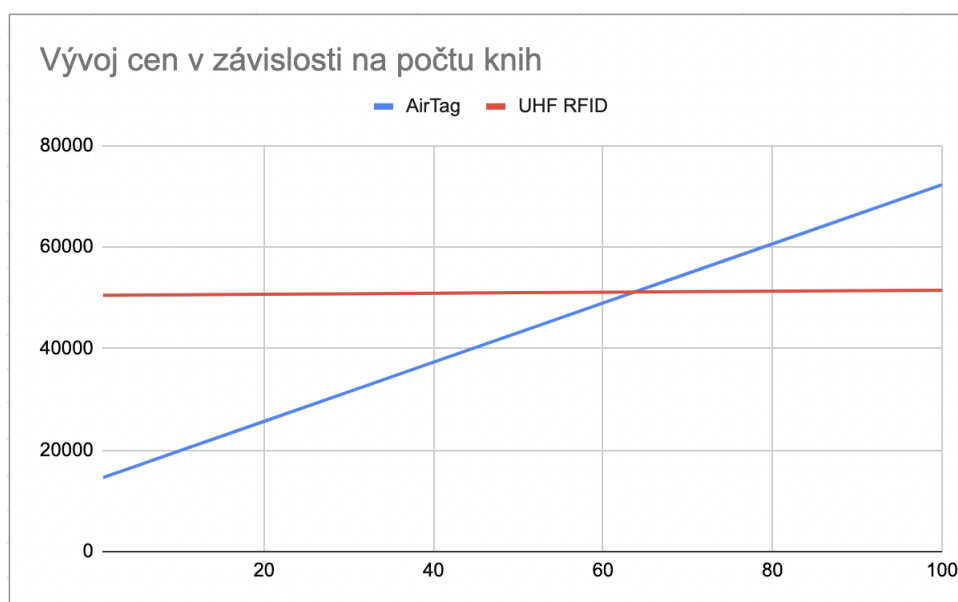
- UHF RFID čtečka - cca 40.000,- Kč
- UHF RFID tag - cca 10,- Kč
- Apple iPhone Xr (nyní nejlevnější iPhone) - 10.500,- Kč

Pro vyhledávání knih v celé knihovně by bylo nutné zakoupit stovky tagů. Systém využívající Apple Airtag je za předpokladu, že se využívají nové nabitě Airtagy, levnější pouze do počtu 63 označených knih. Řešení s využitím UHF RFID je tedy cenově výhodnější v případě větší knihovny.

V tabulce 2.1 jsou vypsané příslušné částky na základě počtu označených knih v závislosti na zvoleném systému. Na obrázku 2.3 jsou tato data zobrazena graficky.

Tabulka 2.1: Počty označených knih

počet knih	cena Airtag systému	cena UHF RFID systému
1	14583	50510
10	19830	50600
50	43150	51000
64	51312	51140



Obrázek 2.3: Porovnání systémů označování knih

2.4 Využití RFID pro inventarizaci knih

Byl proveden průzkum aktuálního využití technologie RFID v praxi, tedy v knihkupectví (jen namátkově na území města Liberec) a ve školní knihovně TUL a došlo se k následujícím závěrům.

Knihy v knihkupectví

Po návštěvě knihkupectví byly zjištěny následující informace. Každá kniha má vlastní čárový kód, některé knihy mají kód vytištěný přímo na svém obalu, některé knihy

mají čárový kód ve formě samolepky. RFID tagy, podobné těm, které jsou využívány v této práci u žádné knihy nebyly nalezeny. U knih bývají často bezpečnostní etikety ať už RF (radiofrekvenční), AM (akustomagnetické) nebo EM (elektromagnetické), které slouží k zabezpečení a ochraně knih a jiného zboží. Tyto zmíněné etikety jsou asi jedinou technologií, která je schopna „aktivně komunikovat“ (na rozdíl od čárových kódů) například s detekčními anténami.

Knihy v knihovně

V univerzitní knihovně má každá kniha přiřazený svůj čárový kód a RFID tag. K identifikaci každé knihy se používá jak čárový kód, tak i RFID tag. Tedy RFID technologie nyní ještě nenahrazuje technologii čárových kódů, ale je pouze součástí identifikace dané knihy. U východu z knihovny jsou umístěny detekční antény, které pravděpodobně komunikují s RFID tagy a plní funkci zabezpečení proti zcizení knihy.

Logickým krokem, který lze v budoucnu očekávat je, že se technologie RFID rozšíří postupem času do mnoha odvětví a pravděpodobně tuto technologii začnou využívat i zmíněná knihkupectví.

Například obchodní řetězec Decathlon již technologii RFID využívá. RFID tagy jsou připevněny na zboží a poté se u pokladny pomocí čtečky načtou, což zjednodušuje práci obsluhujícímu personálu.

3 Programování pro iOS, Android a knihovna NurApi

3.1 Vývoj aplikace pro iOS

Aplikaci pro operační systém iOS je možné vyvíjet ve vývojovém prostředí Xcode. Pro účely vytvoření aplikace, která je předmětem této práce, byla použita verze 13.3. Zvoleným programovacím jazykem je jazyk Swift, konkrétně ve verzi 5. Pro samotné programování a testování aplikace je nutné použít počítač, jehož operačním systémem je macOS. Pro testování aplikace je nutné použít fyzické zařízení, které bude komunikovat se čtečkou pomocí technologie Bluetooth.

Prostředí Xcode

Vývojové prostředí Xcode je aplikace vytvořená společností Apple. Aplikace je pro majitele zařízení s macOS zdarma a je tedy volně ke stažení v AppStore. Nejnovější verze je označena jako verze 13.3 a tato aplikace v počítači zabírá 12.7 GB paměti.

Programovací jazyk Swift

Tento programovací jazyk byl vytvořen jako alternativa k jazyku označovanému jako „Objective-C“. Na rozdíl od objektového C je Swift bezpečnější. Swift například vyžaduje inicializaci proměnných před jejich použitím, automaticky spravuje paměť, kontroluje, aby nedošlo k přetečení velikosti polí a mnoho dalšího.

Vytváření a zveřejnění aplikace

Aplikaci je možné vytvořit přímo v Xcode editoru jako prázdný, nový projekt. Je také možné využít možnosti naklonovat již předpřipravený projekt například z GitHubu. Tato druhá možnost může být nápomocná při zrychlení vývoje aplikace, například mohou být v aplikaci již předpřipravená okna nebo již vytvořené logo aplikace a vývojář se může soustředit na důležitější části vývoje.

Aby bylo možné aplikaci nainstalovat i na fyzické mobilní zařízení, tedy nejen spustit v simulátoru na počítači, je nutné mít vytvořené tzv. Apple ID. Vytvoření Apple ID je zdarma. Poté je nutné nastavit na fyzickém mobilním zařízení povolení pro vývoj aplikací. Každý projekt musí mít svůj tzv. jedinečný svazkový identifikátor

(Bundle Identifier) pro případ zveřejnění aplikace v App Store. Pro zveřejnění vytvořené aplikace a následné zveřejňování nových aktualizací je nutné zaplatit každý rok Apple vývojářskou licenci v ceně 99 USD.

Nahrání aplikace na mobilní zařízení a následné spuštění

Aplikaci je během vývoje možné spouštět na simulátoru zařízení. Ve chvíli, kdy je aplikace připravena na testování na fyzickém zařízení, je nutné mobilní zařízení připojit k počítači, zvolit v Xcode spuštění aplikace na daném zařízení a nechat aplikaci sestavit a poté spustit. Po nějaké době se aplikace sestaví a na mobilním zařízení se objeví její ikona. Pokud se aplikace spouští za pomoci Xcode, tedy mobilní zařízení je připojeno k počítači, aplikace se spustí sama. V tomto módu je možné využít logování do ladícího okna. Zmíněné okno se zobrazuje přímo v Xcode a je možné nechat si vypsat různé akce, které se v aplikaci udály.

Nahraná aplikace i po odpojení od počítače na mobilním zařízení stále zůstává. Je možné aplikaci spouštět i přímo z mobilního zařízení, tedy i bez připojení k počítači. Pokud je aplikace spouštěna přímo z mobilního zařízení není možné využít již zmiňované vypisování logování do ladícího okna v Xcode.

Další poznatky

Aplikace na mobilním zařízení zůstane spustitelná několik týdnů, poté je nutné aplikaci znovu nahrát z počítače. Bez privilegovaných práv která se získají již zmíněnou Apple vývojářskou licenci, je možné na mobilním zařízení testovat maximálně 4 aplikace najednou. Je tedy možné nahrát 4 aplikace bez toho, aby vývojové prostředí Xcode hlásilo, že je nutné některou z nainstalovaných aplikací odinstalovat.

3.2 Rozdíly v knihovně NurApi pro Android

Název knihoven pro operační systém Android je NurApi a NurApiAndroid. Pro operační systém iOS se tato knihovna nazývá NurApiBluetooth.

Knihovny NurApi a NurApiAndroid se přidávají do Android aplikace jako moduly, přímo do projektu. V případě iOS aplikace se NurApi se používají tzv. Pody, což je obdoba modulů u Android aplikace. Poté, co jsou moduly, popřípadě Pody přidány do projektu, je práce při vývoji aplikace s NurApi velmi podobná. Jak již bylo zmíněno, knihovna pro operační systém Android disponuje širší škálou funkcí a díky tomu je mnohem snazší samotný vývoj aplikace.

3.3 Knihovna NurApi

Výrobci čtečky, která je použita při vyvíjení a při následném testování aplikace, poskytli k dispozici GitHub repozitář [6]. Zmíněný repozitář obsahuje jeden ukázkový příklad aplikace pro operační systém iOS. Pro komunikaci se čtečkou existuje obecně

nazvaná knihovna NurAPI, pro operační systém iOS je to konkrétně knihovna s názvem NurApiBluetooth. Knihovna je částečně popsána na zmíněném GitHubu.

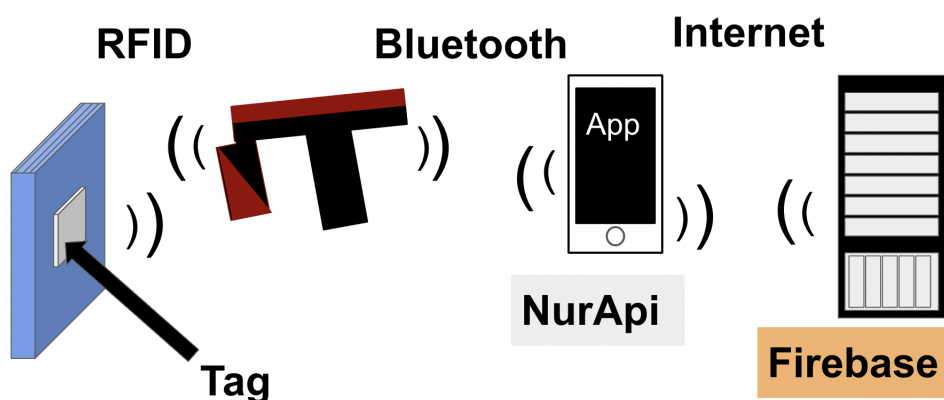
Získávaná data od čtečky

Čtečka posílá aplikaci následující informace které se týkají RFID tagu:

- EPC - unikátní číslo používané pro označení a následnou identifikaci tagu, konkrétně čtečka aplikaci posílá řetězec, o 24 hexadecimálních znacích
- Channel - udává do kterého kanálu spadá frekvence, na které tag operuje
- RSSI - hodnota přijímané síly signálu který získá čtečka od tagu
- Scaled RSSI - měřítko hodnoty RSSI
- Timestamp - čas v milisekundách od doby, kdy byla vytvořena instance NurApi
- Frequency - frekvence, na které RFID tag operuje
- Antenna ID - ID antény - v tomto případě má čtečka jen jednu anténu

4 RFID aplikace

Aplikace komunikuje pomocí Bluetooth s UHF RFID čtečkou, která pomocí technologie RFID komunikuje s UHF RFID tagem. Čtečka aplikaci posílá data o tagu a ta je následně zpracovává. Pro zpracování dat je využita knihovna od výrobců čtečky, tedy knihovna NurApiBluetooth. Zpracovaná data se následně v aplikaci zobrazují uživateli. Uživatel má možnost uložit informace o knize do cloudové databáze Google Firebase, se kterou aplikace komunikuje prostřednictvím Internetu.



Obrázek 4.1: Princip systému inventarizace knihovny

Spojení se čtečkou

Čtečka s mobilním zařízením, respektive s aplikací komunikuje za použití technologie Bluetooth. Aplikace pro zpracování informací od čtečky využívá knihovnu NurApiBluetooth.

V prvním kroku je potřebné získat spojení se čtečkou. K ověření, zda je čtečka dostupná se využije metoda objektu Bluetooth nazvaná „sharedInstance”, která vrátí singleton instanci tohoto objektu. Protože se toto ověřování dostupnosti čtečky volá v úvodním okně aplikace, není ještě vytvořen nurApiHandle. Při zavolání metody „sharedIntance” se nurApiHandle inicializuje. Pokud je tedy nějaká čtečka dostupná, je možné ze singleton instance získat například její název.

Dalším krokem je získávání dat od spárované čtečky. Informace/oznámení od čtečky zpracovává funkce nazvaná „notificationRecieved”, jejíž argumentem jsou

mimo jiné i data získaná od čtečky. V případě, že byla přijata data reprezentující informace o tagu, jsou předána dalším funkcím, které informace zpracují, uloží nebo předají dál. Pokud se nejedná o data s informacemi o tagu, jsou jen vypsaná do konzole a vývojář je vidí v ladícím okně.

4.1 Návrh aplikace a zvolená databáze

4.1.1 Databáze

Základní informace o Firebase

Firebase je platforma patřící pod společnost Google od roku 2014. Dlouhou dobu byla využívána pro jejich hlavní produkt, kterým byla tzv. „real-time“ databáze. Nyní již nabízí více produktů, mimo jiné nástroj pro autentizaci, hostování a Firestore databázi. Narozdíl od některých jiných platform Firestore není open-source, nedá se tedy přistupovat přes GraphQL nebo REST API.

Proč Firebase?

Tato platforma poskytuje ukládání dat na cloud. Firebase také nabízí mnoho nástrojů pro práci s daty. Je možné využívat Firebase pro ukládání dat z aplikace, která běží jak na operačním systému iOS tak i na operačním systému Android. Neměl by být problém propojit již vytvořenou databázi, která komunikuje s iOS aplikací i s aplikací vytvořenou pro operační systém Android.

Umožňuje i ukládání dat v případě, že mobilní zařízení nemá přístup k Internetu. Firebase tedy poskytuje prostředky pro ukládání dat lokálně nebo pro předejití problémů s vysokou latencí. Dle [7] je možné aplikaci používat i pokud se uživatel dostane mimo dosah Internetu. Data se uloží lokálně a později, když je internetové připojení znovu přístupné, data se zaktualizují v Cloud Firestore.

Založení databáze ve Firebase je velmi snadné. Tato platforma má v porovnání s některými jinými platformami velmi dobře napsanou dokumentaci. Na webových stránkách Firebase jsou i na některá témata natočená videa, kde je dané téma velmi dobře vysvětlené.

Alternativy Firebase

Pro porovnání s Firebase byly vybrány následující dvě alternativy. Dle zdroje [8] mají tyto výhody:

Back4App je řešení, které nabízí relační databázi, ke které je možné přistupovat přes GraphQL nebo REST API. Poskytuje pomoc v „hostování“ IoT, webových aplikací a pomáhá urychlit proces vývoje. Toto řešení je open-source, což umožňuje vývojářům přidávat bez problémů nové funkcionality.

Parse nabízí snadné použití ve srovnání s některými dalšími platformami. Je to technologie s open-source kódem, která umožňuje vývojářům spravovat efektivně různé backend služby.

Hlavním rozdílem je tedy, že platforma Firebase není open-source a nelze k ní přistupovat přes GraphQL nebo REST API. Pro účely této práce ale nejsou tyto vlastnosti překážkou. Umožňuje velmi jednoduché a přímočaré připojení a má velmi dobrou dokumentaci. Vytvářená aplikace není velkých rozměrů a není tedy ani potřebné přidávat nové funkcionality, Firebase je tedy dobrou volbou.

Co se stane po měsíci od založení databáze? Firebase nabízí dva módy při vytváření databáze. První možnost je začít používat databázi v produkčním módu, tedy data budou zabezpečená a čtení spolu se zápisem ze strany klienta bude umožněno pouze za určitých podmínek. Druhým módem je testovací mód, kdy je umožněno číst a zapisovat komukoli. Zmíněný mód je tedy nezabezpečený a je nutné změnit ihned po testování pravidla přístupu a zabezpečit tak uložená data.

Po měsíci od registrace, respektive od založení databáze se pravidla přístupu změní. Tedy pokud byla databáze spravována v testovacím módu, změni se testovací mód na produkční mód.

Pokud vývojář aplikace nezmění pravidla přístupu k databázi, tak po měsíci od založení databáze obdrží email od Firebase. V emailu se uživateli sděluje, že se pravidla pro přístup nastaví na stav, kdy k databázi nemůže přistoupit nikdo. Pokud ale vývojář aplikace změni pravidla, na ta, která se pro účel dané aplikace hodí, je vše opět v pořádku a databázi je možné dál používat. Tato změna pravidel zabraňuje tomu, aby uživatel nevědomě nechal povolený přístup všem. Následně by se mohlo stát, že někdo nepovolaný přepíše databázi. Způsobů jak zabezpečit data je více. Nejčastějším způsobem je použití autentizace přihlášeného uživatele. V případě této aplikace se využívá autentizace uživatele. Je možné použít i jiný způsob ochrany dat, například lze kontrolovat typ dat přicházejících do databáze.

4.1.2 Propojení aplikace s Firebase

Základem je mít založený projekt, respektive aplikaci. Pro instalaci databáze Firebase je nutné použít správce závislostí na úrovni aplikace. V tomto projektu je použit správce CocoaPods, s jeho pomocí je možné později inicializovat tzv. Pod příslušící Firebase platformě.

Druhým krokem je založení projektu ve Firebase webovém rozhraní. Po založení projektu je nutné v založeném projektu zaregistrovat aplikaci. Při registraci aplikace je vyžadováno tzv. BundleID aplikace (které je vyžadováno například i pro nahrání aplikace na fyzické zařízení). Následně po registraci aplikace je vygenerován soubor GoogleService-Info.plist, který obsahuje konfigurace potřebné pro propojení s mobilní aplikací. Zmíněný soubor se umístí dovnitř do složky se zdrojovými soubory aplikace. Dále je nutné za pomoci již zmíněných CocoaPods stáhnout SDK, tedy tzv. Pody ke službám, které jsou pro funkci aplikace potřeba. V případě této aplikace to jsou Pody 'Firebase/Firestore' a 'Firebase/Auth'. Pro to, aby se Firebase

propojila s aplikací, ve chvíli kdy se aplikace načte, je nutné do třídy AppDelegate přidat inicializaci databáze.

Posledním krokem je finální kontrola funkčnosti komunikace s aplikací. Projekt je potřebné zkompilovat. Pokud se kompilace provede a nenastanou nějaké chyby, je mobilní aplikace úspěšně propojena s Firebase. Popřípadě je možné zkontrolovat v projektu, v příslušné složce, zda se zde opravdu nachází potřebné Pody pro funkčnost Firebase databáze a autentizace.

4.1.3 Firebase autentizace - Firebase Authentication

Prvním krokem k funkční autentizaci je mít v projektu obsažený tzv. Pod příslušící autentizaci. Dále je nutné ve Firebase webovém grafickém rozhraní daného projektu povolit přihlašování například za pomoci emailu a hesla.

Třetím krokem je inicializace databáze ve třídě AppDelegate, pokud tak nebylo učiněno již dříve. Následně je potřebné ve třídě, která je propojena s oknem pro přihlašování a registrování uživatele, přidat metodu pro autentizaci. Pokud ještě ve třídě není importována knihovna Firebase, je nutné její import přidat. V některých případech může být nutné více specifikovat importovanou knihovnu pomocí 'Firebase/Auth'.

Po přidání potřebných knihoven je již možné Firebase autentizaci využívat. Tato služba poskytovaná platformou Firebase umožňuje po zpracování vstupů od uživatele jeho registraci, přihlášení a na závěr i odhlášení. Pro zmíněné operace jsou ze strany Firebase definována určitá pravidla, například email zadávaný uživatelem při registraci musí být validní a heslo zadávané uživatelem musí být minimálně šest znaků dlouhé.

4.1.4 Firebase databáze - Firebase Firestore

Po úspěšném propojení aplikace s Firebase a autentizaci uživatele je dalším krokem nastavení Firestore databáze, kterou aplikace využívá. Jak již bylo zmíněno, při zakládání databáze je možné vybírat ze dvou módů, z produkčního nebo testovacího. Po vybrání módu následuje zvolení umístění cloudového úložiště, v případě této aplikace bylo zvoleno eur3 (europe-west). Po zvolení těchto dvou parametrů je možné již vytvořit databázi Firestore.

Posledním krokem je inicializace databáze v již zmiňované třídě AppDelegate. Zkontrolovat, zda inicializace databáze proběhla v pořádku je možné například za použití výpisu do konzole. Vypíše se proměnná, do které se uloží instance databáze. Pokud se do konzole vypíše například následující řetězec <FIRFirestore: 0x600011d72c70>, vše proběhlo v pořádku a je možné databázi používat pro účely aplikace.

Pro práci s Firebase databází je poté potřebné vytvořit její instanci i ve třídě, která s databází pracuje a která zpracovává informace od uživatele. Databáze obsahuje kolekce dat, které reprezentují databázové tabulky a do příslušné kolekce se vkládají záznamy, které jsou označovány jako dokumenty. Ve výchozím nastavení Firebase se v databázi generuje jednotlivým záznamům, respektive dokumentům

vlastní identifikátor, tento identifikátor lze ale nastavit na jinou hodnotu. V případě této aplikace byla jako identifikátor zvolena hodnota epc. Výhodou využití hodnoty epc je jednodušší identifikace záznamů.

4.1.5 Návrh databáze

Data jsou uložena v cloudovém úložišti. Přesněji v cloudovém úložišti řešeném platformou Google Firebase.

Do databáze posílá aplikace do hlavní tabulky tato data: epc, název knihy, autora knihy, žánr, do kterého kniha spadá, popis - poznámku ke knize a email uživatele, který tento záznam do databáze uložil. Všechny zmíněné ukládané položky jsou datového typu string. Povinné hodnoty pro korektní uložení do databáze jsou následující: epc, název knihy, autor knihy, žánr a email uživatele. Položky tag epc a email uživatele nejsou však od uživatele vyžadovány, aplikace je předává automaticky. Všechny tyto hodnoty se vyplňují v okně, které je pro tento účel vytvořené. Konečná verze formuláře na vyplnění informací o knize bude popsána a ukázána dále v dokumentu.

Pro umožnění třídění knih, což je jedním z požadavků zadání, je do databáze přidána ještě jedna tabulka. Zmíněná tabulka obsahuje jen druhy žánrů a každý druh žánru je označen identifikačním řetězcem, který vygeneruje platforma Firebase. Žánry jsou přidávány za pomoci okna, které je vytvořené pro tento účel.

Ve Firebase projektu založeném pro tuto aplikaci se také ukládají záznamy o uživateli, je ukládán email a heslo. Tyto záznamy se neukládají do databázové části, ale jsou k nalezení pod autentizací. Je možné tyto záznamy získat a dále s nimi pracovat, například je možné je ukládat do databáze. V případě potřeby je možné ukládat i více informací o uživateli, například mobilní číslo nebo je možné propojení s různými účty uživatele, například s Google účtem. Není tedy nutné využívat pro přihlášení email a heslo, je zde možnost i jiných způsobů registrace.

4.1.6 Návrh základní funkcionality aplikace

Aplikace uživateli umožňuje dvě základní akce. Uživatel je schopen pomocí aplikace načítat všechny dostupné tagy, které je možné dále uložit do již zmíněné databáze. Do databáze se ukládají všechny potřebné informace pro následnou identifikaci tagu, dále pak informace týkající se samotné knihy, do které je příslušný tag vložen. Je nutné umožnit uživateli ukládat a mazat záznamy o knihách.

Druhou funkcí aplikace je zobrazení záznamů uložených v databázi. Záznamy aplikace umožňuje třídit a vyhledávat. Třídění záznamů probíhá dle definovaného parametru, který je uložen v databázi, v tomto případě se záznamy třídí dle žánru. Dále je nutné uživateli umožnit nalezený záznam o knize vybrat a zvolit tím tag, který se bude vyhledávat. Posledním krokem je samotné navigování uživatele k zvolenému tagu v prostoru.

Pro ukládání seznamu žánrů je vhodné vytvořit samostatnou tabulku. Uživateli je tedy umožněno přidávat libovolný počet nových žánrů, dle kterých se mají knihy

následně třídit. Pro přidávání nových žánrů je v aplikaci vytvořeno samostatné okno, kde je umožněno zkontrolovat i které žánry jsou již v databázi uloženy.

4.1.7 Vyhledávání knihy uložené v databázi

Uživatel smí vyhledávat jen knihy, které jsou uloženy v databázi. Pokud je potřebné najít nějakou knihu, s pomocí UHF RFID čtečky a UHF RFID tagu, je nutné vybrat danou knihu v seznamu knih uložených v databázi. Zmíněný seznam aplikace zobrazuje v odpovídajícím okně. V databázi by mělo být velké množství knih a proto by bylo dobré, aby existovala možnost vyhledávání knih podle zadané sekvence, například podle jména autora. Záznamy knih v databázi musí být možné ukládat, hledat a mazat.

4.2 Finální řešení aplikace a poznatky při vývoji

4.2.1 Finální aplikace

Po spuštění aplikace (zdrojový kód [zde](#)) je nutné pro pokračování zapnout čtečku, se kterou má aplikace komunikovat. Na mobilním zařízení je nutné povolit používání Bluetooth. Pokud je tedy čtečka zapnuta a používání Bluetooth povoleno, tak se daná čtečka spojí s mobilním zařízením a v aplikaci se zobrazí název čtečky. Po vybrání čtečky se uživateli zobrazí další okno aplikace určené pro přihlášení.

Přihlášení uživatele bylo přidáno z důvodu bezpečnosti, respektive z důvodu zabezpečení dat uložených v databázi. Databázový nástroj Firebase požaduje od vývojáře aplikace nějakým způsobem zabezpečit uložená data. Nejvhodnějším způsobem je umožnit přístup k datům jen uživatelům, kteří jsou přihlášení.

Aplikace umožňuje uživateli jak registraci, tak přihlášení.

Nový uživatel tedy může zvolit možnost registrace. Pro registraci je nutné zadat korektní emailovou adresu a heslo, které musí splňovat určitá kritéria, například musí obsahovat nejméně šest znaků.

Pro přihlášení již registrovaného uživatele je nutné správně zadat přihlašovací údaje uložené v databázi, tedy emailovou adresu a heslo.

Pokud se uživatel úspěšně přihlásí nebo zaregistruje, je možné pokračovat dál do základního menu, ve kterém je mimo jiné uživateli umožněno odhlášení. Aplikace je rozdělena na dvě hlavní části dle functionality. Dále jsou v aplikaci ještě dvě doplňující aktivity.

První část slouží pro uložení knih do databáze a pro úpravu obsahu databáze. Je možné ukládání knih a také mazání uložených knih z databáze. Po kliknutí na tlačítko, které je určeno pro inventarizaci, se uživateli zobrazí okno s jehož pomocí se vyhledají všechny dostupné tagy v okolí čtečky. Dostupné tagy se uživateli zobrazují seřazené podle hodnoty RSSI, která reprezentuje sílu signálu získaného od tagu. Nejvyšší hodnota RSSI teoreticky představuje tag nejbližší ke čtečce. Po rozkliknutí tagu, který je reprezentovaný hodnotou epc (společně s již zmíněnou RSSI

hodnotou), se uživateli zobrazí další okno s informacemi o daném tagu. Uživatel zde má možnost uložit knihu do databáze, smazat knihu z databáze a prohlédnout si informace o tagu, které aplikace získává za pomoci knihovny NurApi.

Ukládání tagů, respektive knih do databáze je možné pokud uživatel vyplní všechny povinné údaje o příslušné knize. Povinné údaje k vyplnění jsou tedy název knihy, jméno a příjmení autora a žánr. Nepovinný údaj je případná poznámka ke knize, která může sloužit například pro rozlišení kopií stejné knihy.

Pokud uživatel chce uložit knihu od stejného autora, který je již v databázi uložen, je možné zobrazit si nápovědu jména autora. Zmíněná nápověda může být vhodná u autorů jiné národnosti se složitějšími jmény.

Pokud je kniha s příslušným epc již v databázi uložena, je uživateli zobrazeno upozornění. Uživateli je v upozornění předána i informace, pod jakým názvem knihy je daný tag v databázi uložen. Aplikace nedovolí nový záznam o knize (název, autor, žánr a poznámka) znovu uložit a tak přepsat stávající data. Pro aktualizaci, je nutné daný záznam z databáze nejdříve vymazat s pomocí tlačítka tomu určenému a následně vyplnit údaje pro aktualizaci a uložit nová data do databáze. Pro komunikaci aplikace s uživatelem a pro zpětnou vazbu bylo do okna vloženo textové pole, do kterého aplikace wpisuje aktuální stav nebo informaci, například zda bylo uložení úspěšné, nebo co bylo příčinou nezdařené operace.

Poslední functionalita první části aplikace představuje zobrazení informací o tagu. K prohlédnutí jsou následující hodnoty: EPC, kanál, RRSI, Timestamp, frekvence a ID antény.

Jednotlivá důležitá okna aplikace

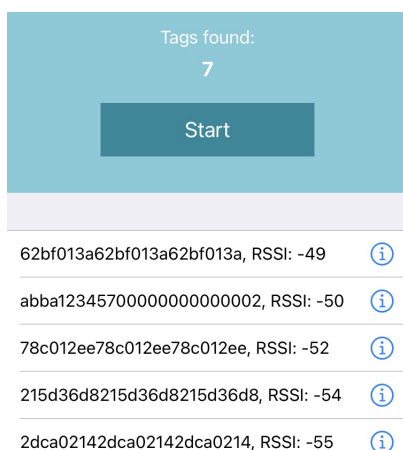
Toto okno (obrázek 4.2) zobrazuje seznam tagů, které jsou dostupné v okolí používané čtečky. Po stisku tlačítka se začne postupně objevovat na obrazovce seznam tagů. Seznam je průběžně řazený dle hodnoty RRSI způsobem kde nejvyšší hodnota je nahoře. Po opětovném stisknutí tlačítka, se přestanou načítat další nové tagy. Uživatel poté může vybrat tag, který chce nějakým způsobem modifikovat. Po kliknutí na řádek s daným epc, se uživateli zobrazí další okno.

Na obrázku 4.3 je okno aplikace, které je určeno pro ukládání záznamů o knize do databáze. Kromě ukládání do databáze je v tomto okně možné vidět informace o tagu, tedy jeho epc, RRSI a další.

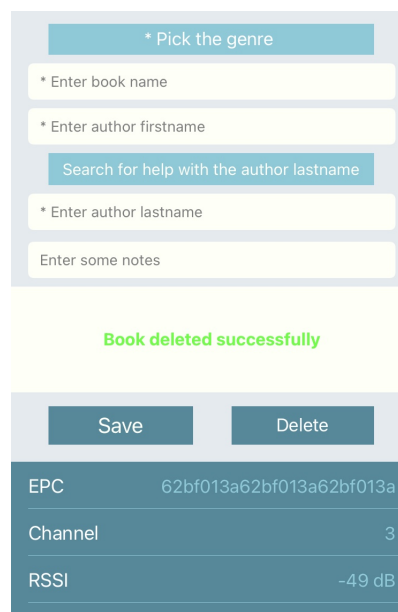
Při ukládání do databáze je možné vyžádat nápovědu jména autora, pokud je již daný autor uložen v databázi. Pokud aplikace nalezne autora, jehož jméno obsahuje zadanou sekvenci, je uživateli do okénka zobrazeno celé autorovo jméno. Uživatel však musí jméno napsat sám, ale má alespoň nápovědu jak se autorovo jméno píše. Zamezí se tak ukládání různých verzí autorova jména v případech kdy má autor velmi neobvyklé jméno.

Protože se nápověda zobrazuje jen do okénka, které není uživateli dovoleno editovat, je možné uživateli zobrazit více výsledků v případě nalezení více autorů.

Povinným atributem je také žánr, do kterého kniha spadá. Pro vybrání žánru je na obrazovce tlačítko, po jehož rozkliknutí se uživateli zobrazí seznam všech žánrů, ze kterých je možné vybírat. Pokud chybí daný žánr, je možné přejít do okna apli-



Obrázek 4.2: Dostupné tagy



Obrázek 4.3: Informace o tagu

kace, které je určené pro ukládání nových žánrů do databáze. Uživatel uloží nový, potřebný žánr a přejde zpět k ukládání knihy do databáze. Pokud je v seznamu žánrů požadovaný žánr, uživatel jej vybere a dále již aplikace ví, který žánr uložit k dané knize.

Pokud se uživatel pokusí uložit knihu, která je již v databázi uložená, je mu zobrazena hláška, že tato operace již není možná. Stejně tak i pokud uživatel zapomene vyplnit například jméno nebo příjmení autora, není mu povoleno uložit data v tomto neúplném stavu do databáze.

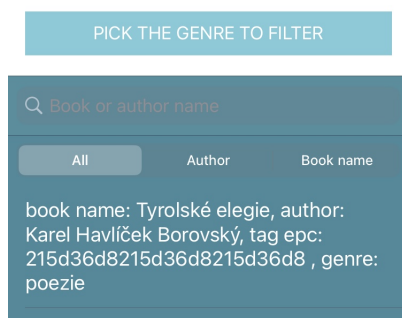
Druhá část představuje hledání samotné knihy v prostoru. Pro hledání knihy v prostoru jsou nutné dva základní kroky.

Prvním krokem je vybrání knihy, kterou chce uživatel hledat. Pro účely tohoto kroku je v aplikaci vytvořeno okno. V tomto okně viz obrázek 4.4 jsou uživateli zobrazeny všechny knihy uložené v databázi. Uživatel může použít vyhledávací okénko. Během psaní do vyhledávacího okénka se automaticky obnovuje a konkretizuje seznam knih na základě znaků, které uživatel do okénka vepisuje. Aplikace uživateli umožňuje filtraci výsledků hledání. Pokud uživatel zvolí filtraci dle jména autora, aplikace se zaměří jen na jméno autora a budou se uživateli zobrazovat jen knihy jejichž autor má ve jménu požadované znaky.

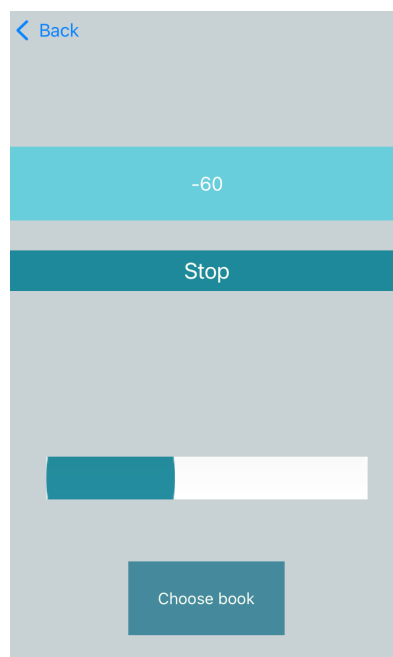
Alternativou k filtraci a vyhledávacímu okénku je třídění knih dle žánru. Pokud chce uživatel knihy třídit dle žánru, stiskne tlačítko k tomu určené a zobrazí se výběr žánrů. Požadovaný žánr se vybere a aplikace zobrazí jen knihy, které spadají do daného žánru.

Filtrování a třídění nelze kombinovat, pokud uživatel vybere žánr knihy a výsledky bude chtít ještě filtrovat, filtrování se uskuteční bez ohledu na žánr.

Pokud si uživatel vybere knihu, kterou chce hledat, stiskne tlačítko pro přechod



Obrázek 4.4: Knihy v databázi



Obrázek 4.5: Hledání tagu

na další okno určené pro hledání tagu v prostoru.

Druhým krokem je tedy vyhledávání knihy v prostoru. Uživatel musí mít v ruce kromě mobilního zařízení i UHF RFID čtečku s jejíž pomocí se vyhledávání uskuteční. Po stisku tlačítka „Start“ čtečka začne vysílat signál. Pokud je hledaný tag v dosahu čtečky, odešle odpověď, kterou aplikace zpracuje a zobrazí uživateli. Zároveň se relativní vzdálenost ukazuje i v tzv. ukazateli průběhu, který připomíná teploměr. V závislosti na tom, jak se uživatel přibližuje/vzdaluje od knihy (respektive tagu), ukazuje se příslušná hodnota RSSI. Pokud je čtečka v bezprostřední blízkosti tagu, je ukazatel průběhu plný.

Získávané hodnoty RSSI jsou v aplikaci dále ještě zpracovávány předtím, než se zobrazí uživateli. Komunikace mezi čtečkou a daným tagem je velmi rychlá. Pokud by se uživateli měla zobrazovat nezpracovaná data tak, jak je získává aplikace od čtečky, bylo by to pro uživatele příliš rychlé a náročné na sledování. Proto aplikace provádí s daty operaci ve formě velmi jednoduché filtrace.

Filtrace zpracovává definovaný počet RSSI hodnot, které přijme aplikace. Po přijetí daných hodnot využívá klouzavého průměru, který se po získání prvotních hodnot začne aktualizovat při každé nové hodnotě. Uživatel má tak velmi rychle zpětnou vazbu v podobě RSSI hodnot, které jsou díky průměru realistické. Další informace o tom, jak je možné zpracovávat hodnoty, budou rozebrány dále v práci v kapitole o testování aplikace.

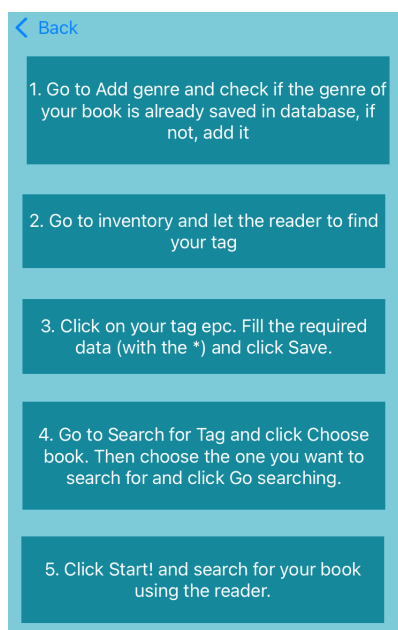
Vzorec pro jednoduchý klouzavý průměr (SMA) je následující:

$$SMA = \frac{P1 + P2 + \dots Pn}{n}$$

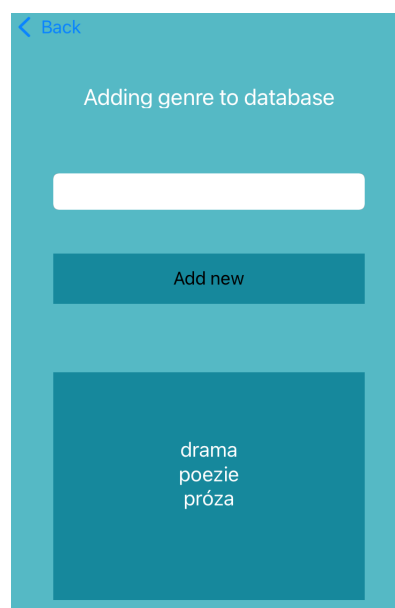
Kde n bylo zvoleno 6.

Zmíněné doplňující aktivity aplikace jsou okna určená pro ukládání nových žánrů do databáze a pro nápovědu, respektive jednoduchý návod.

První částí je okno určené pro ukládání žánrů (obrázek 4.7). Ideálním postupem použití aplikace by bylo prvně zkontrolovat toto okno, zda žánr knihy, kterou uživatel chce do databáze ukládat je v tabulce žánrů již uložený. Uložené žánry jsou zobrazeny v textovém poli. Pokud uživatel nenalezne požadovaný žánr stačí jen zaplat název nového žánru do určeného pole a stisknout tlačítko pro přidání. Pokud by se uživatel spletl a chtěl by přidat žánr, který již v databázi uložený je, aplikace tuto akci nedovolí a do textového pole zapíše zprávu, že daný žánr již v databázi je a že se tedy ukládání do databáze neprovedlo.



Obrázek 4.6: Návod k aplikaci



Obrázek 4.7: Přidávání žánru

Posledním oknem (obrázek 4.6), které je možné najít v základním menu aplikace je návod, jak by se měla aplikace používat. Návod obsahuje pět kroků, ve kterých je popsána základní funkcionality aplikace. Za předpokladu, že uživatel tyto kroky bude ve správném pořadí následovat, by mělo použití aplikace proběhnout bez komplikací. Návod je napsán v anglickém jazyce, stejně jako zbytek aplikace. Pokud by však aplikaci využívalo více uživatelů, je možné nastavit změnu jazyka na jazyk preferovaný daným uživatelem.

4.2.2 Poznatky při vývoji aplikace

Používání aplikace - řešení některých nedostatků aplikace

Aplikace je psána v jazyce Swift a je používána knihovna NurApi Bluetooth. Podpory a dokumentace ze strany NurApi vývojářů je pro jazyk Swift velmi málo. Některé funkcionality, které nabízí knihovna pro vývoj Android aplikací, vůbec nejsou dostupné pro iOS. Je tedy nutné se této situaci přizpůsobit a nahradit některá řešení jinými.

Pokud chce uživatel načíst novou knihu a uložit ji do databáze, je nejlepším řešením snížit výkon čtečky, čímž se zajistí, že uživatel nalezne jen jednu knihu, tedy knihu, která je v bezprostřední blízkosti čtečky. Tuto funkci je velmi složité nastavit ve Swift verzi této knihovny. Dospělo se tedy k jinému řešení, kterým je zobrazení seřazených tagů podle hodnoty RSSI a vypsání příslušné RSSI hodnoty, tedy síly, podle které se uživatel může řídit a zvolit tak tag s nejvyšší hodnotou RSSI. Tag s nejvyšší hodnotou RSSI, respektive silou je považován za tag nejbližší k čtečce. Hodnoty RSSI se pohybují v záporných hodnotách, v případě této aplikace se nejčastěji objevují hodnoty v rozsahu -75 až -40 dBm.

Tag ID

Tag ID hodnota je neměnná. Hodnota je přidělována při výrobě tagu a měla by být jedinečná. S pomocí tag ID by se mohl vyřešit problém více kopií stejné knihy. Problém spočívá v tom, jak ukládat a zobrazovat uživateli stejné knihy. Tento problém má minimálně dva způsoby řešení.

První řešení je mít dvě tabulky v databázi. Jedna tabulka by byla hlavní se základním, obecným seznamem knih a v druhé tabulce by byly uloženy konkrétní výtisky knih. Při tomto řešení by bylo výhodné použít tag ID, kde hodnota epc by mohla být nastavená u stejných knih na stejnou hodnotu a identifikace kopie by byla prováděna pomocí odlišných tag ID hodnot. Problémem ale je, že knihovna NurApiBluetooth neposkytuje možnost práce s touto hodnotou.

Druhým řešením zmíněného problému je uložení všech informací ohledně knih do jedné tabulky. Identifikace by byla prováděna s použitím hodnoty epc, která ale musí být u každého tagu nastavena na jedinečnou hodnotu. Byla tedy zvolena tato druhá možnost.

4.2.3 Rozšíření vyhledávání i pro QR a čárové kódy

Pokud by bylo nutné vytvořit aplikaci, která je schopna zpracovávat i QR a čárové kódy, bylo by potřebné udělat hned několik změn. Nejzásadnější změna/rozdíl je ve čtení/komunikaci s těmito dalšími technologiemi. Původní aplikace je koncipovaná pro komunikaci s RFID čtečkou. RFID čtečka ale není schopná číst QR a čárové kódy. Tedy pokud by měla aplikace zpracovávat tyto kódy, musel by tyto kódy číst přímo chytrý mobilní telefon. Aplikace by musela mít nové okno, přes které by se kódy skenovaly.

Pomocí QR a čárových kódů se knihy nebudou moci hledat. Pokud by byl v knize jen QR nebo čárový kód byla by klíčová funkcionalita aplikace ztracena. Bylo by totiž možné jen naskenovat kód do aplikace a uložit knihu do databáze knih nikoli ji najít.

Na druhou stranu by bylo použití i čárových a QR kódů logickým krokem, protože všechny knihy v knihovně by měly být nějakým způsobem označeny a pravděpodobně jsou označeny právě čárovým nebo QR kódem. Tedy čárové a QR kódy by zůstaly a RFID tagy by byly jen dodatečným vylepšením již fungujícího systému sledování knih.

4.2.4 Potencionální úpravy pro zveřejnění aplikace

Aplikace nebyla vyvíjena se záměrem pro zveřejnění a zpřístupnění širokému spektru uživatelů. Z tohoto důvodu je aplikace aktuálně ve stavu, kdy by bylo potřebné ji optimalizovat a upravit pro zveřejnění v App Store. Nebyl ani definován způsob jakým by se mohla aplikace používat, respektive kdo by byli její uživatelé. Všechny dále zmíněné úpravy jsou jen návrhem v závislosti na pravděpodobném způsobu použití. Předpokladem pro používání aplikace by bylo zavedení RFID technologie v knihovnách, ve kterých by se měla aplikace využívat.

Pokud by k aplikaci mělo přístup více lidí a byla by využívána ve více knihovnách, bylo by vhodné doplnit do databáze také tabulku s knihovnami a na základě toho dále upravit databázi.

Využití funkce hledání knihy, kterou aplikace nabízí, by mohlo být následující. Aplikace by mohla být propojena se systémem knihovny, ze kterého by získávala data o stavu knihy, například zda není hledaná kniha rezervována. Uživatel by mohl chodit po knihovně a knihy které by zvolil, že si chce odnést domů, by načel pomocí čtečky a v aplikaci by zadal, že si danou knihu půjčuje domů. Možné by bylo také například nasměrování uživatele přesně ke zvolenému výtisku knihy nebo ke všem výtiskům, které se aktuálně v knihovně nachází. Aplikace by tak nesloužila jen pro zjednodušení vyhledávání knih, ale poskytovala by další praktické funkce.

4.3 Aplikace v praxi

4.3.1 Použití aplikace s RFID čtečkou a tagy

Aplikace byla psána pro následující čtečku a tagy.

Použitá čtečka

Použitá čtečka je od značky Alien, konkrétně model ALR-S350-EMA. Tato čtečka pracuje na ultra vysoké frekvenci, tedy UHF. Dosah signálu je tedy kolem jednotek až desítek metrů.

Tagy

Tagy použité pro testování aplikace jsou od značky SMARTRAC - Dogbone. Pracují stejně jako čtečka na ultra vysoké frekvenci. Tagy jsou pasivní, tedy nejsou schopné vysílat, musí počkat až čtečka vyšle rádiové vlny, díky kterým je poté tag schopen odeslat odpověď čtečce.

Aplikace v praxi

Čtečka s mobilním zařízením komunikuje za použití technologie Bluetooth. Pro použití aplikace způsobem, jak bylo zamýšleno, je nutné umístit RFID tagy do knih v knihovně. Například použité tagy, které jsou od značky SMARTRAC mají z druhé strany samolepící povrch, který je možné využít k připevnění tagu ke knize. Pro práci s tagem je nutné tag načíst a uložit do databáze. Načítání a následné ukládání tagů je možné uskutečnit v případě, kdy je tag mimo knihu i pokud je ke knize již připevněn.

Po otevření aplikace se v úvodním okně zobrazí všechny dostupné zapnuté čtečky. Vybráním požadované čtečky, což je realizováno kliknutím na název dané čtečky, daná čtečka naváže spojení s aplikací. Spojení trvá do doby, než se aplikace ukončí. Během spojení aplikace se čtečkou dochází ke stálé komunikaci a získávání informací od čtečky. Čtečka získává od RFID tagu důležité informace, kterými jsou již zmíněné hodnoty - tag epc, RSSI a další.

Jak již bylo vysvětleno v části s popisem finální verze aplikace, je aplikace schopna umožnit ukládání tagů, respektive knih do databáze a také je schopna uživatele nasměrovat k hledanému tagu. Uživatel si vybere hledanou knihu ze seznamu všech uložených knih a stiskne tlačítko pro začátek hledání. Ve chvíli, kdy čtečka zachytí signál od tagu, pošle dané informace aplikaci. Aplikace získané hodnoty RSSI začne zpracovávat pomocí klouzavého průměru. Většinou, když čtečka zachytí slabý signál, je hodnota tohoto signálu cca -75 dBm. Jak se uživatel postupně přibližuje k tagu, síla signálu stoupá a aplikace zobrazuje vyšší RSSI hodnoty, které mohou dosahovat -40 dBm. Zároveň je v aplikaci i tzv. zobrazení pokroku (Progress View), které uživateli zobrazuje hodnoty interaktivněji a pohodlněji.

Většinou se podaří zachytit slabý signál ve vzdálenosti cca 1-1,5 m od tagu. Plný signál je možné zachytit v bezprostřední blízkosti tagu. Výjimečně je možné, že aplikace zachytí silnější signál z relativně větší vzdálenosti. Důvodem tohoto jevu je většinou zaměření čtečky přímo na místo, kde je se tag nachází.

Uživateli aplikace je doporučeno hledání tagu následujícím způsobem. Uživatel stojí před policemi plnými knih, zkusí ze vzdálenosti cca 0,4 metru projet odshora dolů přes celý regál s knhami. Dalším krokem je projet vodorovně polici, u které zachytil nejvyšší signál. Po tomto kroku by měl být uživatel schopen knihu bez problému nalézt.

Poznatky

Pro získání nejvyšší hodnoty RSSI je potřeba přímé viditelnosti mezi tagem a čtečkou. Pokud je docílena přímá viditelnost, je aplikace, respektive čtečka schopna

tag zaznamenat již z větší vzdálenosti. Pokud se tag umístí do knihy a je natočen směrem na čtečku, hodnota RSSI se výrazně nezmění. Hodnota RSSI se ale částečně sníží, pokud je kniha ke čtečce otočena hřbetem.

Knihy, respektive papír není pro průchod signálu zásadní překážkou. Pokud by bylo ovšem v místnosti vlhko a knihy by nasály vlhkost, mohly by již představovat překážku pro signál.

Pokud by byly knihy uloženy v kovových regálech je pravděpodobné, že by se maximální získaná hodnota RSSI ještě více snížila. V případě, že je mezi čtečkou a hledaným tagem kovové těleso, které tag plně zakrývá, je velmi obtížné, až nemožné pro čtečku získat od tagu zpětný signál ve formě RSSI.

4.3.2 Testování

Tag v knize

Pokud se RFID tag vloží do knihy, jeho schopnost vysílat odpovědi ve formě signálu mohou být snižené v závislosti na následujících případech.

Síla signálu tagu vloženého do knihy kde kniha není uložena do knihovny, ale leží například na stole, je snížena jen nepatrně. Pokud se kniha spolu s tagem uvnitř vloží do knihovny, je síla signálu již patrně snížena. Samozřejmě také záleží na typu knihy, zejména na vazbě, zda je měkká nebo tuhá.

Vyhledávání tagu

Vyhledávání požadovaného tagu funguje na základě následujícího principu. Uživatel si zvolí v seznamu knih požadovanou knihu. Aplikace předá obrazovce (určené k hledání tagu) příslušné epc tagu, který náleží dané knize. Po stisknutí tlačítka Start začne čtečka vysílat rádiové vlny, které tagy přijmou, zpracují je a odešlou odpověď čtečce. Čtečka přijme odpověď od všech dostupných tagů a pošle informace aplikaci. Aplikace má uloženou hodnotu epc, která přísluší hledanému tagu a pokaždé když získá data od čtečky projde všechny hodnoty epc a porovná je s uloženou epc hodnotou. Pokud dojde ke shodě v hodnotách epc, tedy aplikace získá od čtečky informace o hledaném tagu, uloží se získaná hodnota RSSI. Uložená hodnota je ještě dále zpracovávána před samotným zobrazením uživateli.

Testování metod pro filtraci

Při vyhledávání tagu, jak již bylo zmíněno v popisu finální aplikace, je použit klouzavý průměr z definovaného počtu RSSI hodnot, které čtečka pošle aplikaci. Aplikace tedy počká až od čtečky získá daný počet RSSI hodnot. Získané hodnoty zpracuje, uloží si je do fronty, poté s každou novou hodnotou aktualizuje frontu, vypočítá a zobrazí konečnou hodnotu (průměr) uživateli. Bylo testováno více způsobů filtrování dat.

Prvním testovaným způsobem bylo vytvoření průměru ze získaných třech hodnot RSSI. Tento způsob byl jistě krokem ke zlepšení uživatelské přívětivosti v porovnání s původně žádným filtrem. Tento filtr nepoužíval klouzavý průměr, tedy

byly vzaty tři hodnoty, zpracovány a poté se čekalo na další tři hodnoty. Protože se zpracovávaly hodnoty jen jednou a poté zahazovaly, nebylo vhodné dělat průměr z vyššího počtu hodnot. Problém tohoto řešení byly časté vysoké výkyvy v zobrazovaných hodnotách. I přes výskyt výkyvů v hodnotách bylo možné rozlišit hodnoty, podle kterých se řídit a proto i tato metoda alespoň částečně uživateli pomohla nalézt požadovaný tag.

Druhým způsobem bylo vytvoření filtrace, která vrací medián z přijatých hodnot RSSI. Pořád byly použity trojice hodnot, které se po zpracování zahazovaly. Způsob se zjišťováním a zobrazováním mediánu se ukázal jako vhodnější řešení. Nedošlo k takovým výkyvům hodnot. Postupné přibližování se k hledanému tagu mělo tak plynulejší průběh a použití aplikace při hledání požadovaného tagu bylo tak pro uživatele přívětivější.

Třetím testovaným způsobem bylo použití zmíněného klouzavého průměru. Hledání probíhá mnohem plynuleji, byl zvolen počet šesti hodnot, které se postupně přidávají a poté mažou z fronty. Hodnota se přidá do fronty, šestkrát s ní aplikace pracuje, posouvá se postupně a poté se z fronty smaže. V porovnání se zjišťováním mediánu místo průměru, je průměr mnohem rychleji schopen reagovat na změny v odchýlení se od tagu. Varianta s mediánem je více plynulá, tedy se může zdát lepší, ale přesnost, na kterou je schopná detekovat odchýlení od tagu je mnohem menší. Bylo testováno odchýlení jak do stran, tak směrem nahoru a dolů. Použití průměru se jeví jako lepší varianta.

Vzdálenosti dosahu v závislosti na překážkách

Byly testovány různé situace, které mohou nastat, a na jaký dosah je v těchto situacích možné data získávat. Nejprve byl testován samotný dosah signálu, dále byly použity překážky a bylo sledováno jak se dosah signálu mění. V neposlední řadě bylo testováno, jaký dosah signálu je při hledání v knihovně.

Pro testování byly použity tagy od značky SMARTRAC (pokud není jinak specifikováno, jde při testech o tyto tagy) společně s jedním vzorkem tagu od značky Alien.

Při testování samotného dosahu se dospělo k následujícím výsledkům za následujících podmínek.

Dosahy signálu:

- **Přímá viditelnost** - tagy byly schopné komunikovat se čtečkou na vzdálenost **cca 6 metrů**, tag od značky Alien byl schopen komunikace na vzdálenost **kolem 4,5 metru**.
- V knize s měkkou vazbou, kdy byla kniha směrem ke čtečce, bylo možné signál zachytit ze vzdálenosti v rozmezí **od 4,5m do 6m**, u tagu značky Alien bylo možné signál zachytit ze vzdálenosti **cca 4 metrů**.
- V knize s měkkou vazbou, kdy kniha byla otočena hřbetem, bylo možné signál zachytit z cca stejné vzdálenosti, tedy ze vzdálenosti **4,5 až 6 metrů** a u Alien tagu **4 metry**.

- V knize s pevnou vazbou bylo možné signál zachytit ze vzdálenosti **cca 6 metrů**, v případě tagu značky Alien byl signál zachycen ve vzdálenosti **cca 4 metrů**.
- V knize s pevnou vazbou, kde hřbet je otočený směrem ke čtečce byl získán signál znovu z podobné vzdálenosti, tedy **kolem 6 metrů** a u Alien to bylo **kolem necelých 4 metrů**.
- V případě že překážku tvořil předmět, který byl kovový, nebylo signál možné získat, stejně tomu bylo i u vody.

Při testování dosahu v knihovně (obrázek 4.8), kde je pohromadě více tagů a více knih, byla získána následující data.

- Signál je možné zachytit ze vzdálenosti **cca 1,5 metru**.
- Pokud je čtečka od tagu ve vzdálenosti **cca 40 cm** je možné již poměrně snadno zjistit polohu tagu s **přesností cca 6 knih**, tedy pokud se na knihu s hledaným tagem míří rovně a poté, když se začne čtečka posouvat jiným směrem, začne se v aplikaci zobrazovat, že je tag ve větší vzdálenosti od čtečky. Tag značky Alien byl nepatrně citlivější na změnu směru a bylo tak možné nalézt danou knihu s větší přesností.



Obrázek 4.8: Vyhledávání v prostoru

Všechna tato testování proběhla s použitím vzorku tagů SMARTRAC a jen jednoho tagu od značky Alien, není tedy možné generalizovat zmíněné výsledky. Je

ale zajímavé vidět, že se dosahy signálu tagů mohou odlišovat i na základě toho, jaký je jejich výrobce.

Uživatelská přívětivost

Během vývoje aplikace se myslelo i na uživatelskou přívětivost, jakými způsoby aplikaci upravit tak, aby se uživatel s aplikací pracovalo dobře. Jak je v práci několikrát zmíněno, byla přidána nápověda jména autora, praktická okna určená k vybírání žánru a textové pole, které poskytuje uživateli zpětnou vazbu.

Co se týče přihlašování uživatele, tak při testování aplikace není velmi praktické mít povinnost při každém spuštění aplikace vyplňovat uživatelský email a heslo. Bylo by například možné vytvořit automatické vyplňování přihlašovacích údajů v případě, že si uživatel své přihlašovací údaje uloží a zvolí variantu automatického vyplňování.

4.3.3 Kritické zhodnocení aplikace, možná vylepšení

Zhodnocení

Aplikace splňuje ve své funkčnosti všechny body zadání. Je možné načítat dostupné tagy, ukládat je a mazat. Úpravy záznamu o knize v databázi lze dosáhnout smazáním a následným přidáním knihy již v požadované podobě. Aplikace také uživateli zobrazuje informace o daném tagu. Dobrým implementovaným vylepšením je zpětná vazba pro uživatele, která například uživatele upozorní v případě, kdy je daná kniha již v databázi uložena. Nedojde tak k nedorozumění a je jasné, jak postupovat, pokud uživatel chce změnit informace o dané knize. Jistě uživatel také ocení nápovědu jména autora a uživatelsky přívětivé vybírání žánru knihy. Nápovědu jména by bylo možné optimalizovat již zmíněným vybráním z nabízených autorů.

Pro snazší zvolení knihy ke hledání jsou v aplikaci implementovány dvě funkce. První funkcí je možnost vyhledávání dle žánru knihy. Pro výběr žánru knihy, který se má uživateli zobrazit je v aplikaci implementováno jednoduché okno pro vybrání z uložených žánrů. Toto okno se uživateli zobrazí po kliknutí na tlačítko pro vybrání žánru a po tom, co uživatel vybere požadovaný žánr, okno zase zmizí. Není tedy zbytečně zabírán prostor a je více místa na zobrazování výsledků hledání. Druhou funkcí je filtrování výsledků hledání podle jména autora nebo podle názvu knihy. Obě zmíněné funkce ulehčují uživateli práci při hledání dané knihy. Drobným nedostatkem je, že nelze kombinovat tyto dvě funkce dohromady, tedy uživatel může buď jen filtrovat dle názvu knihy popřípadě jména autora nebo může třídit výsledky hledání dle žánrů.

Při vyhledávání knihy v prostoru jsou zobrazovány hodnoty reprezentující relativní vzdálenost dvěma způsoby. Oba způsoby jsou praktické, každý v jinou situaci. Zobrazování hodnoty je vhodné v případě, kdy uživatel potřebuje přesnou hodnotu a zobrazování v tzv. ukazateli průběhu je vhodné pro samotné hledání z důvodu lepší orientace uživatele když vzdálenost vidí graficky.

Během testování aplikace při opakovaném hledání různých knih se někdy stane, že se hledání zpomalí, v krajních případech není možné. Pokud se však aplikace restartuje je vše znovu funkční. Tento jev byla snaha odstranit, ale kvůli již zmiňované

mizivé dokumentaci ke knihovně NurApiBluetooth se řešení nepodařilo nalézt. Pravděpodobně je chybně implementovaná funkce pro zastavení a spuštění datového toku mezi čtečkou a mobilním zařízením. Aplikace tyto funkce používá po vzoru demo verze aplikace, kde se ale tento problém také vyskytuje, tedy pravděpodobně není chyba na straně samotné aplikace.

Aplikace uživateli umožňuje přidávat nové žánry a mít kontrolu nad již přidanými. V neposlední řadě je možné si v aplikaci zobrazit jednoduchý návod s kroky vysvětlujícími jak správně aplikaci používat, aby nedocházelo ke zbytečným komplikacím.

Kvůli zabezpečení dat bylo do aplikace přidáno povinné přihlašování uživatele. Pro osobní používání a testování aplikace není tato funkcionality nutná přidávat, ale protože aplikace využívá databázi Firebase je ze strany této platformy doporučováno využít již zmíněných pravidel. Tato pravidla ve většině případů využívají právě autentizaci uživatele a proto byla tato funkce přidána.

Vylepšení

Jak bylo již zmíněno, podpora pro operační systém iOS ze strany vývojářů knihovny NurApiBluetooth není silná. Příkladem problému, který by bylo dobré vyřešit, může být již zmíněný příliš vysoký vysílaný výkon ze strany čtečky, který není vítaný při snaze naskenovat jen jeden určitý tag. Pokud je výkon čtečky příliš velký a v okolí čtečky je více než jeden tag, není pak jasné, který tag vybrat. Tento problém by bylo možné vyřešit snížením výkonu čtečky na minimum a aplikace by načetla jen ten tag, na který by čtečka přímo mířila a byla by s ním v bezprostřední blízkosti. Výkon však nelze s pomocí knihovny NurApiBluetooth snížit, nebo alespoň dokumentace se o této možnosti nezmiňuje. Proto by bylo vhodné zaměřit se na tento problém a pokusit se najít alternativní řešení tohoto problému, které by snížení výkonu umožnilo.

Následující vylepšení aplikace se týká obrazovky umožňující ukládání informací o knize do databáze. Jak bylo již dříve popsáno, zmíněná obrazovka obsahuje možnost zobrazení nápovědy jména autora, pokud je daný autor již uložen v databázi. Nynější řešení funguje na principu, kde aplikace zobrazí jen návrhy jmen autorů. Nalezená jména jsou zobrazována do textového pole, které je použito jen na zobrazení. Vylepšením by mohla být možnost vybrat některého ze zobrazených autorů a aplikace by toto jméno rovnou doplnila za uživatele. Toto řešení nebylo aplikováno kvůli nároku na velkou část obrazovky, která již i tak obsahuje mnoho objektů. Problém zaplněné obrazovky by mohl být vyřešen zobrazením dalšího okna pro tento účel vytvořeného.

G senzor je dalším možným vylepšením aplikace. Jak již bylo zmíněno na začátku práce, společnost Apple prodává produkt nazvaný AirTag, tento tag komunikuje s mobilním zařízením za pomoci technologie Bluetooth. Pro tento produkt existuje aplikace, kde je možné uživatele nasměrovat k danému tagu. Nasměrování uživatele funguje na principu hry Samá voda, přehořívá a hoří. Zmíněná aplikace je ale také schopna uživatele směřovat příslušným směrem. Tato funkcionality by byla výrazným zlepšením co se týče uživatelské přívětivosti vyvíjené aplikace. Pro

toto vylepšení by se mohl použít zmíněný G senzor [9]. G senzor je měřič zrychlení, tedy měří zrychlení popřípadě naklonění mobilního telefonu ve směrech doprava, doleva, nahoru a dolů. Problémem je však relativní nestabilita dat, které aplikace přijímá od čtečky, tedy hodnota RSSI trpí velkými výkyvy. Kvůli výkyvům hodnoty RSSI není možné bezpečně poznat vzdálenost a také nelze přesně poznat, co zapříčinilo nižší hodnotu signálu v daný čas. Možná je například následující situace, uživatel nevědomě zamíří přímo na tag, když je v relativně velké vzdálenosti, získá se vysoká hodnota RSSI, tedy v aplikaci se bude uživateli zobrazovat, že je poměrně blízko k danému tagu. Poté se uživatel nepatrně pohne jiným směrem a již nebude možné získat žádný signál. Až poté, co se uživatel více přiblíží k tagu, začne daný tag komunikovat se čtečkou a ta aplikaci následně pošle další hodnoty RSSI získané od hledaného tagu.

V souvislosti s výkyvy hodnoty RSSI je v aplikaci použito filtrování s použitím klouzavého průměru. Tato filtrace by také mohla mít určitý vliv na chování aplikace při navigování uživatele pomocí G senzoru. V případě klouzavého průměru by mohlo dojít k optimalizaci hodnot RSSI, tedy nedocházelo by ke zmíněným výrazným výkyvům.

5 Závěr

Cílem této bakalářské práce bylo seznámení se s technologií RFID a vytvoření aplikace pro operační systém iOS, která umožňuje uživateli práci s RFID čtečkou a tagy v prostředí knihovny.

Práce se ze začátku zaměřuje na popis technologie RFID. V popisu jsou vysvětleny základní principy fungování technologie RFID, jsou popsány klíčové prvky, tedy ruční UHF RFID čtečka a UHF RFID tag.

Dále se práce zabývá programováním pro operační systém iOS, kde jsou vysvětleny základní principy společně s rozdíly v programování pro operační systém Android. Kromě těchto informací je popsána knihovna NurApi a jak se využívá pro komunikaci se čtečkou.

Následně je již pozornost věnována samotné aplikaci. Nejprve je popsána databáze, která se využívá společně s obecným návrhem aplikace. Je vysvětleno, proč byla pro aplikaci zvolena platforma Google Firebase jako databázový nástroj. Kromě samotného popisu platformy Firebase je popsáno, jak bylo této platformy využito v aplikaci a co všechno bylo nutné k dosažení funkčního spojení s touto databází. Dále je popsáno finální řešení aplikace společně s poznatky, ke kterým se dospělo během vývoje.

Ke konci práce je popsána aplikace v praxi, kde je specifikováno pro jaký model čtečky a pro jaké tagy byla aplikace vytvářena, jak se s aplikací pracuje a jaký je aktuální stav používání RFID technologie v praxi. Následně jsou v práci uvedena data vycházející z testování aplikace, návrhy na úpravy, které by vedly ke zlepšení uživatelské přívětivosti aplikace, společně s kritickým zhodnocením celé aplikace.

Použitá literatura

- [1] *What is RFID? — The Beginner's Guide to RFID Systems* [online] [cit. 2021-06-15]. Dostupné z: <https://www.atlasrfidstore.com/rfid-beginners-guide/>.
- [2] *RFID technology basics* [online] [cit. 2022-03-15]. Dostupné z: <https://rfid.averydennison.com/en/home/explore-rfid/rfid-technology-basics.html>.
- [3] *rfid4u* [online] [cit. 2022-04-20]. Dostupné z: <https://rfid4u.com/rfid-frequency/>.
- [4] *What is RFID? — The Beginner's Guide to RFID Systems* [online] [cit. 2022-03-15]. Dostupné z: <https://www.atlasrfidstore.com/rfid-insider/rssi-role-rfid>.
- [5] *Smartrac-Dogbone-Monza-4D* [online] [cit. 2022-04-22]. Dostupné z: https://www.intrasys.com.sg/Product/ViewProduct/Smartrac_Dogbone_Monza_4D.
- [6] JAN EKHOLM, Mikko Lähteenmäki. *Sample NURAPI applications for iOS* [online]. 2021-03 [cit. 2021-06-15]. Dostupné z: https://github.com/NordicID/nur_sample_ios.
- [7] *Access data offline* [online] [cit. 2022-03-29]. Dostupné z: <https://firebase.google.com/docs/firestore/manage-data/enable-offline>.
- [8] *best-firebase-alternatives* [online] [cit. 2022-04-20]. Dostupné z: <https://blog.back4app.com/best-firebase-alternatives/>.
- [9] *Different Sensor Types* [online] [cit. 2022-04-22]. Dostupné z: <https://world.wikomobile.com/a1062-different-sensor-types>.