



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**AUTOMATICKÉ ZADÁVÁNÍ DOCHÁZKY STUDENTŮM
NA VUT**

AUTOMATIC ATTENDANCE OF STUDENTS AT BUT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FREDERIK URBÁNEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ALENA TESAŘOVÁ

BRNO 2023

Zadání bakalářské práce



143877

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Urbánek Frederik**
Program: Informační technologie
Specializace: Informační technologie
Název: **Automatické zadávání docházky studentům na VUT**
Kategorie: Uživatelská rozhraní
Akademický rok: 2022/23

Zadání:

1. Analyzujte různá řešení zadávání docházky napříč fakultami VUT a prozkoumejte zájem o zautomatizování tohoto procesu.
2. Prozkoumejte možnosti automatického zadávání docházky studentům na hodině. Zvažte i využití nativních mobilních funkcí (Bluetooth, NFC, GPS) a najděte dostupné knihovny podporující jazyk React Native.
3. Z analyzovaných možností vyberte minimálně 2 a navrhnete k nim vhodná rozhraní, která budou sloužit učitelům a studentům.
4. Testujte navržená řešení a iterativně je vylepšujte.
5. Integrujte navržené řešení do aplikace Moje VUT a diskutujte výhody a omezení Vašeho řešení.
6. Vytvořte plakát nebo video reprezentující naimplementované řešení.

Literatura:

- Tidwell et al.: Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly, 2020
- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN: 978-0321657299
- Joel Marsh: UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly 2016
- React Native: <https://reactnative.dev/>
- EL-SEOUD, M. Samir Abou a Islam A. T. F. TAJ-EDDIN. Developing an Android Mobile Bluetooth Chat Messenger as an Interactive and Collaborative Learning Aid. 545. In: *Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, 2017, s. 3-15. ISBN 3319503391. ISSN 2194-5357.

Při obhajobě semestrální části projektu je požadováno:

Body 1 až 3

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

Vedoucí práce: **Tesařová Alena, Ing.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 10.5.2023
Datum schválení: 31.10.2022

Abstrakt

Problematika časovej náročnosti manuálneho zadávania dochádzky je často prehliadnutá, no veľmi dôležitá. Evidenciu dochádzky študenta je nutné zadávať ručne, či už do aplikácie alebo na papier, čo predstavuje priamu úmeru náročnosti k počtu študentov na vyučovanej hodine. Tento problém je riešiteľný pomocou moderných technológií, ktoré je možné implementovať do mobilnej aplikácie. Po dôkladnom skúmaní bolo zistené, že Bluetooth Low Energy a QR kódy sú najvhodnejšie technológie pre túto implementáciu. Vďaka výmene informácií s blízkymi zariadeniami pomocou týchto technológií sme schopní zaistiť evidenciu prítomného študenta s mobilným zariadením. Súčasťou tejto bakalárskej práce je aj funkčný prototyp implementácie automatizovanej dochádzky do mobilnej aplikácie Moje VUT.

Abstract

The time-consuming issue of manually entering attendance is often overlooked, but very important. Student attendance records must be entered manually, either in the application or on paper, which represents a direct proportion of difficulty to the number of students present in the lesson. This problem can be solved using modern technologies that can be implemented in a mobile application. After extensive research, Bluetooth Low Energy and QR codes were found to be the most suitable technologies for this implementation. Thanks to the exchange of information with nearby devices using these technologies, we are able to secure the records of the students present with a mobile device. Part of this bachelor's thesis is also a functional prototype of automated attendance implemented in the My BUT mobile application.

Kľúčové slová

Automatizovaná evidencia, Mobilná aplikácia, React Native, Typescript, JavaScript, API, QR, BLE, NFC, Moje VUT, Blízkosť zariadenia

Keywords

Automatic attendance, Mobile App, React Native, Typescript, JavaScript, API, QR, BLE, NFC, My BUT, Device proximity

Citácia

URBÁNEK, Frederik. *Automatické zadávaní dochádzky študentům na VUT*. Brno, 2023. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Alena Tesařová

Automatické zadávání docházky studentům na VUT

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Tesařovej Aleny. Další informace mi poskytli Ing. Rudolf Musil a Ing. Petr Kohut. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Frederik Urbánek
9. mája 2023

Podakovanie

Ďakujem vedúcej bakalárskej práce Ing. Alene Tesařovej za jej neustálu ochotu pomôcť a cenné rady, ktoré mi poskytla pri spracovaní mojej bakalárskej práce. Taktiež by som chcel poďakovať Ing. Petrovi Kohoutovi a Ing. Rudolfovi Musilovi za pomoc pri riešení technických problémov na strane servera a implementáciu koncových bodov.

Obsah

1	Úvod	3
2	Vývoj mobilných aplikácií	4
2.1	Natívne aplikácie	4
2.2	Hybridné aplikácie	5
2.3	Multiplatformné aplikácie	5
2.3.1	React Native	5
3	Analýza súčasného stavu	7
3.1	Zadávanie dochádzky na VUT	8
3.1.1	Papierová	10
3.1.2	Elektronická	11
3.2	Prieskum záujmu o automatizáciu	11
3.3	Analýza technológií vhodných na automatizáciu	12
3.3.1	Bluetooth	12
3.3.2	NFC	15
3.3.3	GPS	16
3.3.4	QR	17
4	Návrh riešenia	20
4.1	Výber vhodných technológií	20
4.2	Návrh funkcionality	21
4.2.1	Bluetooth Low Energy	21
4.2.2	QR kód	22
4.3	Výber vhodných knižníc	23
4.4	Návrh užívateľského rozhrania	29
4.5	Návrh rozhrania API	32
5	Implementácia	35
5.1	Moje VUT	35
5.2	Zahájenie evidencie	37
5.3	Evidencia dochádzky pomocou BLE	38
5.3.1	Obmedzenia	41
5.4	Evidencia dochádzky pomocou QR	42
6	Testovanie	44
6.1	Distribúcia aplikácie	44
6.2	Testovanie v reálnom prostredí	46

6.3	Budúci vývoj	47
6.4	Zhrnutie	49
7	Záver	50
	Literatúra	52

Kapitola 1

Úvod

V dnešnej dobe sa stretávame s neustále napredujúcimi technológiami takmer všade. Tento vývoj je veľmi dôležitý pre spoločnosť, no zároveň je vďaka nemu aj veľmi náročné udržať krok s dobou, obzvlášť vo svete informačných technológií. Všetko sa vyvíja rýchlejšie, ako kedykoľvek predtým a človek už nemusí strácať čas nazvyš na niečom, čo je možné urýchliť. V súčasnosti sa kladie veľký dôraz na zjednodušovanie a automatizáciu časovo náročných procesov, ktoré umožňujú úsporu času a sústredenie sa na činnosti s vyššou prioritou. Jednou z takýchto činností, ktorá sa málokedy automatizuje a pritom je pomerne otravná a časovo náročná, je evidencia študentov.

Napriek všetkým technológiám, ktoré sú pre nás voľne dostupné, je stále najpopulárnejšia forma evidencie papierový dochádzkový list. Tento spôsob je ale príliš starodobý a predstavuje viacero rizík. Častým problémom je strata listu, oneskorenie začiatku vyučovania kvôli dlhému procesu evidencie, alebo sfaľšovanie záznamu prítomnosti študenta, nakoľko neexistuje žiadna história zápisov a zmien. Azda najväčším problémom je prípadné porušenie GDPR, ak sa nechá list s osobnými údajmi položený v blízkosti iných študentov, ktorí ho môžu odfoťiť a následne zdieľať. Takýto incident sa už na pôde VUT raz vyskytol, preto je vhodné prijať bezpečnostné opatrenia a modernizovať spôsob evidencie dochádzky čo najskôr.

Na VUT sa už niekoľko talentovaných zamestnancov a študentov venovalo myšlienke modernizácie. V dôsledku toho bola vytvorená mobilná aplikácia Moje VUT, ktorá ponúka nielen prehľadné užívateľské rozhranie, ale aj niekoľko užitočných funkcií, medzi ktorými je aj digitálne evidovanie dochádzky. Táto funkcia umožňuje každému vyučujúcemu evidovať dochádzku študentov priamo v internom systéme VUT. Hlavnou výhodou tejto formy evidencie je bezpečnosť a odolnosť voči faľšovaniu a strate, keďže je dostupná história zmien. Hlavným problémom tejto formy evidencie je jej lineárna časová náročnosť podobná papierovej, ktorá sa zvyšuje s počtom študentov na hodine a tým znižuje efektivitu procesu zadávania dochádzky.

Cielom tejto práce je uskutočniť prieskum záujmu automatizácie evidencie dochádzky študentov a minimalizovať potrebu manuálneho vstupu od vyučujúceho využitím moderných technológií, ako sú napríklad Bluetooth Low Energy a QR kódy. Tieto technológie boli vybrané po dôkladnej analýze v sekcii 3, pretože sú najvhodnejšie pre tento účel vzhľadom na ich nízku spotrebu energie, efektivitu a jednoduché použitie v porovnaní s inými možnosťami. Implementácia tejto automatizácie umožní nielen zefektívnenie procesu evidencie dochádzky, ale aj zníženie administratívnej záťaže vyučujúcich na rôznych fakultách VUT.

Kapitola 2

Vývoj mobilných aplikácií

Mobilné aplikácie sa stávajú stále dôležitejšou súčasťou každodenného života. Sú to práve aplikácie, ktoré zjednodušujú mnohé aktivity, od sledovania videí, až po kontrolovanie výsledkov skúšky. Preto je dôležité, aby boli vývojári ohladuplní k potrebám a požiadavkám užívateľov. Vývoj mobilných aplikácií predstavuje proces, ktorý vyžaduje znalosť programovania a vývojových nástrojov pre jednotlivé operačné systémy, primárne pre Android alebo iOS. Dôležité kroky pri vývoji aplikácie sú návrh užívateľského rozhrania, návrh funkcionality, návrh architektúry a testovanie aplikácie [30]. Následne sa musí vývojár rozhodnúť, aký prístup k vývoju je najvhodnejší. Existujú tri hlavné prístupy k vývoju mobilných aplikácií [29]:

1. Natívne aplikácie
2. Hybridné aplikácie
3. Multiplatformné aplikácie

Vývojári musia tiež zohľadniť dostupné technológie v mobilných zariadeniach, ako napríklad Bluetooth alebo GPS. Rôzne zariadenia môžu mať zabudované rôzne technológie, a teda je treba mať ošetrené všetky možné variácie zariadení, aby bola aplikácia čo najviac spoľahlivá. Okrem toho musia byť aplikácie obzvlášť optimalizované, nakoľko mobilné zariadenia nemajú vysoký výpočtový výkon, musia byť schopné dynamicky zobrazovať obsah na rôznych veľkostiach obrazoviek, a zároveň podporovať rôzne verzie operačných systémov. Vývojári sa musia taktiež starať o bezpečnosť aplikácie a ochranu súkromia užívateľov. Najbežnejším spôsobom ochrany dát a súkromia je šifrovanie, autentifikácia a autorizácia [8]. Celkový vývoj mobilnej aplikácie je často veľmi časovo náročný, vyžaduje mnoho znalostí a skúseností v oblasti vývoja mobilných aplikácií a moderných vývojových technológií, a preto platí, že kvalita programátora odráža kvalitu finálneho produktu.

2.1 Natívne aplikácie

Natívne aplikácie sú vyvinuté pre konkrétne mobilné operačné systémy, ako sú Android a iOS, pomocou programovacích jazykov a nástrojov špecifických pre daný systém. Výhodou natívnych aplikácií je, že ponúkajú najvyšší výkon, pretože sú optimalizované pre danú platformu. Veľkou nevýhodou ale je, že vývoj natívnych aplikácií je často časovo náročný, najmä ak je potrebné vyvinúť aplikáciu pre viacero platforiem.

Android

Android je operačný systém pre mobilné zariadenia od spoločnosti Google [33]. Vývoj natívnych aplikácií pre platformu Android vyžaduje znalosť programovacích jazykov Java alebo Kotlin. Primárne prostredie pre vývoj je Android Studio, ktoré ponúka množstvo nápomocných nástrojov pre vývojára. Pre vývoj na platforme Android je taktiež dôležitá práca s rôznymi API. Systém Android ich poskytuje mnoho, vďaka čomu je umožnený prístup k rôznym funkciám a technológiám zariadenia, ako napríklad fotoaparát, Bluetooth a GPS. Okrem toho poskytuje aj SDK, ktoré obsahuje množstvo nástrojov na tvorbu a testovanie aplikácií.

iOS

iOS je operačný systém dostupný len pre zariadenia iPhone od spoločnosti Apple [35]. Vývoj natívnych aplikácií pre platformu iOS vyžaduje znalosť programovacieho jazyka Swift alebo Objective-C. Pre vývoj aplikácií na platforme iOS sa používa vývojové prostredie Xcode, ktoré má taktiež zabudovaných mnoho funkcií potrebných na kompletný vývoj aplikácie. Tak isto ako aj pri platforme Android, iOS má mnoho rôznych API a SDK, ktoré umožňujú prístup k funkciám zariadenia. Systém iOS je ale obecné známy ako pomerne obmedzujúci pre vývojárov, pretože rámci bezpečnostnej politiky systému je veľa funkcií obmedzených, aby sa zabezpečila väčšia úroveň zabezpečenia.

2.2 Hybridné aplikácie

Hybridné aplikácie sú vyvinuté pomocou webových technológií ako HTML, CSS a JavaScript, a následne sú tieto webové technológie zapuzdrené do natívneho kontajnera pre danú platformu. Výhodou hybridných aplikácií je, že sú ľahšie prenositeľné medzi platformami a zároveň ponúkajú vyšší výkon a užívateľskú skúsenosť ako klasické webové aplikácie. Nevýhodou je, že vývoj hybridných aplikácií môže byť zložitý, najmä ak sú potrebné pokročilé funkcie, ktoré sú špecifické pre danú platformu.

2.3 Multiplatformné aplikácie

Multiplatformné aplikácie sú vyvinuté pomocou jedného kódu a sú následne preložené pre rôzne platformy. Existuje mnoho multiplatformných frameworkov ako sú React Native, Flutter alebo Xamarin, ktoré umožňujú vývoj mobilných aplikácií s použitím jedného kódu pre rôzne platformy [39]. Výhodou multiplatformných aplikácií je, že umožňujú rýchlejší a efektívnejší vývoj pre viacero platforiem s menšími nákladmi. Nevýhodou môže byť horší výkon v porovnaní s natívnymi aplikáciami.

2.3.1 React Native

React Native je framework s otvoreným zdrojovým kódom na vytváranie multiplatformových mobilných aplikácií pre Android a iOS pomocou knižnice React a jazykom JavaScript/Typescript [24, 20]. Bol vytvorený spoločnosťou Facebook v roku 2015 a odvtedy sa stal veľmi populárnym v vývojárskej komunite. Aby sa React Native aplikácia mohla preložiť na kód pre konkrétnu platformu, je potrebné použiť takzvané “bridge”. Bridge je softvérová vrstva zabudovaná priamo v frameworku, ktorá prekladá kód z JavaScriptu alebo Typescriptu na natívny kód pre danú platformu.

Vývoj mobilných aplikácií môže byť uskutočnený v rôznych vývojových prostrediach, ako je napríklad Expo, ktorý uľahčuje prácu pri prvotnom nastavovaní a testovaní projektu, alebo React Native CLI, ktorý je zabudovaný priamo do frameworku. Pri vývoji je dôležité dbať na kvalitu kódu a testovanie. Existuje množstvo nástrojov a technológií, ktoré pomáhajú vývojárom dosiahnuť tieto ciele, ako je napríklad Redux pre správu stavov a Jest pre testovanie kvality kódu.

Výhody

React Native má niekoľko výhod, ktoré robia tento framework veľmi obľúbený pre vývoj mobilných aplikácií. Jednou z týchto výhod je schopnosť vytvárať aplikácie s podobným vzhľadom a funkčnosťou ako natívne aplikácie. Navyše používa podobné značky ako React pre webové aplikácie, čo znamená, že vývojári nemusia učiť nový jazyk pre prácu v tomto frameworku. Okrem toho sú aplikácie veľmi udržateľné, pretože knižnice pre tento framework používajú zdroje s otvoreným zdrojovým kódom, ktoré sú udržiavané a aktualizované komunitou vývojárov. To znamená, že vývojári nemusia platiť za licencie a môžu používať existujúce zdroje pre vývoj aplikácií, čo znižuje náklady a zvyšuje efektívnosť. Ďalšou výhodou je funkcia "Hot Reload", ktorá umožňuje okamžite vidieť zmeny, ktoré sa vykonali v kóde po uložení. Táto funkcia zvyšuje produktivitu vývojárov tým, že umožňuje rýchlo testovať a zisťovať chyby v aplikácii. Navyše umožňuje vývojárom rýchlo iterovať a prispôbovať aplikáciu podľa potrieb užívateľa.

Hlavným princípom React Native je sústrediť sa na stav komponentov. Stav predstavuje dáta, ktoré sa zobrazujú v aplikácii a menia sa v závislosti od akcií užívateľa. Na spravovanie a ukladanie stavov sa používajú knižnice **Redux** alebo **MobX**, ktoré integrujú centrálnu úložisko (store) priamo v mobilnom zariadení. Na interakciu s jednotlivými prvkami v aplikácii sa používa **Virtual DOM**. Je to abstraktná reprezentácia DOM-u (Document Object Model), ktorý je spravovaný Reactom a používa sa na efektívne upravovanie DOM-u bez nutnosti jeho opakovaného prekresľovania. Keď sa zmení stav komponenty, React tento DOM prekreslí a porovná ho s pôvodným. Ak sa niečo zmenilo, React upraví len zmenené časti, čím zlepšuje výkon a minimalizuje zbytočné prekreslenia aplikácie. Pre popis užívateľského rozhrania a jeho rozmiestnenia sa používa syntaktické rozšírenie JavaScriptu **JSX**.

Komponenty

React Native aplikácie sa skladajú z komponentov, ktoré sú základnou stavebnou jednotkou pre tvorbu užívateľského rozhrania. Každý komponent obsahuje definíciu vlastností (props) a stavov (state). Stav komponentov môže byť aktualizovaný pomocou vstupných hodnôt alebo pri interakcii s užívateľom, čo zabezpečuje reaktívnosť zobrazovaných elementov a dát. Každý komponent je navrhnutý tak, aby bol znovupoužiteľný. Teda, skladaním základných komponentov môžeme vytvárať komplexné a viacúrovňové komponenty. V React Native sú k dispozícii mnohé vstavané komponenty ako napríklad Text, Image, View, ScrollView, FlatList, alebo TextInput. Okrem toho, existuje aj veľa volnodostupných knižníc s množstvom ďalších komponentov s rôznymi funkcionalitami.

Kapitola 3

Analýza súčasného stavu

Evidencia dochádzky je dôležitou súčasťou vysokých škôl, ktorá zabezpečuje informáciu o návštevnosti na danú prednášku alebo cvičenie. Vďaka evidencii môžu dostať študenti body, alebo dokonca aj zápočet, preto je veľmi dôležité, aby bola evidencia dochádzky čo najspoľahlivejšia. Zároveň je dôležité, aby evidencia dochádzky netrvala príliš dlho, inak len zatažuje vyučujúceho, čo má za následok menej času na vyučovanie. V súčasnej dobe existuje niekoľko spôsobov, ako evidenciu dochádzky na vysokých školách realizovať. Tieto spôsoby sa obecné rozdelujú na:

1. papierová evidencia
2. elektronická evidencia

Každá má však svoje výhody a nevýhody, a teda je dôležité zhodnotiť, ktorá je najprospešnejšia pre špecifickú vyučovaciu hodinu.

Najväčší problém pri evidencii formou papierovej dochádzky je možnosť falšovania záznamu. Študenti si môžu navzájom podpísať dochádzku alebo použiť podpisy iných študentov na nej, čím vytvoria falošný záznam o dochádzke. Preto sa v súčasnosti čoraz častejšie používa elektronická evidencia dochádzky, ktorá sa môže realizovať pomocou rôznych technológií. Tieto riešenia môžu nahrnúť využitie bezkontaktných študentských kariet ISIC, QR kódov alebo aplikácií pre mobilné zariadenia.

Bezkontaktné karty ISIC sa používajú tak, že študent pri vstupe do budovy priloží svoju kartu na čítačku, ktorá zaznamená jeho prítomnosť. Tento spôsob je považovaný za spoľahlivý a rýchly, no je potrebné mať k dispozícii dostatočné množstvo čítačiek. Ak náhodou prestanú čítačky fungovať, je to problém ktorý je potrebné vyriešiť inou formou evidencie. Stále tu ale existuje riziko falšovania záznamu dochádzky, keďže študent môže priložiť aj kartu svojho kamaráta, ktorý sa na vyučovanej hodine nenachádza.

Ďalším moderným spôsobom evidencie sú QR kódy. Na vysokých školách sa používajú tak, že študenti pri vstupe do budovy naskenujú QR kód, ktorý sa nachádza na vchodových dverách alebo v učebniach [23]. Tento spôsob je rýchly a jednoduchý, no vyžaduje, aby študenti mali k dispozícii zariadenie s kamerou a vhodnú aplikáciu, ktorá umožňuje skenovanie QR kódu. Zároveň je potreba tento kód buď nalepiť na každú učebňu, kde sa koná vyučovacia hodina, alebo ho zobrazíť pomocou projektoru, ktorý ale nemusí byť dostupný v každej učebni.

Celkovo možno povedať, že evidencia dochádzky na vysokých školách stále trpí rôznymi nedostatkami a problémami. Avšak s využitím moderných technológií a inovatívnych riešení sa táto oblasť postupne modernizuje a zlepšuje.

3.1 Zadávanie dochádzky na VUT

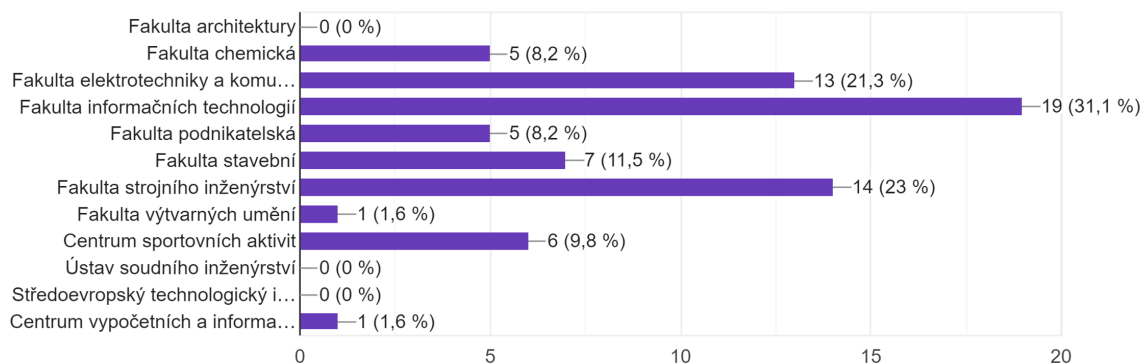
S cieľom automatizovať a optimalizovať proces evidencie dochádzky na VUT bolo nevyhnutné získať relevantné dáta týkajúce sa aktuálne používanej formy evidencie, záujmu o automatizáciu a prípadných požiadavkov na implementáciu. Na získanie týchto dát bol rozposlaný dotazník **219** zamestnancom z rôznych fakúlt a iných súčastí VUT. Počet odpovedí dosiahol **61**. Dotazník má celkovo 8 detailne popísaných otázok zhrnutých do 4 častí, ktoré slúžia na získanie všetkých potrebných dát o evidencii dochádzky na VUT.

Otázky položené v dotazníku pre zamestnancov VUT:

- Postavenie zamestnanca
- Stav súčasných foriem evidencie
- Záujem o automatizáciu
- Návrh na implementáciu a vylepšenie

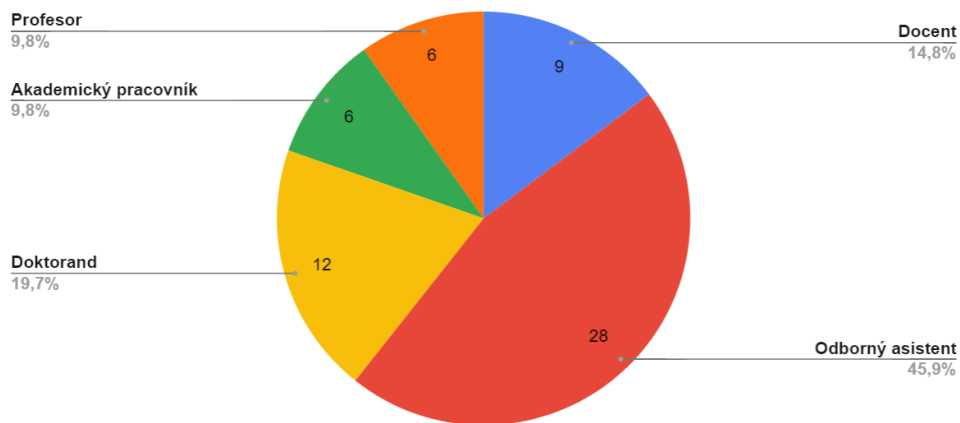
Štatistiky postavenia zamestnanca

V úvodnej časti dotazníku boli položené otázky na postavenie zamestnanca VUT, a to s cieľom získať prehľad o jeho pôsobisku na fakultách.



Obr. 3.1: Fakulty, na ktorých zamestnanec VUT vyučuje

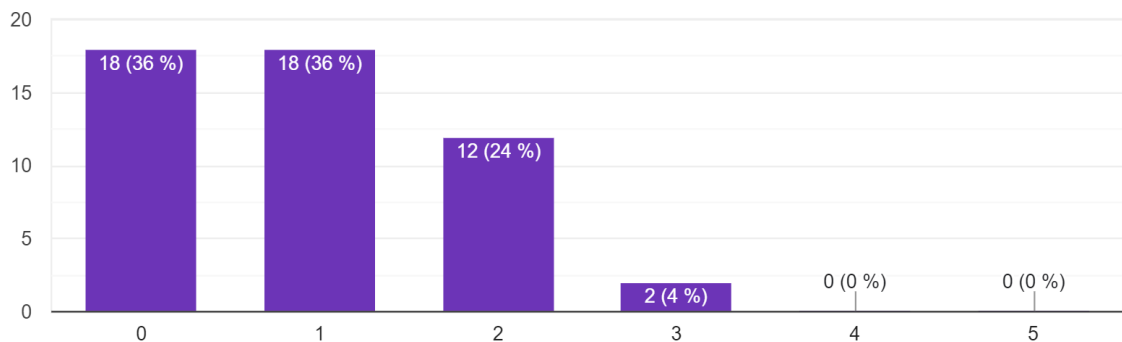
Ako je možné vidieť na grafe 3.1, z výsledkov dotazníka vyplýva, že väčšina odpovedí je od zamestnancov fakúlt, ktorí sú primárne orientovaní na IT a iné technické odvetia, a to najmä z fakúlt informačných technológií, elektrotechniky a komunikačných technológií, a strojného inžinierstva. Na základe týchto dát možno predpokladať, že záujem o automatizáciu evidencie dochádzky bude najväčší práve na týchto fakultách.



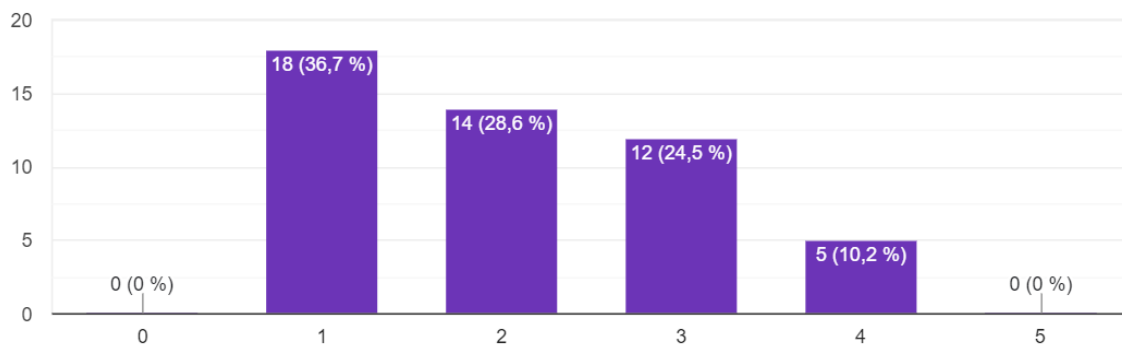
Obr. 3.2: Pozícia zamestnanca VUT

Štatistiky stavu súčasných foriem evidencie

Následujúca časť dotazníka bola zameraná na súčasnú formu evidencie dochádzky, s dôrazom na časovú náročnosť a namáhavosť tohto procesu. Zároveň je položená otázka ohľadom záujmu o automatizáciu tohto procesu. Hodnotenie časovej náročnosti a namáhavosti prebieha na stupnici od **0-5**, pričom 0 reprezentuje absenciu náročnosti a namáhavosti.



Obr. 3.3: Časová náročnosť súčasných foriem evidencie



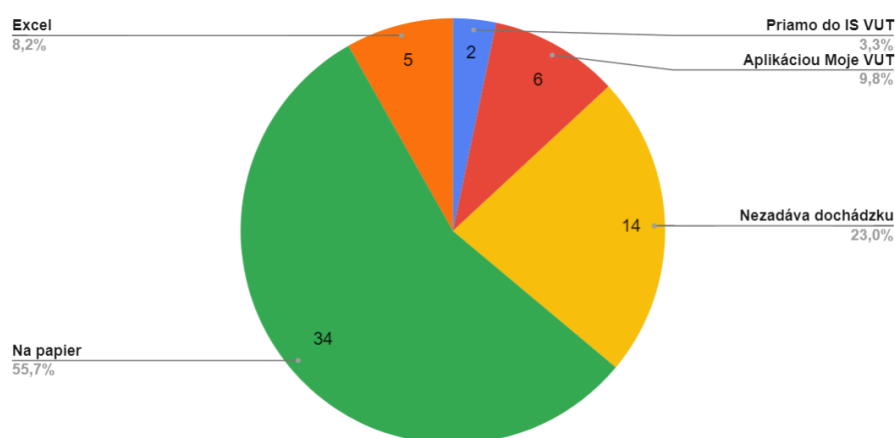
Obr. 3.4: Namáhavosť súčasných foriem evidencie

Po zhodnotení grafov 3.3 a 3.4 je evidentné, že namáhavosť a náročnosť aktuálnych foriem evidencie dochádzky študentov nie sú v kritickom stave. Avšak, výsledky z dotazníka poukazujú na veľký potenciál v oblasti zjednodušenia a zefektívnenia práce vyučujúceho. Z získaných dát je zrejmé, že priemerná úroveň namáhavosti súčasnej evidencie je vyššia, čo môže negatívne ovplyvniť priebeh výučby.

V rámci prieskumu boli získané aj štatistiky o aktuálne používaných formách evidencie dochádzky študentov. Tieto informácie poskytujú dôležitý prehľad o súčasnom stave evidencií, čo umožňuje viac zefektívniť návrhy a finálne riešenie implementácie automatizovanej dochádzky.

3.1.1 Papierová

Jedným z najbežnejších spôsobov je manuálna evidencia dochádzky, kedy študenti ručne podpisujú svoju dochádzku v dochádzkovej listine, ktorá je následne spracovávaná vyučujúcim na konci vyučovania. Tento spôsob evidencie dochádzky je však časovo náročný, náchylný na chyby a sfalšovanie, čo vedie k mnohým nepresnostiam a problémom. Pri tejto variante je si treba dávať pozor na porušenie **GDPR**¹. Ďalšia alternatíva k papierovej dochádzkovej listine je taká, kde si vyučujúci sám poznačuje študentov, ktorý sú prítomný na hodine. Tento spôsob je o niečo bezpečnejší, no stále je možné papier stratiť alebo poškodiť, čo môže spôsobiť nežiadane problémy pre vyučujúceho aj pre študentov. Výhodou ale je, že si môže vyučujúci poznačiť, kde študenti sedia, a teda si nemusí pamätať ich mená, čo ulahčí prácu hlavne pri testovaní vedomostí študentov.



Obr. 3.5: Štatistika spôsobov evidencie dochádzky na VUT

Z grafu 3.5 vyplýva, že až 55% respondentov preferuje papierovú listinu ako formu evidencie dochádzky. Ak zanedbáme skupinu vyučujúcich (23%), ktorí neevidujú dochádzku vôbec, finálna hodnota je **72%**, čím sa len potvrdzuje evidentne preferovaná forma papierovej evidencie. Hlavnou príčinou tohto preferovania je dlhodobá prax vyučujúcich, ktorí sú s touto formou evidencie dlhodobo oboznámení. Napriek tomu, že sa jedná o overený spôsob, existujú inovatívnejšie a efektívnejšie metódy, ktoré by mohli nahradiť papierovú evidenciu na VUT, a tým zabezpečiť bezpečnejšiu a efektívnejšiu formu evidovania študentov. Avšak,

¹GDPR (Všeobecné Nariadenie o Ochrane Osobných Údajov) je európska legislatíva, ktorá určuje pravidlá pre spracovanie a ochranu osobných údajov občanov EÚ.

na zavedenie nového systému evidencie dochádzky, bude potrebné vytvoriť aplikáciu, ktorá bude intuitívna a jednoduchá na používanie, inak sa táto implementácia nemusí ujať.

3.1.2 Elektronická

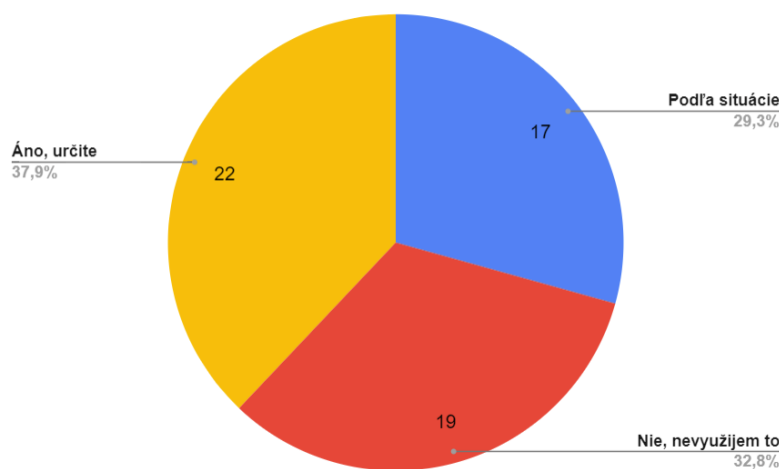
Elektronická evidencia dochádzky je v súčasnosti čoraz častejšou formou evidencie dochádzky na vysokých školách. Vyučujúci používajú rôzne formy aplikácií a nástrojov na zjednodušenie a zefektívnenie procesu evidencie. Podľa prieskumu uvedeného v grafe 3.5 je zrejmé, že iba 21% vyučujúcich používa nejakú formu elektronickej evidencie študentov, či už zosiť Excel alebo mobilnú aplikáciu Moje VUT. Ak znova zanedbáme skupinu vyučujúcich (23%), ktorí neevidujú dochádzku, získame výslednú štatistiku 28% vyučujúcich, ktorí evidujú dochádzku elektronicke.

Elektronická evidencia dochádzky má mnoho výhod. Okrem zvýšenia efektivity a presnosti, táto forma evidencie umožňuje aj rýchly a jednoduchý prístup k záznamom a histórii zmien dochádzky študentov. V prípade potreby sa dá rýchlo zistiť, ktorí študenti boli v triede prítomní a ktorí nie, a to aj na diaľku z hocijakého múdreho zariadenia, ktoré je schopné sa pripojiť na internet, čo môže byť v súčasnej dobe veľmi užitočné. Navyše, ak je evidencia uskutočnená pomocou informačného systému VUT alebo mobilnej aplikácie Moje VUT, sú tieto dáta bezpečne uložené na servery, kde ich nieje možné stratiť.

Napriek tomu, že elektronická evidencia dochádzky ponúka mnoho výhod, niektorí vyučujúci stále preferujú papierovú formu evidencie, pretože im je bližšia a ovládateľnejšia. Aj keď je elektronická forma zatiaľ menej rozšírená, očakáva sa, že s narastajúcim záujmom a vývojom technológií sa bude táto forma stávať čoraz populárnejšou.

3.2 Prieskum záujmu o automatizáciu

Po zistení, že papierová forma evidencie je dominantný spôsob evidencie dochádzky na fakultách VUT, sa stáva dôležitým skúmať záujem vyučujúcich o automatizáciu elektronickej formy evidencie. Aj keď existuje predpoklad menšieho záujmu z radov starších profesorov, ktorí sa možno nechcú vzdať papierovej formy, stále existujú vyučujúci, ktorí radi uvítajú nové technológie, ktoré im nejakým spôsobom zjednodušia prácu, a sú ochotní takúto modernizáciu otestovať a využiť v praxi.



Obr. 3.6: Záujem o automatizáciu evidencie dochádzky

Na základe získaných dát z grafu 3.6 môžeme skonštatovať, že približne **38%** zamestnancov VUT prejavilo aktívny záujem o automatizáciu evidencie dochádzky študentov, pričom ďalších **29%** uviedlo, že by boli ochotní túto modernizáciu využiť v závislosti od konkrétnej situácie a vyučovanej hodiny. Teda môžeme s istotou povedať, že až **67%** zamestnancov, ktorí sa rozhodli na túto problematiku reagovať, by bolo ochotných túto automatizáciu využiť na svojich hodinách. Získané dáta teda naznačujú vysokú úroveň záujmu o modernizáciu evidencie dochádzky, čo je skvelá správa pre budúcnosť vývoja aplikácie Moje VUT.

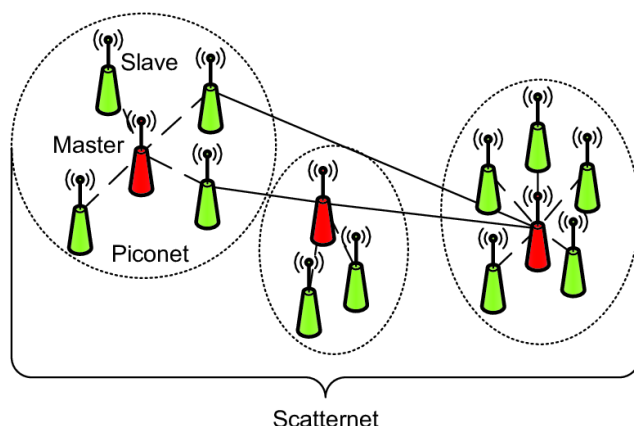
3.3 Analýza technológií vhodných na automatizáciu

V rámci prieskumu boli získané nielen informácie o záujme vyučujúcich o automatizáciu dochádzky, ale aj cenné poznatky o ich predstavách týkajúcich sa implementácie a fungovania automatizovanej dochádzky. Aby bola aplikácia implementovaná s očakávanou efektivitou a splnila očakávania užívateľov, je kľúčové vykonať dôkladnú analýzu existujúcich technológií a vybrať tie najvhodnejšie pre tento účel. Nesprávne alebo nedostatočné využitie technológií by mohlo mať negatívny vplyv na záujem o použitie aplikácie a viesť k nezáujmu zo strany užívateľov. Cieľom tejto analýzy je dosiahnuť čo najlepšie výsledky, ktoré prinesú zvýšenie efektivity procesu evidencie a zároveň zvýšia záujem o aplikáciu Moje VUT aj medzi tými, ktorí ju ešte nevyužívajú. Pre dosiahnutie kvalitnej implementácie je nevyhnutné analyzovať, akým spôsobom bude možné overiť prítomnosť študenta na vyučovanej hodine a zároveň zabezpečiť, aby takéto overenie nebolo ľahko sfaľovateľné. Kvôli tomu je nevyhnutné zvoliť vhodnú technológiu, ktorá bude schopná buď spoľahlivo identifikovať lokalizáciu študenta, alebo jeho prítomnosť v blízkosti vyučujúceho.

3.3.1 Bluetooth

Bluetooth je bezdrôtová technológia, ktorá umožňuje prenos dát medzi zariadeniami v blízkosti pomocou rádiových vlnových signálov. Bluetooth využíva frekvenciu 2,4 GHz, vďaka čomu je signál schopný preniknúť stenami, pričom ale dosah signálu závisí od vysielacieho zariadenia, prekážok v ceste a prijímacieho zariadenia [6]. Teoretický limit dosahu je 100-400 metrov, pričom v praxi sa toto číslo pohybuje na okolo 20-100 metroch. Bluetooth sa používa napríklad na bezdrôtové pripojenie k slúchadlám, reproduktorom, klávesniciam a myšiam, alebo na prenos súborov medzi dvoma a viacerými zariadeniami. V rámci Bluetooth existuje viacero profilov, ktoré popisujú, ako môže byť Bluetooth použitý [5]. Tieto profily zahŕňajú napríklad prenos súborov, riadenie zvuku, ovládanie zariadení alebo prenos dát s nízkym výkonom (BLE). Bluetooth umožňuje aj vytváranie sietí ako Piconet a Scatternet, ktoré umožňujú komunikáciu medzi viacerými zariadeniami v rámci vytvorenej siete [9].

Bluetooth umožňuje dve základné formy komunikácie: asynchrónnu prenosovú službu (ACL) a synchronizačnú prenosovú službu (SCO). ACL slúži na prenos menších dát, ako napríklad hudbu, SCO slúži na prenos hlasu, alebo iných dát vo vysokej kvalite. Umožňuje pripojenie až s 8 zariadeniami naraz, pričom každé zariadenie môže byť buď v režime Master alebo Slave. Zariadenie v režime Master je zodpovedné za riadenie celého prenosu dát, zatiaľ čo zariadenie v režime Slave iba prijíma dáta. Komunikácia medzi Master a Slave zariadeniami prebieha pomocou párovacieho procesu, ktorý zabezpečuje bezpečné spojenie medzi zariadeniami. Prenos dát je taktiež možný aj bez párovania medzi zariadeniami, no súvisí s tým bezpečnostné riziko, preto je dôležité zabrániť prenosu citlivých informácií bez predchádzajúceho párovania.



Obr. 3.7: Ukážka prepojených zariadení pomocou Bluetooth, obrázok prevzatý z [40]

Bluetooth Low Energy

Bluetooth Low Energy (**BLE**) je bezdrôtový komunikačný protokol, ktorý slúži na prenos dát medzi rôznymi zariadeniami. BLE bol vyvinutý v roku 2010 ako nový protokol technológie Bluetooth, s cieľom poskytnúť nízkoenergetické riešenie pre zariadenia s obmedzenou batériou. Je súčasťou každého zariadenia, ktorý má Bluetooth 4.0 a vyššie [3, 15]. BLE využíva rovnakú frekvenciu ako Bluetooth, avšak jeho dosah je oveľa kratší. Prenosová rýchlosť je obmedzená na 200 Kb/s a jeho teoretický dosah je okolo 60 metrov, v skutočnosti je to však okolo 20 metrov, z dôvodu prekážok a iných zariadení, ktoré môžu vytvárať interferenciu. Táto technológia využíva len malé množstvo energie, a preto je vhodná na komunikáciu s zariadeniami, ktoré nevyžadujú veľké množstvo prenesených dát.

BLE je obzvlášť populárne v oblasti mobilných zariadení. Mobilné zariadenia často využívajú túto technológiu pre pripojenie k rôznym zariadeniam, ako napríklad k bezdrôtovým slúchadkám. Využitie BLE v mobilných zariadeniach umožňuje vývojárom vytvárať nové aplikácie a služby, ktoré sú využívané v mnohých oblastiach, napríklad v zdravotníctve, fitness, alebo inteligentných domácnostiach.

Bluetooth Low Energy má takzvané komunikačné kanály, ktoré umožňujú výmenu dát medzi zariadeniami s nízkou spotrebou energie [38]. Existujú tri základné typy komunikačných kanálov:

1. **Advertising kanál** - Tento kanál sa používa pre nadviazanie spojenia a zahájenie prenosu dát medzi zariadeniami. Zariadenie, ktoré chce prijať dáta vysielané na Advertising kanáli, sa označuje ako Central, a zariadenie, ktoré tieto dáta vysielajú, sa označuje ako Peripheral. Informácie sú odosielané v nepravidelných intervaloch a môžu obsahovať napríklad názov zariadenia, typ zariadenia alebo MAC adresu² zariadenia. Advertising kanály sa skladajú z troch rôznych kanálov s frekvenciami 2402 MHz, 2426 MHz a 2480 MHz.
2. **Connection-oriented kanál** - Tieto kanály sú používané na výmenu dát medzi Central a Peripheral zariadeniami, ktoré už boli pripojené pomocou Advertising kanála. Existujú dva typy connection-oriented kanálov - ATT (Attribute Protocol) a L2CAP

²MAC adresa (Media Access Control address) je unikátny identifikátor zariadenia, ktorý je pridelený sieťovému adaptéru a slúži na jeho identifikáciu v sieti.

(Logical Link Control and Adaptation Protocol). ATT kanál sa používa pre prenos dát s vysokou spoľahlivosťou a L2CAP kanál sa používa pre prenos dát s menšou odozvou, ale nízkou spoľahlivosťou. Po vytvorení spojenia sa dáta vymieňajú v rámci data kanálu, ktorý môže byť jednoduchý (1:1) alebo mnohoznačný (1:N). Pri jednoduchom kanáli je vždy jedno zariadenie pripojené k druhému, zatiaľ čo pri mnohoznačnom kanáli môže byť jedno zariadenie pripojené k viacerým zariadeniam súčasne.

3. **GATT (Generic Attribute Profile) kanál** - Tento kanál poskytuje štruktúru pre definovanie a výmenu dátových profilov medzi dvoma zariadeniami. GATT kanál je založený na ATT kanáli a používa ho pre výmenu dát.

Celkovo teda existuje až 40 rôznych BLE komunikačných kanálov, z toho 3 kanály sú určené pre Advertising, 37 kanálov sú connection-oriented a jeden kanál slúži pre GATT. Každý z týchto kanálov má svoje špecifické použitie a parametre, ktoré je možné konfigurovať. Pri súbežnej komunikácii viacerých zariadení je možné preťažiť tieto kanály, a preto je nevyhnutné riadiť objem prenášaných dát, aby nedochádzalo k interferencii a zpomaleniu BLE komunikácie medzi zariadeniami.

Z dôvodu vysokého počtu mobilných zariadení, ktoré disponujú technológiou Bluetooth, jej spoľahlivosti a dosahu pre prenos dát je možné usúdiť, že je táto technológia vhodná pre účel implementácie automatizovanej evidencie dochádzky študentov v mobilnej aplikácii Moje VUT.

Google Nearby Messages API

Google Nearby Messages API je služba od spoločnosti Google, ktorá umožňuje mobilným zariadeniam komunikovať medzi sebou pomocou technológie BLE (popísanej v 3.3.1). Funguje na princípe modelu **Publisher-Subscriber**, kde Publisher zariadenie dáta vysiela a Subscriber zariadenie ich prijíma [13]. Funguje tak, že Publisher zariadenie odošle dáta na najbližší dostupný Google server a následne vysiela párovací token pomocou BLE ostatným zariadeniam v okolí. Keď nejaké zariadenie tento token prijme, overí ho voči Google serveru a následne môže stiahnuť dáta, čím sa ukončí proces komunikácie medzi mobilnými zariadeniami.

Táto služba je dostupná na platformách Android a iOS. V Android systémoch je táto služba dostupná v *Google Play Services*, čo znamená, že každé Android mobilné zariadenie ju má k dispozícii bez potreby mať aktívny Google účet. Naopak, v systéme iOS je nutné mať nainštalovaný *Google Play Services*, ale ak má užívateľ inú aplikáciu od Google nainštalovanú, napríklad Gmail, tak už má túto službu dostupnú. Nearby Messages využíva zabudované algoritmy, ktoré zaručujú spoľahlivé a efektívne prenášanie dát medzi mobilnými zariadeniami, a teda vývojár sa môže zamerať na vývoj aplikácie, namiesto spravovania a optimalizovania komunikácie medzi zariadeniami. Dáta sú medzi zariadeniami prenášané vo forme čistého textu bez šifrovania, preto je dôležité neprenášať citlivé dáta pomocou tejto služby.

Vzhľadom na to, že služba Google Nearby Messages API je už dostupná v mnohých zariadeniach a jej spoľahlivosť v prenose dát je vysoká, je táto služba vhodná na implementáciu automatizovanej evidencie dochádzky študentov v mobilnej aplikácii Moje VUT.

Google Nearby Connections API

Google Nearby Connections API je služba, ktorá umožňuje mobilným zariadeniam vytvárať *peer-to-peer*³ spojenia bez použitia internetu. Spojenia môžu byť založené na rôznych technológiách, ako je Wi-Fi Direct alebo **Bluetooth Low Energy** [12, 4]. Funguje podobne ako Nearby Messages (3.3.1) s tým rozdielom, že nevyžaduje spojenie s centralizovaným Google serverom. Vytvorenie spojenia prebieha pomocou modelu **Endpoint Discovery**, ktorý umožňuje zariadeniam v okolí nájsť sa navzájom. Zariadenia sa môžu zoskupovať do skupín, ktoré umožňujú zdieľanie dát medzi zariadeniami v rámci skupiny a poskytujú vývojárom väčšiu kontrolu nad spojením. Google Nearby Connections umožňuje vytvoriť robustné spojenia medzi mobilnými zariadeniami aj v prostrediach s nízkou kvalitou signálu a s vysokou mierou interferencie.

Keďže služba Google Nearby Connections API je taktiež dostupná v mnohých zariadeniach a jej spoľahlivosť v prenose dát je rovnako vysoká ako pri Nearby Messages (3.3.1), je táto služba vhodná na implementáciu automatizovanej evidencie dochádzky študentov v mobilnej aplikácii Moje VUT. Žiaľ, pre túto technológiu aktuálne neexistujú žiadne udržiavané React Native knižnice, a teda integrácia tejto technológie do aplikácie Moje VUT zatiaľ nie je možná.

3.3.2 NFC

NFC (Near Field Communication) je bezdrôtová technológia, ktorá umožňuje komunikáciu medzi zariadeniami vo veľmi malej vzdialenosti (do 4cm). Táto technológia umožňuje bezdotykovú výmenu dát medzi dvomi zariadeniami prostredníctvom elektromagnetického poľa. NFC je založené na technológii RFID (Radio-Frequency Identification), avšak oproti RFID technológii umožňuje NFC dvojstrannú komunikáciu [36]. To znamená, že cez NFC je možné nielen prijímať, ale aj odosielať dáta.

Existujú dva typy NFC zariadení, aktívne a pasívne. Aktívne NFC zariadenia majú vlastný zdroj napájania, čo umožňuje vysielanie dát a podporu rôznych funkcií. Pasívne NFC zariadenia nemajú vlastný zdroj napájania, a sú závislé na elektromagnetickej sile generovanej aktívnym NFC zariadením. Pasívne NFC zariadenia, taktiež známe ako NFC tagy, sa používajú na výmenu dát a interakciu s aktívnymi NFC zariadeniami.

Nevýhodou tejto technológie je fakt, že jej implementácia v mobilných zariadeniach môže byť pomerne zložitá. Navyše, nie každé zariadenie disponuje technológiou NFC, čo môže obmedziť použiteľnosť tejto funkcionality. Okrem toho sú tu aj veľké problémy s konzistenciou dosahu, ktoré sa môžu vyskytnúť pri starších zariadeniach alebo pri použití telefónov s blokujúcim krytom. Veľmi významným problémom je aj komunikácia medzi dvoma NFC zariadeniami, a to najmä v prípade systému iOS, ktorý neumožňuje emuláciu tagových kariet (HCE). V dôsledku toho nie je možné prenášať údaje z zariadenia iPhone na iné aktívne zariadenia, teda iPhone môže slúžiť iba ako aktívny čítač.

³Peer-to-peer je sieťový model, v ktorom zariadenia komunikujú a vymieňajú si dáta priamo medzi sebou, bez centrálného serveru.

Platforma	Aktívny čítač NFC	Pasívny tag NFC
Android	Áno	Áno (HCE)
iOS	Áno	Nie

Tabuľka 3.1: Podpora NFC v mobilných operačných systémov

HCE

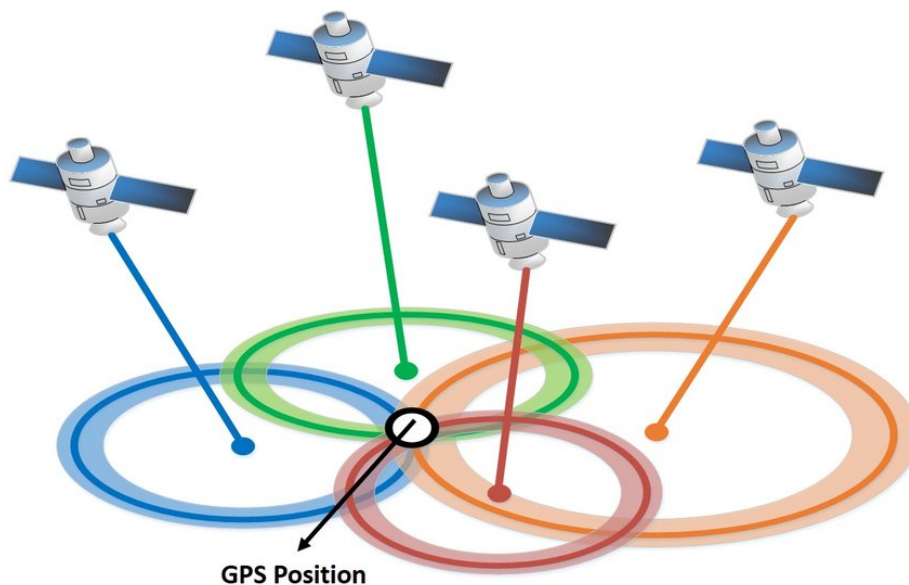
HCE (Host Card Emulation) je technológia v systéme Android, ktorá umožňuje emulovať NFC tagové karty v softvérovej forme [2]. HCE umožňuje aplikáciám vykonávať bezkontaktné platby, napríklad cez Google Pay, alebo overiť identitu pri vstupe do budovy bez nutnosti fyzickej karty. HCE využíva bezpečnostné prvky systému Android, aby virtuálna karta pôsobila rovnako, ako fyzická. Tieto bezpečnostné prvky zahŕňajú napríklad **Secure Element**, ktorý zabezpečuje uchovávanie kryptografických kľúčov a iných citlivých informácií. HCE využíva softvérový Secure Element, čo znamená, že tieto kľúče sú uchovávané v softvérovej forme v pamäti zariadenia.

Z dôvodu obmedzeného dosahu komunikácie medzi zariadeniami, by NFC nebolo optimálnou voľbou pre automatizáciu evidencie dochádzky. V tomto prípade by každý študent musel priložiť svoje zariadenie k mobilnému zariadeniu učiteľa, čo by bolo z praktického hľadiska nevhodné a mohlo by to spôsobiť čakacie fronty a technické problémy. Avšak, táto technológia môže byť použitá ako alternatívna možnosť v prípade, že primárna technológia zlyhá, a teda je možné ju využiť ako sekundárnu funkcionálnosť pre automatizáciu evidencie dochádzky.

3.3.3 GPS

GPS (Global Positioning System) je globálny družicový navigačný systém, ktorý umožňuje vypočítať polohu s vysokou presnosťou na základe prijatých signálov od viacerých družíc. Systém GPS pozostáva z 31 satelitov umiestnených na obežnej dráhe okolo Zeme, kontrolných staníc na Zemi a GPS prijímačov v rôznych zariadeniach, ako sú napríklad automobily alebo mobilné zariadenia [34]. Každý GPS satelit vysiela signál, ktorý obsahuje informácie o jeho presnej polohe a čase vysielania signálu, zatiaľ čo prijímač zachytáva tieto signály od viacerých satelitov a na základe týchto informácií vypočíta svoju polohu. Tento proces sa nazýva trilaterácia⁴ (3.8) a je potrebné získať údaje od aspoň štyroch družíc, aby bola dosiahnutá prijateľná presnosť, ktorá je v priemere okolo 5 metrov. Mobilné zariadenia využívajú GPS na rôzne účely, ako napríklad navigácia alebo lokalizácia aktuálnej polohy. Aplikácie v mobilných zariadeniach môžu využívať GPS z dôvodu poskytovania relevantných informácií, napríklad miesta záujmu v okolí.

⁴Trilaterácia je proces určenia polohy bodu v priestore na základe známych polôh a vzdialeností od tohto bodu k štvorici bodov.



Obr. 3.8: Trilaterácia GPS signálov za účelom získania presnej polohy zariadenia. Obrázok prevzatý z [10]

Aj keď je systém GPS významným a praktickým nástrojom, ktorý umožnil celosvetovú modernizáciu, taktiež trpí rôznymi obmedzeniami. Hlavným problémom je obmedzená dostupnosť signálu v niektorých oblastiach, ako sú napríklad tunely alebo výškové budovy. Okrem toho môže spôsobovať vysokú spotrebu batérie a je náchylný na nežiadajúcu interferenciu, ktorá môže negatívne ovplyvniť jeho presnosť. Ďalším problémom je relatívne jednoduché falšovanie polohy pomocou rôznych aplikácií, ktoré môžu ovplyvňovať údaje modulu, a tým klamať o užívateľovej skutočnej polohe. Okrem toho môže v mobilných aplikáciách využívajúcich GPS nastávať problém s GDPR, ak nie je spracovanie lokalizačných dát dostatočne transparentné a zabezpečené.

V kontexte automatizácie evidencie dochádzky študentov, technológia GPS môže z prvého pohľadu pôsobiť ako užitočný a vhodný nástroj. Avšak, existujú problémy, ktoré výrazne obmedzujú jeho využitie. Jedným z týchto problémov je obmedzená presnosť a dostupnosť signálu GPS v podzemných alebo výškových budovách, čo by mohlo spôsobiť vznik nepresností a negatívne ovplyvniť efektívnosť overovania evidencie. Okrem toho existuje možnosť oklamať polohu GPS pomocou rôznych aplikácií, ktoré sú bežne dostupné v digitálnom obchode Google Play pre platformu Android. Týmito aplikáciami by sa študenti mohli dopracovať k simulovaniu svojej polohy na vyučovanej hodine, aj keď sa tam v skutočnosti nenachádzajú. Preto je využiteľnosť technológie GPS v oblasti automatizácie evidencie dochádzky nízka a nie je považovaná za ideálny nástroj na tento účel.

3.3.4 QR

QR (Quick Response) je technológia kódovania informácií, ktorá využíva grafický dvojrozmerný kód, ktorý sa skladá z čiernych a bielych blokov usporiadaných v štvorcovom formáte, obsahujúce informácie v binárnom formáte. Generovanie, skenovanie a dekódovanie QR kódov umožňuje množstvo voľne dostupných aplikácií a webových nástrojov. QR technológia sa využíva v mnohých oblastiach, ako napríklad v reklamách alebo v logistike. V prípade

leteniek a vstupeniek môže byť použitá ako elektronická verzia tlačenej vstupenky, ktorá je jednoducho skenovateľná a teda umožňuje efektívnejšiu kontrolu. Jednou z jej hlavných výhod je jednoduché použitie a flexibilita pri vytváraní a skenovaní. QR kódy sú tiež vybavené funkciou korekcie chýb, ktorá umožňuje, aby boli čitateľné aj v prípade, že sú poškodené alebo znečistené [37]. Táto funkcionálna je dosiahnutá pomocou redundancie⁵ dát, ktoré sa zakódujú do kódu a umožňujú korekciu chýb pri čítaní.

QR kódy majú štyri úrovne korekcie chýb:

- **L** - 7% dát môže byť opravených
- **M** - 15% dát môže byť opravených
- **Q** - 25% dát môže byť opravených
- **H** - 30% dát môže byť opravených

Každá úroveň korekcie chýb sa líši podľa množstva redundantných dát, ktoré sú zakódované do QR kódu. Vyššia úroveň korekcie chýb znamená, že do kódu je zakódované viac redundantných dát, čo zvyšuje pravdepodobnosť úspešného čítania aj pri poškodenom QR kóde. Nevýhodou môže byť nutnosť, aby používateľ disponoval zariadením, ktoré je schopné kód nasnímať a dekodovať. Okrem toho, je potrebné dôverovať zdroju, ktorý kód vytvoril, pretože môžu obsahovať nebezpečné odkazy. Celkovo je však technológia QR kódov stále veľmi populárna a efektívna vo svojej jednoduchosti a flexibilitate, a preto nájde využitie v mnohých odvetviach.

Technológia QR kódov je jednoduchá, ale zároveň veľmi efektívna forma prenášania dát medzi zariadeniami. Výhodou QR kódov je, že sú schopné ich naskenovať a dekodovať všetky inteligentné zariadenia, ktoré disponujú funkčnou kamerou. Zobrazenie vygenerovaného QR kódu študentom je možné buď cez projektor pre celú triedu, alebo umiestnením zariadenia s kódom na katedru, kde ho môžu naskenovať okoloidúci študenti. V prípade automatizácie evidencie dochádzky študentov však táto forma nie je efektívna pri väčšom počte študentov, keďže to môže viesť k čakacím frontám na naskenovanie kódu. Pri zobrazení kódu cez projektor môže nastať problém pri skenovaní z dôvodu nízkej ostrosti obrazu. QR kódy však môžu byť využité ako alternatívna možnosť v prípade, že primárna, efektívnejšia technológia zlyhá, a teda môžu byť využité ako sekundárna funkcionálna pre automatizáciu evidencie dochádzky.

Študentská karta ISIC

V súčasnej dobe je na mnohých vysokých školách využívaná študentská karta ISIC na evidenciu dochádzky študentov, ktorá zabezpečuje množstvo výhod pre vysokoškolské inštitúcie a študentov. Študentská karta ISIC je typom karty s mikročipom, ktorá slúži na identifikáciu konkrétneho študenta. Mikročip typu MIFARE 1k slúži na uchovávanie informácií o študentovi, ako je napríklad jeho meno a identifikačné číslo. Mikročip na karte je zabezpečený kryptografickým kľúčom, ktorý zabraňuje neoprávnenému prístupu k dátam na karte. Kryptografický kľúč okrem toho zaručuje integritu a autenticitu dát na karte. Pre získanie dát z karty je nevyhnutné použitie certifikovaného čítača takýchto kariet.

⁵Redundantná informácia - informácia, ktorá zaberá viac priestoru, než je nevyhnutné

Študentské karty ISIC predstavujú účinný spôsob identifikácie študentov a sú užitočné v mnohých oblastiach, avšak nie pri využití ako automatizácia technológia evidencie dochádzky študentov. Hlavným dôvodom je skutočnosť, že dáta z ISIC karty nie je možné jednoducho získať pomocou NFC z mobilného zariadenia, čo by vyžadovalo nákup čítačov kariet pre každú miestnosť v každej fakulte. Toto riešenie by bolo nákladné a nepraktické, a preto je táto forma evidencie dochádzky nevhodná a nedostačujúca.

Kapitola 4

Návrh riešenia

Pred návrhom riešenia automatizácie dochádzky študentov bolo nutné získať dostatočné množstvo informácií a dát o aktuálnych formách evidencie dochádzky a vhodných technológiách pre tento účel. Na základe týchto získaných dát bolo možné vyhodnotiť výhody a nevýhody rôznych postupov a navrhnúť prvotné návrhy implementácie do aplikácie Moje VUT.

4.1 Výber vhodných technológií

V kapitole 3.3 boli zistené najvhodnejšie technológie na účel implementácie automatizácie dochádzky. Z analýzy vyplýva, že najvhodnejšie technológie, ktoré sa nachádzajú v mobilných zariadeniach, sú:

- Bluetooth
- QR kód
- NFC

Pre spoľahlivú a kvalitnú implementáciu automatizácie evidencie študentov bolo potrebné vybrať 2 technológie, **primárnu** a **sekundárnu**.

Primárna technológia

Hlavnou technológiou, ktorá bude navrhnutá ako primárne používaná za účelom automatizácie evidencie dochádzky študentov, je potrebné zvoliť takú, ktorá bude schopná zabezpečiť vysokú efektivitu aj pri evidencii vyššieho počtu študentov v rovnaký moment. Musí byť spoľahlivá, bezpečná, dostupná a ľahko použiteľná s dostatočným dosahom, aby mohla byť využívaná aj vo väčších miestnostiach bez potreby presúvania sa. Z analýzy vhodných technológií je možné usúdiť, že najvyhovujúcejšou technológiou je Bluetooth (s podrobnou analýzou v kapitole 3.3.1), konkrétne jej rozšírenie **BLE**. Výhody ako nízka spotreba energie, schopnosť rýchlej výmeny dát medzi viacerými zariadeniami súčasne a dostatočný dosah splňujú všetky dôležité body pre vhodnú technológiu.

Sekundárna technológia

V kontexte evidencie dochádzky študentov je termín sekundárna použitý na označenie náhradnej technológie, ktorá sa použije v prípade, že primárna technológia nebude fungovať

pre niekoľko študentov. Cieľom tejto sekundárnej technológie nie je nahradenie primárnej technológie, ale iba poskytnutie rýchlejšej alternatívy pre evidenciu dochádzky pre daných študentov. S ohľadom na použitie sekundárnej technológie pre evidenciu dochádzky je nevyhnutné, aby táto technológia bola pre študentov a vyučujúceho ľahko použiteľná.

Po zhodnotení rôznych možností technológií je možné s istotou povedať, že najvhodnejšou možnosťou sú **QR kódy** (analýza v 3.3.4). Hlavným dôvodom, prečo je táto technológia považovaná za najvhodnejšiu, je jej jednoduchá použiteľnosť. S ohľadom na to, že takmer každé inteligentné zariadenie je vybavené kamerou, je skenovanie QR kódu pomerne ľahké a dostupné pre všetkých študentov. V prípade, že primárna technológia Bluetooth nebude pre niektorých študentov fungovať, môžu títo študenti naskenovať QR kód, aby potvrdili svoju dochádzku, a ušetrili tak čas a úsilie učiteľa, ktorý by inak musel manuálne overiť každého študenta zvlášť.

4.2 Návrh funkcionality

Dôkladne premyslený návrh tvorí kľúčový začiatok pre úspešnú realizáciu automatizácie evidencie dochádzky. Je nevyhnutné dôkladne zvážiť, ako bude aplikácia fungovať a ako bude navrhnutá jej logika na pozadí. Tento proces má zásadný vplyv nielen na kvalitu implementácie, ale aj na zabezpečenie a stabilitu aplikácie.

Prvým krokom pre zahájenie automatizovanej evidencie dochádzky na vyučovanej hodine je vygenerovanie unikátneho identifikátora, takzvaného dochádzkového **tokenu**, s časovo obmedzenou platnosťou, ktorú určuje vyučujúci. Ak je tento token platný a aktívny, je umožnená výmena dát medzi vyučujúcim a študentmi a overovanie študentov zo strany servera. Pokiaľ token nie je platný alebo vytvorený, nebude možné overovať dochádzku, vymenat dáta pomocou BLE alebo zobrazit QR kód.

4.2.1 Bluetooth Low Energy

Za účelom overenia účasti všetkých študentov prítomných na hodine sa vyžaduje odosielanie a prijímanie viacerých dát súčasne medzi mobilnými zariadeniami. Návrh bezpečného a energeticky úsporného zdieľania dát predstavuje kľúčový prvok návrhu, ktorý je potrebné navrhnuť s cieľom minimalizovať využitie Bluetooth Low Energy a zároveň zachovať jeho zamýšľanú funkcionality.

Finálny návrh funkcionality

1. Vygenerovanie tokenu s určitou časovou platnosťou, za účelom zahájenia dochádzky
2. Vyučujúci inicializuje prijímanie dát od študentov v dosahu technológie BLE
3. Študenti zahájia vysielanie svojich šifrovaných identifikátorov
4. Overenie validity študenta na strane VUT serveru
5. Aktualizácia stavu dochádzky študenta

Bezpečnosť

Pôvodný návrh systému automatickej evidencie dochádzky bol navrhnutý tak, že vyučujúci by dáta vysielal, a študenti by ich prijímali, čím by sa minimalizovalo zahlcovanie komunikačných kanálov BLE, a tým by sa predišlo interferenciám. Výsledkom by bol rýchlejší a energeticky efektívnejší systém, keďže by bolo potrebné iba jedno vysielacie zariadenie a viaceré prijímajúce zariadenia.

Avšak, aby sa predišlo obchádzaniu systému evidencie dochádzky, kde by študenti mohli zdieľať dáta zachytené pomocou proxy¹ zariadenia s ostatnými študentmi, ktorí sa na hodine nenachádzajú, bolo potrebné zmeniť návrh tak, aby študenti vysielali dáta, vyučujúci ich prijímal a následne odosielať na server, za účelom overenia. Takýto prístup je náročnejší na implementáciu, no zabraňuje úmyselnému obchádzaniu systému a zabezpečuje efektívnu a bezpečnú evidenciu dochádzky.

V rámci zaručenia bezpečnosti tejto implementácie bol navrhnutý ďalší bezpečnostný prvok, ktorý spočíva v **šifrovaní identifikátora** študenta pomocou evidenčného tokenu konkrétnej hodiny. Študent teda neposiela svoj bežný identifikátor, ale jeho šifrovanú variantu, ktorú si musí vyžiadať zo servera. Hlavným cieľom tohto návrhu je zabezpečenie anonymity študenta, aby nebol identifikovateľný nežiadúcimi zariadeniami. Okrem toho slúži aj ako prevencia proti snahe o falšovanie dochádzky, keďže študent nepozná vopred svoj zašifrovaný identifikátor. Týmto spôsobom sa zaručí bezpečnosť a efektívnosť implementácie, keďže šifrovaný identifikátor študenta sa bude meniť s každou novou automatizovanou evidenciou dochádzky.

Žiaľ, tak ako každá forma evidencie dochádzky, aj tu existuje spôsob, ako evidenciu dochádzky zneužiť. V tomto prípade azda najväčší problém nastáva, ak študent prinesie aj mobilné zariadenie svojho spolužiaka, ktorý sa na hodine nenachádza, a použije ho na jeho evidenciu. Tento problém je veľmi zložitý riešiť, pretože existuje veľa spôsobov, ako ho obísť. Jeden z návrhov by mohol byť vyžadovanie biometrického overenia², čo by ale nevyriešilo prípad, kedy by študent pridal validný otláčok svojho spolužiaka do tohto zariadenia. Taktiež vzhľadom na to, že nie každé mobilné zariadenie disponuje touto technológiou, tento návrh v praxi nie je možný uskutočniť.

4.2.2 QR kód

Implementácia technológie QR kódu má byť navrhnutá ako rýchla a dostupná alternatíva pre evidenciu dochádzky študentov, ktorý z nejakého technického dôvodu nemôžu využiť technológiu BLE. QR kód bude slúžiť ako doplnkový spôsob zaznamenávania prítomnosti študenta. Skenovanie QR kódu bude možné iba pomocou kamery zariadenia, čím sa vylúči možnosť nahratia obrázka z galérie, a tak sa zmäňší riziko zneužitia tohto kódu na zaznamenanie dochádzky.

Finálny návrh funkcionality

1. Vygenerovanie tokenu s určitou časovou platnosťou, za účelom zahájenia dochádzky
2. Vyučujúci vygeneruje a zobrazí QR kód vo svojom zariadení

¹Proxy funguje ako sprostredkovateľ medzi zariadením a cieľovým serverom, ktorý je schopný dáta čítať alebo zmeniť

²Biometrické overenie v mobilnom zariadení je proces overenia identity užívateľa na základe jeho fyzických vlastností, ako napríklad odtlačkov prstov, rozpoznávania tváre alebo dúhovky oka.

3. Študent tento kód naskenuje
4. Overenie validity študenta na strane VUT serveru
5. Aktualizácia stavu dochádzky študenta

Bezpečnosť

Hlavný problém tejto implementácie je jej bezpečnosť, čo je zároveň dôvod, prečo bola táto funkcionálna navrhnutá len ako náhradná možnosť. Existuje riziko, že študent si QR kód odfoťí, následne ho zdieľa s ostatnými študentami, ktorí sa na hodine nenachádzajú, tí si QR kód zobrazia na inom zariadení alebo monitore, naskenujú ho a tým si potvrdia dochádzku na hodine. Toto žiaľ predstavuje problém, ktorý nemá mnoho riešení, a preto nie je odporúčané používať túto technológiu, pokiaľ to nie je nevyhnutné. Avšak, časový limit platnosti tokenu môže obmedziť takýchto študentov, pretože ak vyprší jeho platnosť, overenie prítomnosti pomocou QR kódu už nebude možné. Taktiež sa tu môže vyskytnúť rovnaký problém ako pri technológii BLE, kedy študent prinesie na hodinu druhé zariadenie s účtom jeho spolužiaka, ktorý sa na danej hodine nenachádza, naskenuje pomocou neho QR kód a tým mu úspešne overí dochádzku.

4.3 Výber vhodných knižníc

Správny výber knižníc pre vývoj mobilnej aplikácie v React Native je kľúčový faktor pre úspešný a efektívny vývoj mobilnej aplikácie. Vzhľadom na rozšírenosť React Native sú k dispozícii mnohé knižnice, ktoré spĺňajú rôzne požiadavky, vďaka čomu nemusia byť navrhnuté a implementované od základov. Pri výbere knižníc pre automatizáciu evidencie dochádzky bolo nutné zvážiť niekoľko kritérií:

1. **Funkčnosť knižnice** a jej schopnosť splniť požiadavky implementácie. Toto kritérium je dôležité, pretože zabezpečujú bezproblémovú implementáciu automatizácie evidencie študentov.
2. **Kompatibilita knižnice** s mobilnými platformami Android a iOS je veľmi dôležitým faktorom pre úspešnú implementáciu v aplikácii Moje VUT, pretože existuje pre oba operačné systémy. Nie všetky knižnice sú kompatibilné a plne funkčné na oboch systémoch, preto je nutné zvoliť knižnicu, ktorá bude plne podporovaná na oboch platformách.
3. **Bezpečnosť knižnice** predstavuje kritický faktor pri výbere knižníc. Knižnica musí zabezpečovať vysokú úroveň bezpečnosti a stability, aby výsledná implementácia bola spoľahlivá a bezpečná pre užívateľov.
4. **Udržiavateľnosť knižnice** je rovnako dôležitý faktor, pretože aplikácia sa bude časom meniť a vyvíjať. Z tohto dôvodu je kľúčové zvoliť knižnicu, ktorá bude schopná sa prispôbiť potenciálnym rozšíreniam a bude ľahko udržiavateľná do budúcnosti.

Aby bola implementácia kompletná, bolo treba nájsť, otestovať a vybrať knižnice pre tieto plánované funkcionality aplikácie:

- Komunikácia pomocou BLE
- Generovanie a skenovanie QR kódov
- Asynchrónny interval schopný fungovať aj na zamknutej obrazovke zariadenia

Knižnice pre podporu BLE

Pri výbere knižnice na komunikáciu pomocou BLE medzi zariadeniami sa vyskytli mnohé problémy. Mnohé testované knižnice mali kritické nedostatky, ktoré bránili ich použitiu do finálnej implementácie. Proces testovania knižníc pre podporu BLE bol veľmi zdĺhavý aj z dôvodu, že veľa knižníc nebolo možné spojzduť na verzií React Native, ktorú aplikácia Moje VUT využíva.

React Native BLE Manager

React-Native-BLE-Manager je knižnica pre React Native, ktorá umožňuje komunikáciu medzi mobilným zariadením a inými zariadeniami pomocou technológie Bluetooth Low Energy [16]. Knižnica je vyvíjaná spoločnosťou Innove, má otvorený zdrojový kód a má širokú podporu zo strany komunity. Táto knižnica poskytuje jednoduché rozhranie pre skenovanie a pripojenie k BLE zariadeniam, a tiež pre odosielanie a prijímanie dát. Jej hlavnou výhodou je podpora pre viacero platforiem, vrátane iOS a Android, a zabezpečuje jednoduchý a spoľahlivý spôsob komunikácie pomocou BLE. Táto knižnica taktiež umožňuje vysielanie a prijímanie notifikácií z BLE zariadení, čo umožňuje rýchlu a efektívnu výmenu dát. Knižnica využíva asynchrónne volania pre efektívne spracovanie prijatých dát.

Medzi hlavné nevýhody tejto knižnice je problematický režim funkcionality na pozadí, ktorý nie vždy funguje spoľahlivo. Ďalším problémom je, že účelom tejto knižnice nie je primárne komunikácia medzi dvoma mobilnými zariadeniami, ale skôr medzi mobilným zariadením a iným, jednoduchším zariadením, ako napríklad inteligentnými hodinkami. Táto knižnica taktiež kladie dôraz na párovanie, čo pre náš účel nie je nutné. To znamená, že komunikácia medzi dvoma mobilnými zariadeniami je relatívne náročná na implementáciu a vyžaduje zavedenie rôznych algoritmov, aby bola komunikácia medzi viacerými zariadeniami úspešná a spoľahlivá. Z tohto dôvodu nie je táto knižnica ideálna na implementáciu automatizovanej evidencie dochádzky.

React Native BLE PLX

React-Native-BLE-PLX je knižnica, ktorá umožňuje vývojárom jednoduché a intuitívne používanie Bluetooth Low Energy technológie v React Native aplikáciách, podobne ako *React-Native-BLE-Manager* (4.3). Je vyvíjaná spoločnosťou Intent ako vrstva nad platformovo špecifickými BLE knižnicami, čím sa zaručuje väčšia kompatibilita so všetkými podporovanými zariadeniami a operačnými systémami [17]. Medzi výhody React Native BLE PLX patrí jej jednoduché použitie, ktoré je zapríčinené jeho intuitívnym API. Taktiež je veľmi dobre zdokumentovaná a poskytuje rôzne príklady použitia. Okrem toho podporuje aj pripájanie k zariadeniam pomocou MAC adres, čo môže byť užitočné pri niektorých prípadoch.

Bohužiaľ, podobne ako pri knižnici *React-Native-BLE-Manager* (4.3), aj táto knižnica nie je zameraná na komunikáciu medzi dvoma mobilnými zariadeniami. V dokumentácii dokonca uvádzajú, že tento typ komunikácie neodporúčajú a nepodporujú. Preto nie je táto knižnica vhodná na účel automatizácie evidencie dochádzky.

React Native Contact Tracing

Knižnica *RN-Contact-Tracing* je knižnica s otvoreným zdrojovým kódom pre React Native, ktorá umožňuje zistiť, či je iné zariadenie v blízkosti. Je navrhnutá tak, aby umožnila vývojárom rýchlo a jednoducho implementovať funkcionality kontaktného sledovania do svojej aplikácie, primárne však pomáhala pri boji proti pandémie Covid-19 [14]. Knižnica využíva Bluetooth Low Energy na výmenu anonymných identifikátorov s inými zariadeniami, ktoré majú túto funkcionality tiež implementovanú. Tieto identifikátory sú šifrované, aby zabezpečili súkromie a bezpečnosť dát. *RN-Contact-Tracing* umožňuje určiť interval vysielania a prijímania identifikátorov, nastavenie obmedzenia počtu kontaktov a definovanie maximálnej vzdialenosti, pri ktorej sa identifikátor pošle.

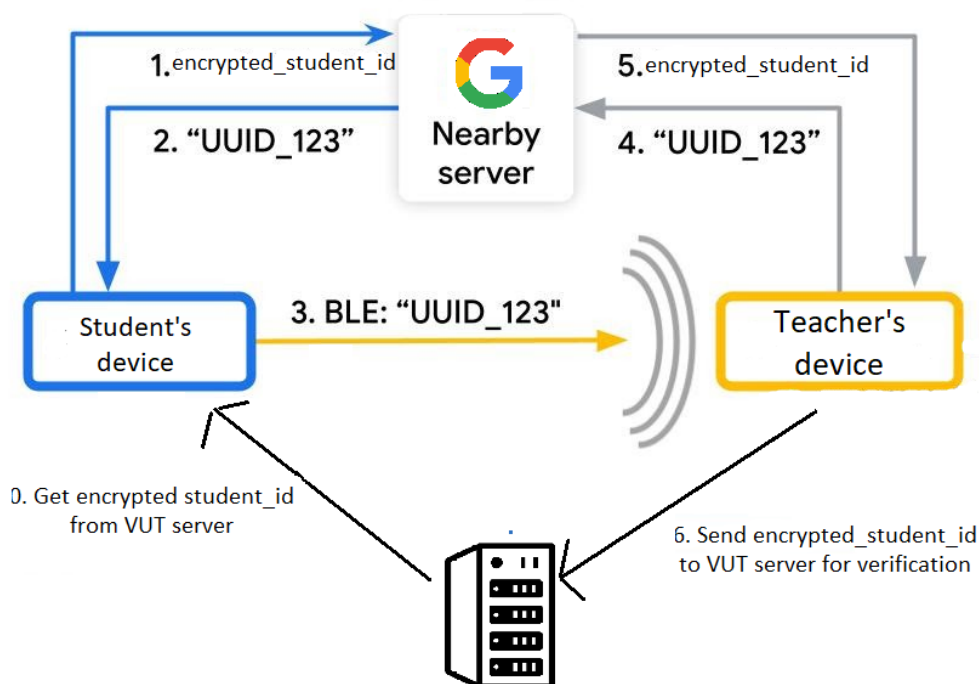
Táto knižnica by sa mohla využiť za účelom automatizácie evidencie dochádzky, keďže poskytuje všetky funkcionality, ktoré sú potrebné na jej funkčnú implementáciu. Bohužiaľ však nie je možné ju integrovať do projektu Moje VUT, pretože táto knižnica vyžaduje veľmi špecifické verzie knižníc, na ktorých je závislá. Tento problém je zároveň zapríčinený dlhodobou neaktívnosťou vývojárov knižnice.

React Native Google Nearby Messages

Knižnica *React-Native-Google-Nearby-Messages* poskytuje jednoduché rozhranie pre prácu s Google Nearby Messages API (popísanej v 3.3.1) v React Native. Táto knižnica umožňuje vývojárom využiť funkcie Nearby Messages API, ako je napríklad identifikácia blízkosti zariadení a komunikácia medzi nimi. Pomocou tejto knižnice je teda možné vytvárať aplikácie, ktoré dokážu odosielať a prijímať správy medzi zariadeniami pomocou technológie BLE. Táto knižnica je jednoduchá na použitie, pretože poskytuje kompletnú dokumentáciu a ukážkový kód na implementáciu rôznych funkcií [26]. Výhoda tejto knižnice je taktiež podpora operačných systémov Androidu a iOS. Medzi nevýhody patrí závislosť od Google služieb a ich dostupnosti, takže v prípade nedostupnosti týchto služieb môže byť funkcionality výslednej aplikácie obmedzená.

Knižnicu sa podarilo spojzdať v projekte aplikácie Moje VUT po viacerých problémoch s kompatibilitou a niekoľkých zmenách v spôsobe zostavenia. Po niekoľkých testoch stability a funkcionality tejto knižnice sa dospelo k záveru, že táto knižnica splňuje všetky potrebné požiadavky pre implementáciu do Moje VUT a môže byť využitá pre efektívnu automatizáciu evidencie dochádzky.

Hlavným problémom s touto knižnicou je problém s kompatibilitou pri zariadeniach s operačným systémom iOS. Pri pokuse o zostavenie samostatne spustiteľnej aplikácie pre tento systém sa vyskytujú zložitejšie chyby, ktoré si vyžadujú dôkladnú analýzu zdrojového kódu knižnice, aby bolo možné zaručiť jej kompatibilitu a stabilitu na iOS. Z tohto dôvodu je zatiaľ plán implementácie automatizovanej evidencie dochádzky iba pre zariadenia s Androidom, kde integrácia tejto knižnice prebieha bez problémov.



Obr. 4.1: Návrh komunikácie mobilných zariadení pomocou služby Nearby Messages API

Z dôvodu rozsiahlosti služby Nearby Messages a jej efektívneho využívania technológie BLE za účelom komunikácie medzi viacerými mobilnými zariadeniami je možné usúdiť, že knižnica *React-Native-Google-Nearby-Messages* je **najvhodnejšia** pre účel implementácie automatizovanej evidencie dochádzky študentov.

Knižnice s podporou skenovania QR kódu

Pri výbere knižníc pre podporu skenovania QR kódov v projekte Moje VUT sa taktiež objavili určité prekážky. Veľké množstvo dostupných knižníc má verziové závislosti, ktoré sa nedajú splniť v projekte Moje VUT. Tento problém sa prejavuje najmä u knižníc, ktoré nie sú vývojármi aktívne podporované. Výhodou však je, že požiadavky na tieto knižnice sú minimálne, keďže musia zvládnuť len skenovanie QR kódov v reálnom čase.

React Native Vision Camera

React-Native-Vision-Camera je knižnica, ktorá poskytuje mnoho funkcionalít, primárne rozhranie pre prístup k fotoaparátu a čítačke QR kódov na mobilných zariadeniach. Knižnica implementuje základné funkcie fotoaparátu, ako napríklad ovládanie blesku alebo správu zaostrenia. Pre čítanie QR kódov poskytuje knižnica API, ktoré umožňuje načítanie QR kódu z živého obrazu kamery a následné získanie dát z tohto kódu. Pre prácu s objektami a dátami v tejto knižnici je potrebné použiť takzvaný **Frame Processors** (snímkový procesor), ktorý umožňuje pracovať s tým, čo kamera vidí [27]. Frame processors sú implementované ako funkcie, ktoré môžu byť volané pri každej snímke z kamery. Toto môže byť využité na rôzne účely, napríklad na detekciu tvárí, rozpoznávanie objektov alebo čítanie QR kódov.

Žiaľ, táto funkcionálnosť vyžaduje knižnicu s názvom *react-native-reanimated*. Táto knižnica nahradzuje existujúci modul animácií a animačné prvky v React Native, čím umožňuje animovať UI prvky v mobilných aplikáciách v reálnom čase [21]. Knižnica React Native Reanimated pracuje na základe takzvaných animovaných uzlov, ktoré reprezentujú grafy zložených z uzlov, ktoré predstavujú rôzne animačné hodnoty, ako sú napríklad pozícia, veľkosť alebo farba. Tieto uzly môžu byť následne prepojené rôznymi spôsobmi a využité na animáciu UI prvkov v aplikácii. Je to skvelá funkcionálnosť s mnohými prípadmi využitia, keďže umožňuje implementovať prvky, ktoré by boli inak veľmi zložité. Nastáva tu ale problém, a to taký, že táto knižnica môže rozbiť iné prvky, ktoré neboli navrhnuté s využitím tejto knižnice.

Knižnica React Native Vision Camera poskytuje rozsiahle funkcie, avšak pre implementáciu QR kódov do aplikácie Moje VUT je príliš komplikovaná a môže predstavovať riziko poruchy existujúcich prvkov aplikácie, zvlášť v spojení s knižnicou *react-native-reanimated*. Preto nie je vhodné použiť túto knižnicu na účel implementácie QR kódov.

React Native Qrcode Scanner

React-Native-Qrcode-Scanner je knižnica pre React Native, postavená na knižnici *react-native-camera*, ktorá je už dlhšiu dobu nepodporovaná a neaktualizovaná [28]. Táto knižnica používa nízkoúrovňové API volania, ktoré umožňujú skenovanie QR kódov, vďaka priamemu prístupu k fotoaparátu zariadenia. Táto knižnica je pomerne jednoduchá a ľahko implementovateľná, vďaka čomu umožňuje vývojárom jednoducho integrovať skenovanie QR kódov do svojich aplikácií.

Bohužiaľ, v rámci testovania pri výbere knižníc sa nepodarilo integrovať danú knižnicu do projektu Moje VUT, znova v dôsledku verziových konfliktov. Tento problém nastal vzhľadom na absenciu aktualizácie danej knižnice vývojárom.

React Native Camera Kit

React-Native-Camera-Kit je knižnica pre React Native, ktorá poskytuje jednoduchý spôsob pridania funkcie fotoaparátu do mobilných aplikácií. Jednou z hlavných výhod tejto knižnice je jej natívny výkon a podpora mnohých funkcionálností, ako napríklad skenovanie QR kódov [31]. Knižnica poskytuje niekoľko rôznych komponentov pre rôzne použitia, ako napríklad *CameraScreen*, ktorý umožňuje nahrávanie videa a fotografií, alebo *CameraZoomControl*, ktorý umožňuje používateľovi priblížiť alebo oddialiť záber.

Celkovo táto knižnica predstavuje jednoduchý spôsob skenovania QR kódov pre vývojárov mobilných aplikácií. Splňuje všetky požiadavky potrebné na úspešnú a efektívnu integráciu do aplikácie Moje VUT. Podobne ako v ostatných knižniciach sa tu objavil problém s verziami potrebných knižníc. Avšak, prostredníctvom úpravy nastavení zostavovania knižnice sa podarilo tento problém vyriešiť, a preto je **ideálnou voľbou** za účelom skenovania dochádzkového QR kódu.

Knižnica s podporou generovania QR kódu

Výber knižnice na generovanie QR kódov bol relatívne jednoduchý, najmä vzhľadom na veľmi nízke požiadavky na takúto knižnicu. Knižnica musí byť schopná iba zobrazíť QR kód vygenerovaný na základe vstupných dát.

React Native Qrcode Svg

React-Native-Qrcode-SVG je knižnica pre React Native, ktorá umožňuje vytvárať QR kódy vo forme vektorovej grafiky pomocou SVG³. Knižnica taktiež umožňuje pridať do kódu logo alebo ikonu, ktorá bude viditeľná v stredovej časti QR kódu [18].

Táto knižnica je pomerne moderná a pravidelne udržiavaná, čo predstavuje veľkú výhodu pri výbere vhodnej knižnice. I keď sa vyskytli drobné problémy s verziami závislostí, ktoré boli ľahko riešiteľné, knižnica fungovala bezproblémovo počas testovania. Preto je táto knižnica **ideálnou voľbou** na generovanie QR kódov.

Knižnica s podporou aplikovania zmien do externých knižníc

Vzhľadom na problémy, ktoré sa vyskytli po inštalácii niektorých knižníc, ktoré viedli až k neschopnosti zostaviť projekt, bolo potrebné vykonať viaceré úpravy v zostavovacích nastaveniach týchto externých knižníc tak, aby boli kompatibilné s projektom Moje VUT. Keďže tieto úpravy boli vykonané špecificky pre tento projekt, bolo nevyhnutné ich distribuovať aj ostatným vývojárom, aby boli schopní na projekte pracovať. Z tohto dôvodu bolo potrebné nájsť knižnicu, ktorá umožní uloženie týchto zmien v externých knižniciach a následne ich aplikovať.

React Native Patch Package

React-Native-Patch-Package je knižnica pre React Native, ktorá umožňuje vytvárať a aplikovať rôzne zmeny v externých knižniciach. Táto knižnica umožňuje rýchlu a jednoduchú možnosť opravy chýb alebo pridávanie vlastných funkcií [7]. Po vykonaní zmien v externej knižnici ich zanalyzuje a uloží do záplatového súboru s názvom **patch**. Patch súbor obsahuje zmeny, ktoré je možné aplikovať na zdrojový kód knižnice. Pre aplikovanie zmien v projekte stačí zavolať túto knižnicu príkazom ‘npx patch-package’.

Knižnica s podporou asynchrónneho časovača

Vzhľadom na fakt, že operačné systémy Android a iOS z dôvodu šetrenia batérie umožňujú aplikáciám na pozadí prechádzať do režimu spánku, všetky funkcie, ktoré sú spustené na pozadí pri uzamknutej obrazovke mobilného zariadenia, sa po chvíli prerušia, vrátane prenosu dát prostredníctvom technológie BLE a vymazania neplatného dochádzkového tokenu v prípade, že jeho platnosť vypršala. Preto je nevyhnutné implementovať špeciálny typ asynchrónneho intervalu, ktorý bude schopný fungovať aj pri uzamknutej obrazovke a umožní tak plynulú funkcionálnu aplikáciu bez prerušenia.

React Native Background Timer

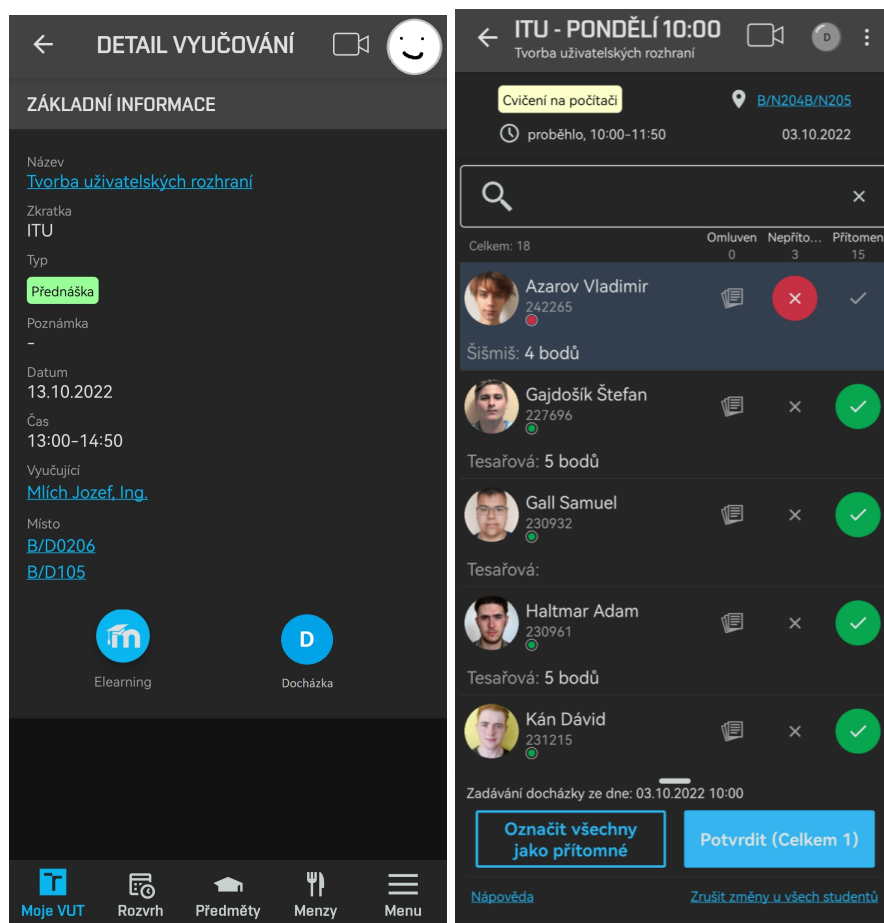
React-Native-Background-Timer je knižnica, ktorá umožňuje vytváranie časovačov a opakovaných úloh v pozadí, aj keď je aplikácia uzamknutá alebo v režime spánku. Fungovanie knižnice spočíva v založení vlastného vlákna, ktoré bude bežať v pozadí [25]. Toto vlákno umožňuje spúšťanie opakujúcich sa úloh v určitom časovom intervale, napríklad odosielanie informácií na server alebo aktualizáciu stavu aplikácie. Vlákno je naviazané na samostatný proces, čo zabezpečuje, že sa bude vykonávať aj po uzamknutí obrazovky a v režime spánku.

³SVG je škálovateľný vektorový grafický formát, ktorý popisuje 2D vektorové obrázky

4.4 Návrh uživatelského rozhrania

Návrh uživatelského rozhrania je veľmi dôležitou časťou procesu vývoja aplikácií, ktorá zohľadňuje potreby a očakávania užívateľov [32, 22]. Jeho správny návrh a implementácia je nevyhnutnosťou pre zaistenie efektívnej aplikácie. Za účelom zistenia potrieb a očakávaní užívateľov bol vykonaný prieskum (viď 3.1), z ktorého boli získané cenné dáta. Pri výbere vhodného výzoru uživatelského rozhrania sa často využíva metóda prototypovania, ktorá umožňuje vytvoriť a otestovať rôzne varianty rozhrania predtým, než sa pristúpi k jej implementácii. Tento proces zahŕňa vytváranie jednoduchých makiet, ktoré sa postupne rozvíjajú a vylepšujú, pričom zohľadňujú pripomienky a návrhy užívateľov. Tieto makety a prvotné návrhy boli vytvárané v nástroji **Figma**⁴ a následne zdieľané s rôznymi zamestnancami VUT, ktorí prejavili záujem o bližšie informácie rámci vývoja automatizácie. Na základe ich spätných väzieb boli návrhy iteratívne vylepšované, aby pôsobili intuitívne.

Návrh navigácie do modulu automatizovanej evidencie



Obr. 4.2: Návrh navigovania do modulu automatizovanej evidencie dochádzky pre študenta (vľavo) a vyučujúceho (vpravo)

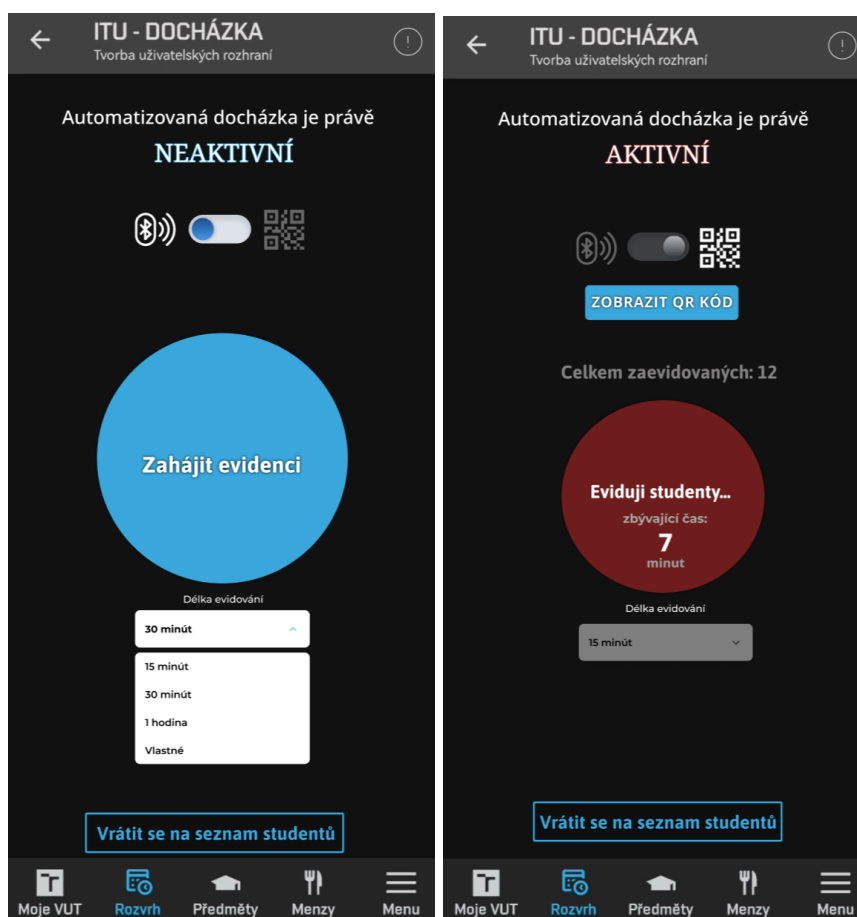
⁴Figma je nástroj pre návrh uživatelského rozhrania, ktorý umožňuje dizajnérom vytvárať, zdieľať a spolupracovať na návrhoch webových stránok a mobilných aplikácií. <https://www.figma.com/>

Nakoľko sa jedná o rozšírenie už v existujúcom module pre evidenciu študentov, je rozumné umiestniť tento prechod na modul automatizovanej evidencie priamo sem. Návrh je postavený tak, že vyučujúci má k dispozícii príslušné tlačidlo v hornom paneli navigačného menu na obrazovke evidencie konkrétnej hodiny, ktoré ho následne presmeruje na obrazovku automatizovanej evidencie dochádzky. Študent má tento preklik v detaile konkrétneho predmetu. Týmto spôsobom je zabezpečená intuitívna integrácia modulu automatizovanej evidencie do existujúceho systému.

Návrh modulu automatizovanej evidencie

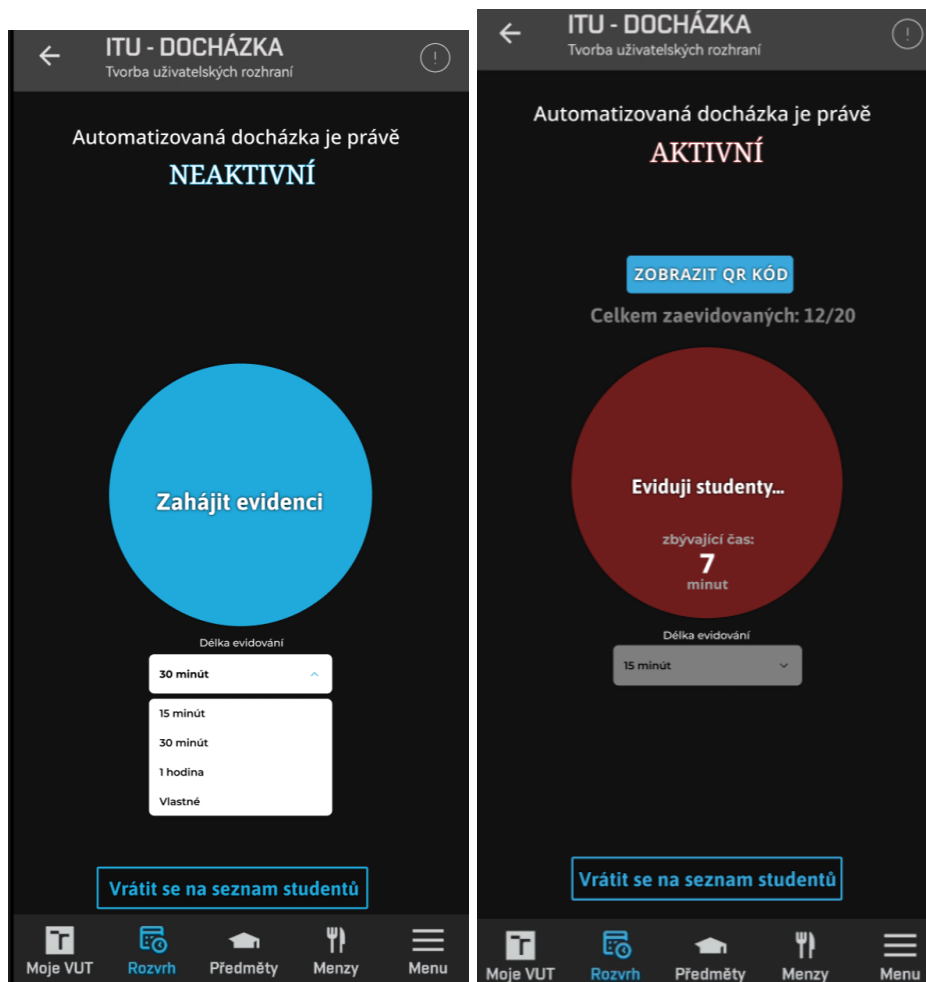
Vzhľadom na fakt, že funkcionality zo strany vyučujúceho a študenta je mierne odlišná, je nevyhnutné navrhnuť dva rôzne typy užívateľského rozhrania, ktoré budú prispôbené funkcionalitám oboch typov užívateľov. Tieto návrhy boli iteratívne vylepšované vďaka spätnej väzbe zamestnancov VUT. Pri navrhovaní týchto rozhraní bol kladený dôraz hlavne na ich intuitívnosť a použiteľnosť, s cieľom zabezpečiť čo najvyššiu úroveň efektívnosti používania tohto systému.

Návrh pre vyučujúceho



Obr. 4.3: Prvotný návrh užívateľského rozhrania automatizovanej evidencie pre vyučujúceho

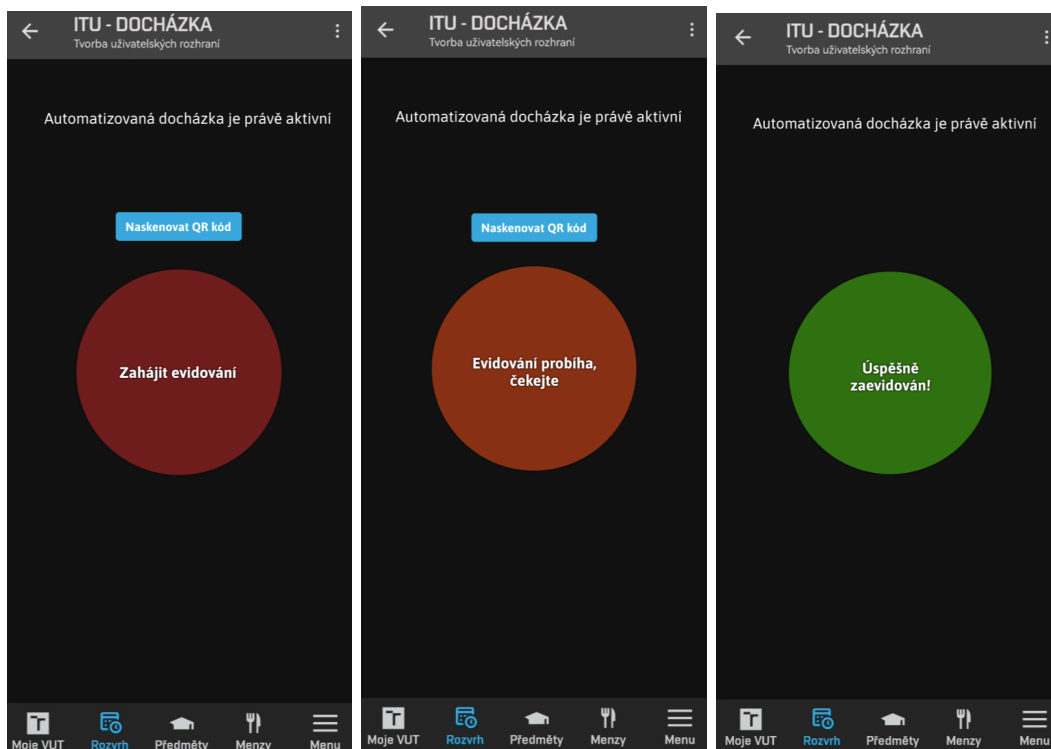
V jednom z prvých dokončených návrhov sa uvažovalo o zavedení obmedzenia na používanie iba jednej technológie v čase zahájenia evidencie. Konkrétne by nebolo povolené využívať súčasne technológiu BLE a QR. Namiesto toho by si vyučujúci musel zvoliť jednu z týchto možností, ako je navrhnuté v 4.3. Takéto riešenie by však bolo príliš restriktívne a nepraktické, preto bol návrh prehodnotený a toto obmedzenie odstránené v ďalšom návrhu.



Obr. 4.4: Finální návrh uživatelského rozhraní automatizované evidencie pre vyučujúceho

V finálnom návrhu implementácie (4.4) pre vyučujúceho bolo spomínané obmedzenie odstránené, čím sa vyučujúcim poskytne väčšia sloboda v používaní dostupných technológií.

Návrh pre študenta



Obr. 4.5: Finálny návrh užívateľského rozhrania automatizovanej evidencie pre študenta

Návrh časti pre študenta (4.5) je o niečo jednoduchší v porovnaní s návrhom pre vyučujúceho, pretože študent nevyžaduje vedieť žiadne dodatočné informácie. Študentovi iba stačí vedieť, aké možnosti evidencie má k dispozícii a či bola jeho evidencia úspešne zaznamenaná.

4.5 Návrh rozhrania API

Aby bolo možné realizovať implementáciu automatizovanej evidencie, je nevyhnutné zabezpečiť komunikáciu so serverom VUT. Tento server musí poskytovať a prijímať všetky dôležité dáta, ktoré sú potrebné na úspešnú evidenciu študentov na hodine do informačného systému. Táto komunikácia prebieha pomocou **REST API**[1], ktorá umožňuje jednoduchú a štandardizovanú komunikáciu medzi klientom a serverom.

REST API

REST API (Representational State Transfer Application Programming Interface) je architektúra, ktorá umožňuje komunikáciu medzi klientom a serverom pomocou sady predom definovaných protokolov a konvencií. Ide o rozhranie, ktoré umožňuje klientovi vytvárať, čítať, aktualizovať a mazať zdroje na serveri pomocou HTTP⁵ protokolu. REST API má niekoľko základných metód, ktoré sa nazývajú CRUD (Create, Read, Update, Delete) operácie, umožňujúce manipulovať so zdrojmi na serveri:

⁵Hypertextový prenosový protokol, skr. HTTP - protokol na prenos HTML dokumentov a dát medzi serverom a klientom. Pôsobí ako primárna metóda prepravy informácií na internete.

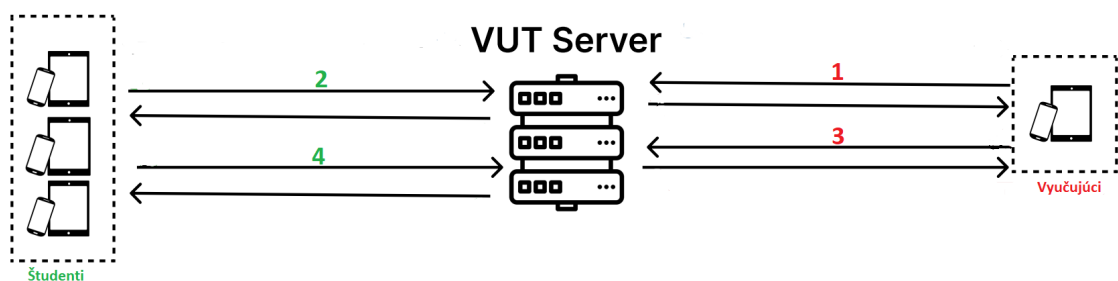
- **GET** slúži na získanie dát z určitého zdroja. Zvyčajne sa používa na získanie informácií o konkrétnom zdroji.
- **POST** slúži na vytvorenie nového zdroja. Používa sa v prípade, že chceme zaslať dáta na server.
- **PUT/PATCH** [11] slúžia na aktualizáciu existujúceho zdroja. V prípade PUT metódy sa zdroj vymení za nový, zatiaľ čo PATCH slúži na úpravu existujúceho zdroja bez jeho úplnej zámény.
- **DELETE** slúži na odstránenie zdroja a jeho dát.

Pri použití REST API vytvárame jednotlivé koncové body (endpoint), ktoré predstavujú konkrétne zdroje, s ktorými môže klient pracovať. Každý koncový bod reprezentuje určitý druh operácie, ktorú je možné na danom zdroji vykonať. Po úspešnej komunikácii pošle server klientovi HTTP odpoveď, ktorá obsahuje stavový kód a prípadne aj dáta. Stavový kód oznamuje klientovi, či bola požiadavka úspešne spracovaná alebo či došlo k nejakej chybe.

Pri návrhu koncových bodov za účelom komunikácie so serverom bolo potrebné zohľadniť všetky potrebné funkcionality, ktoré server musí poskytovať. Po dôkladnej analýze a navrhovaní potrebných koncových bodov, bolo potrebné ich odprezentovať zamestnancovi VUT, ktorý má oprávnenia na vytváranie koncových bodov pre ich následnú implementáciu.

Koncové body API pre BLE

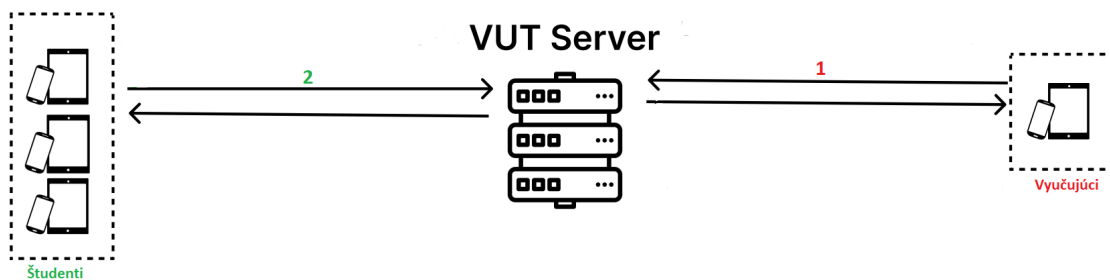
1. Vyučujúci požiada server o zahájenie dochádzky na špecifickú hodinu na určitý počet minút. Ak je overenie vyučujúceho úspešné, odpovedá klientovi s jedinečným tokenom evidencie. (**PATCH**)
2. Študent požiada server o jeho zašifrovaný identifikátor na konkrétnu hodinu. Ak pre túto hodinu existuje token, server odpovedá klientovi s jeho zašifrovaným identifikátorom. (**GET**)
3. Vyučujúci zašle študentov zašifrovaný identifikátor na server za účelom overenia jeho prítomnosti. V prípade úspechu server aktualizuje stav dochádzky študenta a odpovedá s kódom úspechu. (**PATCH**)
4. Študent požiada server o stav jeho dochádzky na konkrétnu hodinu. (**GET**)



Obr. 4.6: Komunikácia klienta so serverom za účelom evidencie pomocou BLE

Koncové body API pre QR

1. Vyučujúci požiada server o zahájenie dochádzky na špecifickú hodinu na určitý počet minút. Ak je overenie vyučujúceho úspešné, odpovedá klientovi s jedinečným tokenom evidencie, rovnako ako pri BLE. (**PATCH**)
2. Študent tento token evidencie naskenuje pomocou QR kódu zo zariadenia vyučujúceho, a následne ho odošle na server za účelom overenia. Server odpovedá, či bola evidencia úspešná alebo nie. (**PATCH**)



Obr. 4.7: Komunikácia klienta so serverom za účelom evidencie pomocou QR

Kapitola 5

Implementácia

Pred samotnou implementáciou bolo nevyhnutné zabezpečiť dostatočnú pripravenosť na implementáciu. Na základe analýzy záujmu, problematiky (3) a navrhnutých riešeníach (4) sa dá usúdiť, že boli splnené všetky potrebné predpoklady pre zaručenie správnej a efektívnej implementácie. Vzhľadom na odlišné funkcionálne požiadavky vyučujúceho a študenta, je nutné vytvoriť dve samostatné obrazovky aplikácie - jednu pre vyučujúceho a druhú pre študenta, ktoré obsahujú všetky komponenty, s ktorými užívatelia interagujú. Ďalším nevyhnutným krokom je oboznámiť sa s projektom mobilnej aplikácie Moje VUT a jej implementovanou architektúrou, ktorá je založená na technológii React Native. Vzhľadom na skutočnosť, že s touto technológiou ešte nemám skúsenosti, je dôležité sa pred implementáciou s ňou podrobne oboznámiť.

5.1 Moje VUT

Moje VUT je mobilná aplikácia, dostupná pre Android a iOS, ktorá je určená pre študentov a zamestnancov Vysokého učení technického v Brně (VUT). Bola vytvorená v roku 2020 ako bakalárska práca Ondreja Klinovskeho [19]. Aplikácia umožňuje prístup k informáciám, ktoré sú pre užívateľov relevantné, ako napríklad rozvrh hodín, zápis do predmetov, evidenciu dochádzky, prehľad študijných výsledkov, mapu fakulty a podobne. Okrem toho umožňuje aj prezeranie študijných materiálov a poskytuje informácie o aktuálnych udalostiach na univerzite. V aplikácii je k dispozícii aj prehľad o ubytovaní a stravovaní v kampuse a mnoho ďalších funkcií, ktoré môžu byť pre študentov a zamestnancov VUT užitočné. Aplikácia poskytuje podobné funkcionality a služby ako webové stránky VUT, avšak s užívateľsky prívetivejším rozhraním a možnosťou prístupu k funkcionality bez nutnosti načítavať a navigovať sa cez viaceré stránky.

Aplikácie Moje VUT využíva na implementáciu modernú technológiu React Native (2.3.1), ktorá ponúka plynulú a intuitívnu interakciu s užívateľmi, vďaka jej vysokému výkonu, responzivite a stabilite. Jazyk vývoja je Typescript, ktorý minimalizuje možnosť chýb za behu aplikácie. Aplikácia taktiež využíva mnoho rôznych API a služieb, ktoré umožňujú získavanie aktuálnych informácií a ich správne zobrazovanie v mobilných zariadeniach. Celkovo možno povedať, že aplikácia Moje VUT je užitočným nástrojom pre študentov a zamestnancov VUT, ktorá uľahčuje organizáciu a zlepšuje komunikáciu s univerzitou.

Adresárová štruktúra

Projekt aplikácie Moje VUT využíva konvenčnú adresárovú štruktúru, ktorá je typická pre projekty využívajúce technológiu React Native. Táto štruktúra umožňuje ľahkú čitateľnosť a zjednodušuje orientáciu v projekte. Vďaka tomu je navigácia pri vývoji aplikácie pomerne intuitívna.

```
/
├── android/ ..... zdrojové kódy a nastavenia pre zostavenie Android aplikácie
├── ios/ ..... zdrojové kódy a nastavenia pre zostavenie iOS aplikácie
├── patches/ ..... zmeny a opravy v externých knižniciach
├── scripts/ ..... skripty na testovanie a zostavenie aplikácie
├── node_modules/ ..... zložka obsahujúca nainštalované knižnice
├── app/ ..... súbory projektu
│   ├── assets/ ..... ikony, fonty a obrázky
│   ├── components/ ..... React Native komponenty použité v aplikácii
│   │   ├── attendance/ ..... skupina komponentov použitých v rámci dochádzky
│   │   │   ├── BLEButton.tsx ..... interaktívne tlačidlo pre BLE
│   │   │   ├── QRComponents.tsx ..... QR komponenty
│   │   │   ├── TokenCountdown.tsx ..... odpočítavadlo platnosti tokenu
│   │   │   └── MyAttendanceStatusText.tsx ..... text statusu dochádzky študenta
│   ├── hooks/ .. vlastné hooky (napr. useStores, umožňujúci prístup k MobX úložiskám)
│   ├── i18n/ ..... definície prekladov do češtiny a angličtiny
│   ├── models/ ..... definície dátových štruktúr
│   ├── parsers/ ..... definície spracovania odpovedí serverov
│   │   └── autoAttendanceParser.ts
│   ├── screens/ ..... definície všetkých obrazoviek v aplikácii
│   │   ├── AutoAttendanceStudentScreen.tsx
│   │   └── AutoAttendanceTeacherScreen.tsx
│   ├── services/ ..... definície tried pre prácu s backendom
│   │   └── VutService.ts ..... obsahuje všetky definície koncových bodov API
│   ├── stores/ ..... definícia úložisk dát MobX
│   ├── utils/ ..... pomocné funkcie rôznych funkcionalít
│   └── Styles.ts ..... definícia globálnych štýlov a vzhľadu aplikácie
├── Index.js ..... hlavný vstupný bod pre aplikáciu
├── app.json ..... súbor s konfiguráciou aplikácie
└── package.json ..... súbor s informáciami o projekte a verziách knižníc
```

V adresárovej štruktúre je možné vidieť, kde sú umiestnené jednotlivé prvky implementácie automatizovanej evidencie dochádzky. Konkrétne ide o prvky ako napríklad jednotlivé obrazovky **AutoAttendanceStudentScreen.tsx** a **AutoAttendanceTeacherScreen.tsx**, ktoré definujú použité komponenty, ich funkcie a interakciu s užívateľom. Tieto obrazovky definujú to, čo užívateľ vidí v svojom mobilnom zariadení.

Komunikácia so serverom

Za účelom komunikácie so serverom sa v aplikácii používajú koncové body REST API (4.5). Všetky koncové body, ktoré aplikácia používa, sú definované v súbore **VutService.ts** spolu s ich metódami, dátami v tele požiadavky a parserom, ktorý spracúva odpoveď zo servera. Keďže server reaguje na požiadavky odpoveďou, ktorá obsahuje stavový kód požiadavku a dáta vo formáte XML alebo JSON, tieto dáta sú spracované parserom, aby bol proces implementácie funkcionalít pre vývojára pohodlnejší.

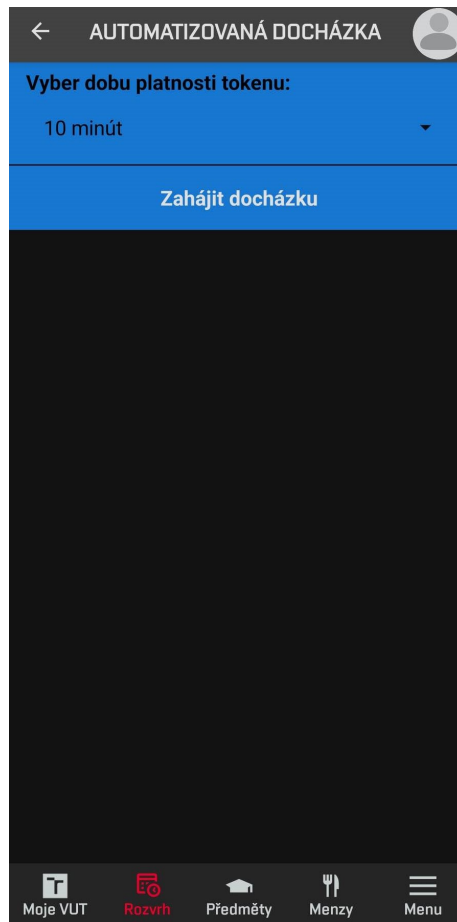
Ukladanie dát

V aplikácií Moje VUT sa na ukladanie dát získaných zo servera a aktuálnych stavov aplikácie používajú takzvané MobX Stores. Každé úložisko (Store) je zodpovedné za ukladanie relevantných dát, napríklad tokenov evidencie alebo stav dochádzky študentov. V aplikácii je viacero úložísk, preto bol implementovaný RootStore, ktorý tieto úložiská vytvára a poskytuje na nich referencie. Použitie MobX stores umožňuje zjednodušiť a zorganizovať kód aplikácie, čo v konečnom dôsledku prispieva k jej efektívnejšiemu vývoju. Hlavnou výhodou MobX stores je, že zabezpečujú deklaratívny spôsob popisu stavu aplikácie, kde sa stav mení automaticky na základe zmien dát, bez potreby ručne upravovať komponenty, čo zaručuje dynamickosť aplikácie. V projekte Moje VUT sa MobX úložiská používajú tak, že obrazovky aplikácie sú napojené na úložiská pomocou funkcie **observer**¹. Týmto spôsobom sú všetky komponenty na obrazovke informované o každej zmene v úložiskách a môžu na ne reagovať podľa potreby. Po ukončení aplikácie sa tieto úložiská vymažú, a teda nezaberajú žiadne zbytočné miesto navyše v zariadení užívateľa.

5.2 Zahájenie evidencie

Na úvod musí vyučujúci zahájiť automatizovanú evidenciu dochádzky, bez ktorej študent ani vyučujúci nie je schopný pracovať s technológiou QR a BLE, pretože by to mohlo spôsobiť nežiadúce zaťaženie BLE komunikačných kanálov (3.3.1). Pre zahájenie evidencie musí vyučujúci najprv vybrať dĺžku platnosti tokenu a kliknúť na tlačidlo ‘Zahájiť Dochádzku’. Následne sa na server pošle PATCH požiadavka s cieľom získať platný dochádzkový token pre danú vyučovaciu hodinu. Ak server úspešne overí požiadavku, odpovie mobilnému zariadeniu vyučujúceho s platným tokenom, ktorý sa uloží do príslušného Store úložiska a spustí sa proces automatizovanej evidencie dochádzky. Po spustení evidencie sa aktivuje aj asynchrónny časovač (4.3), ktorý sa stará o aktualizáciu zostávajúceho času platnosti tokenu a vymazanie tohto tokenu v prípade, ak jeho platnosť skončí. Po uplynutí platnosti tokenu sa token vymaže z úložiska a akákoľvek interakcia s BLE technológiou, či už na strane študenta alebo vyučujúceho, sa ukončí.

¹Funkcia Observer sleduje zmeny v Stores a automaticky vyvoláva prekreslenie komponenty



Obr. 5.1: Implementovaná funkcionality zahájenia automatizovanej evidencie dochádzky

Aby sa predišlo časovým nezrovnalostiam medzi mobilným zariadením a serverom, ktorý odpovedá nielen tokenom dochádzky, ale aj časom zahájenia a skončenia jeho validity, je nutné vyriešiť problém, ktorý sa môže vyskytnúť, ak má mobilné zariadenie a server rôzny čas. Riešením tohto problému je určenie dĺžky platnosti tokenu, ktorý server vytvoril, a následné pripočítanie tejto hodnoty k lokálnemu času zariadenia, čím sa zabezpečí jednotný čas platnosti tokenu a zabráni sa možným situáciám, kedy by vyučujúci videl platný token, ale server by ho už nebol schopný použiť na overenie dochádzky.

```
const diffInMs =  
    new Date(response.tokenTo).getTime() -  
    new Date(response.tokenFrom).getTime();  
this._tokenFrom = new Date();  
this._tokenTo = new Date(this._tokenFrom.getTime() + diffInMs);
```

5.3 Evidencia dochádzky pomocou BLE

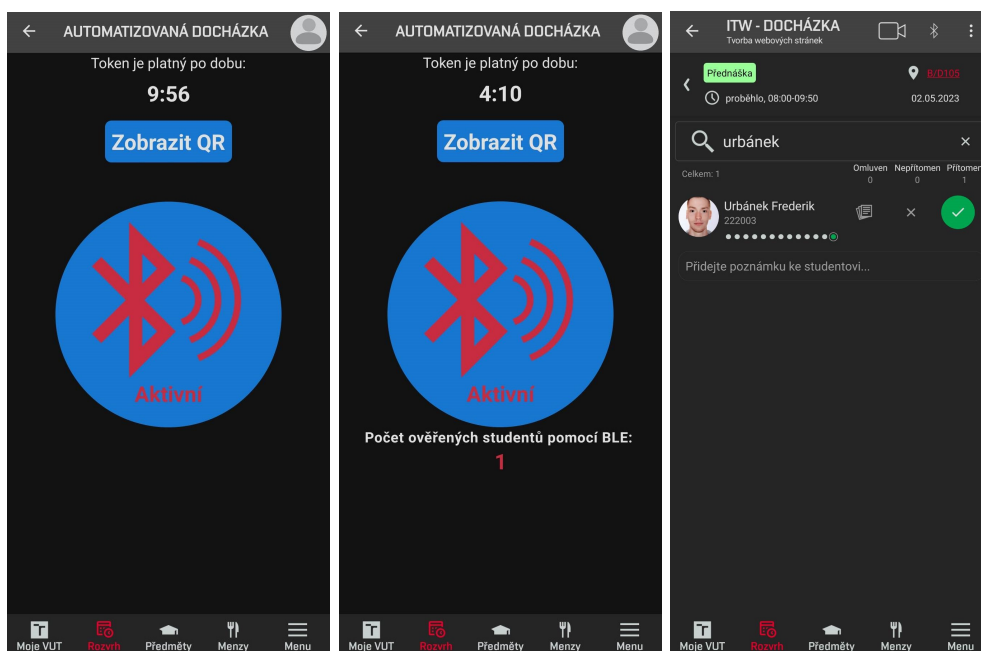
Implementácia tejto funkcionality musela zohľadniť dôležité kritériá, ako intuitívnosť používania a úsporu batérie. Ak Bluetooth nie je zapnutý, aplikácia ho automaticky zapne

po zahájení procesu evidencie a vypne ho po jeho skončení. Pri prvom spustení evidencie dochádzky je potrebné povoliť používanie technológie Nearby Messages (3.3.1), aby bolo možné použiť BLE za účelom evidencie.

Vyučujúci

Primárnym komponentom, s ktorým vyučujúci interaguje, je komponent s názvom **BLE-Button**. Tento komponent zobrazuje tlačidlo v strede obrazovky, ktoré po kliknutí mení svoj stav a ukladá ho do úložiska s názvom **classAttendanceStores**, ktoré je unikátne pre každú vyučovaciu hodinu. To znamená, že vyučujúci môže spustiť automatizovanú evidenciu aj pre viacero vyučovacích hodín naraz. Táto funkcia sa môže hodiť napríklad v prípade, keď vyučuje predmet, ktorý navštevujú študenti z rôznych fakúlt a daný predmet má teda viacero názvov.

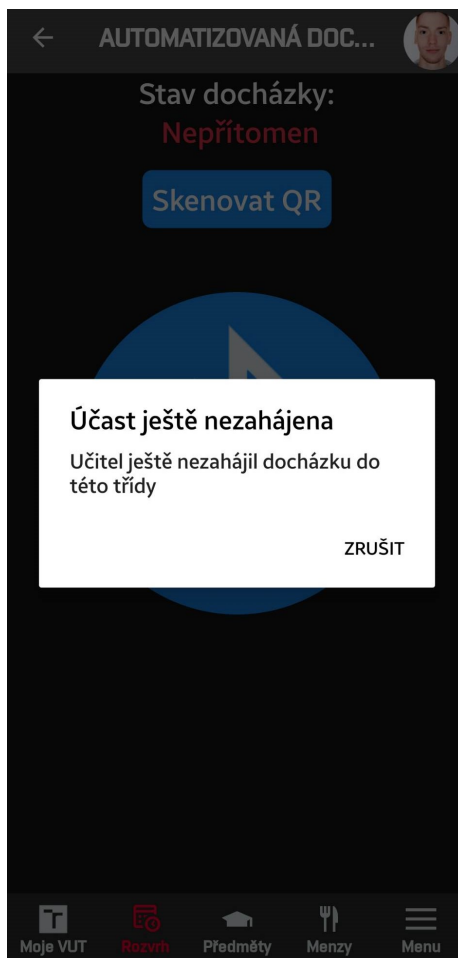
Po spustení evidencie je tlačidlo automaticky aktivované, čo spustí prijímanie dát (Subscribe) v mobilnom zariadení vyučujúceho pomocou technológie Nearby Messages. Funkciu prijímania dát je možné kedykoľvek vypnúť a zapnúť podľa potreby. Ak platnosť tokenu vyprší, prijímanie dát sa automaticky ukončí. V režime prijímania dát zachytáva mobilné zariadenie vyučujúceho dáta, ktoré zodpovedajú očakávanému formátu šifrovaného identifikátora študenta. Tieto dáta sú následne pridané do zoznamu potenciálnych študentských identifikátorov. Po každom zachytení dát sa tento zoznam spracuje a jeho jednotlivé prvky sú odoslané na server (PATCH) za účelom overenia. Ak je overenie úspešné, server aktualizuje stav dochádzky študenta a označí identifikátor ako spracovaný, aby sa predišlo opakovanému odosielaniu na server. Ak server prijaté dáta vyhodnotí ako neplatný študentský šifrovaný identifikátor, tak sa k týmto dátam pridá hodnota, ktorá indikuje, že tieto dáta netreba znova posilať na server. Týmto spôsobom sa šetrí vyťaženie servera a zabezpečuje sa ochrana proti možnému spamovaniu.



Obr. 5.2: Implementovaná funkcionálna režimu Subscribe a ukážka úspešného overenia študenta

Študent

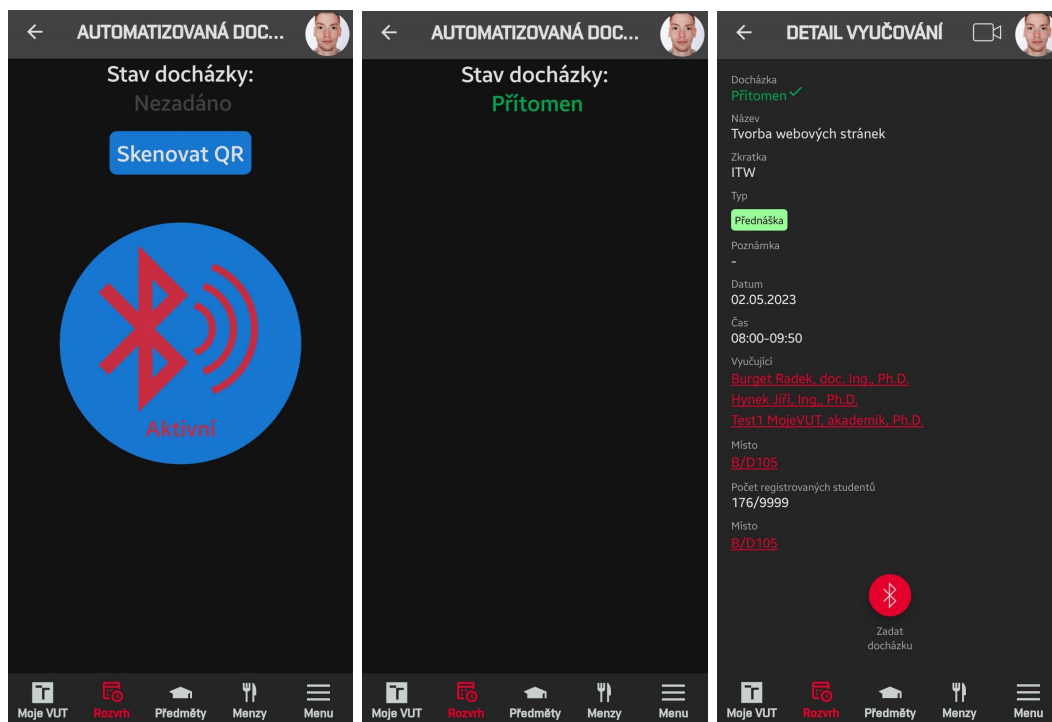
Študent interaguje s tým istým BLEButton komponentom ako vyučujúci, no v tomto prípade má opačnú funkcionality, teda dáta neprijíma, ale vysiela (Publish). S týmto komponentom môže študent interagovať iba vtedy, pokiaľ nemá dochádzku na danú hodinu v stave 'prítomen'. Pokiaľ ešte nebola zahájená automatizovaná evidencia dochádzky vyučujúcim, študentovi nie je umožnené vysielať dáta pomocou BLE. Toto overenie je možné vďaka tomu, že server je schopný poslať študentovi jeho zašifrovaný identifikátor (GET) len vtedy, ak existuje dochádzkový token na danú hodinu. Teda ak server neodpovie s príslušným identifikátorom tak je jasné, že dochádzka ešte nebola zahájená.



Obr. 5.3: Příklad, ak nebola evidencia zahájená a študent sa pokúsi zapnúť vysielať dát

Ak je dochádzka zahájená, tak po kliknutí na tlačidlo začne študentovo mobilné zariadenie proces evidencie. Tento proces začína požiadavkou na server o zašifrovaný identifikátor študenta (GET) na špecifickú hodinu. Následne sa spúšťa asynchrónny interval, ktorý pravidelne kontroluje stav jeho dochádzky a platnosti tokenu. Študent následne začne svoj identifikátor vysielať pomocou technológie BLE a Nearby Messages, a bude s týmto vysielať dát pokračovať, až kým nenastane jedna z nasledujúcich situácií:

- Študent vysielanie ukončí opätovným stlačením tlačidla
- Študentovi sa zmení stav dochádzky na ‘přítomen’
- Validita dochádzkového tokenu skončila



Obr. 5.4: Ukážka evidencie študenta pomocou BLE

5.3.1 Obmedzenia

Pri používaní BLE ako formy evidencie dochádzky sa môže vyskytnúť viacero problémov, ktoré nie je možné riešiť. Výskyt týchto problémov môže byť spôsobené napríklad tým, že mobilné zariadenie učiteľa alebo študenta nemusí podporovať Bluetooth 4.0+ alebo môže byť Bluetooth čip poškodený, čo bráni správnej funkcionalite. Okrem toho sa môžu vyskytnúť aj problémy s kompatibilitou, ako na zariadeniach s operačným systémom iOS, ktoré si vyžadujú úpravu zdrojového kódu knižnice s podporou Nearby Messages.

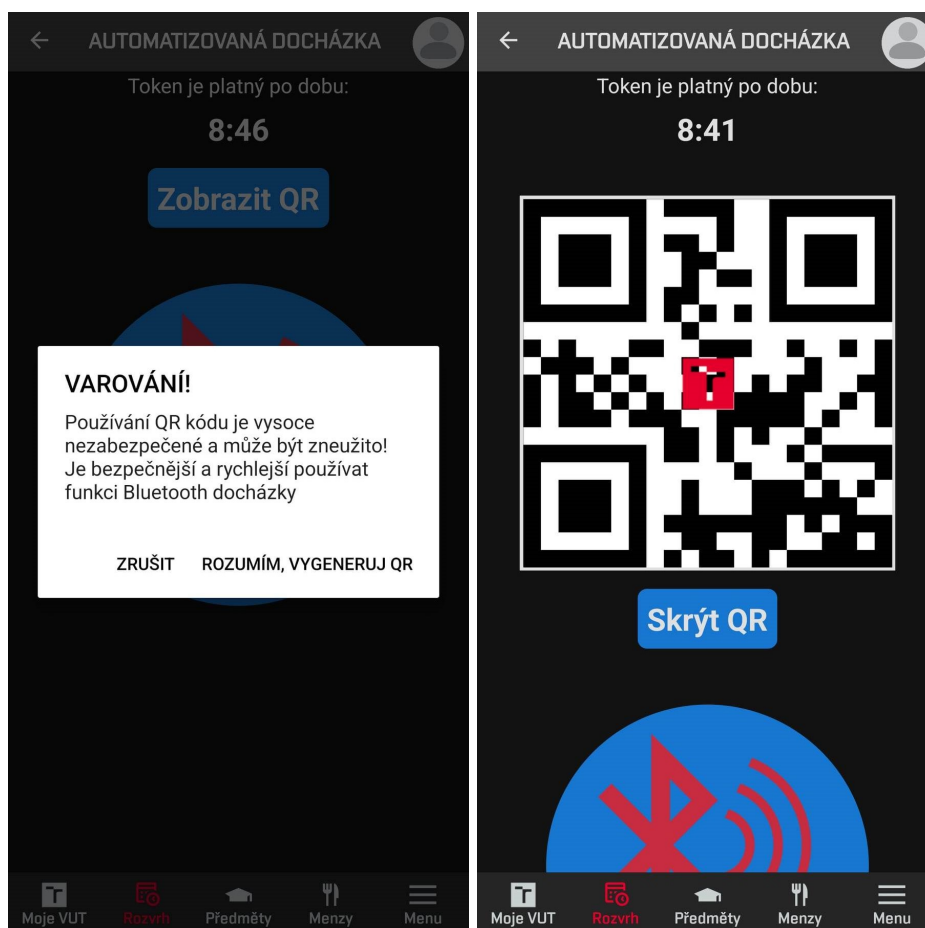
Azda najväčší problém spojený s implementáciou technológie BLE do aplikácie Moje VUT je nedostatočná správa práv pre technológiu Nearby Messages v systéme Android. Klasické vyskakovacie okno s možnosťou povolenia alebo odmietnutia práv nie je pri tejto technológii k dispozícii, namiesto toho sa zobrazí iba upozornenie v zozname notifikácií, ktoré užívateľ musí potvrdiť. Technológia Nearby Messages nie je zabudovaná priamo v zariadení, ale v nainštalovanej aplikácii Google Play Services, a preto sa nezobrazuje v zozname povolených práv. Tento nedostatok môže spôsobiť problémy v prípade, že užívateľ neumožní notifikácie pre aplikáciu Moje VUT a teda nebude vedieť, prečo BLE evidencia dochádzky nefunguje. Taktiež žiaľ neexistuje spôsob, ako zistiť, či bola technológia Nearby Messages pre aplikáciu povolená alebo nie, a teda sme odkázaný na užívateľovu ochotu potvrdiť notifikačné práva.

5.4 Evidencia dochádzky pomocou QR

Implementácia funkcionality QR vyžaduje riešiť menej problémov, ako technológia BLE, a preto bola pomerne nenáročná.

Vyučujúci

Pre evidovanie dochádzky pomocou QR kódu je nevyhnutné vytvoriť QR kód, ktorého generáciu zabezpečuje komponenta **QRCodeGenerator**. Táto komponenta dostane na vstupe hodnotu, konkrétne dochádzkový token, ktorý je získaný zo serveru (PATCH) a použije ho na vytvorenie QR kódu. Tento kód môže byť zobrazený a skrytý v ľubovoľnom čase, pri jeho zobrazení sa však zobrazí upozornenie o možnosti zneužitia kódu inými študentmi, ktorí sa v danej chvíli nezúčastňujú vyučovanej hodiny. Po odsúhlasení možného rizika QR kódu sa zobrazí a študenti ho následne môžu naskenovať. QR kód sa automaticky skryje, ak platnosť dochádzkového tokenu vypršala, keďže už by nebol validný a teda by nebolo možné si vďaka nemu overiť dochádzku.

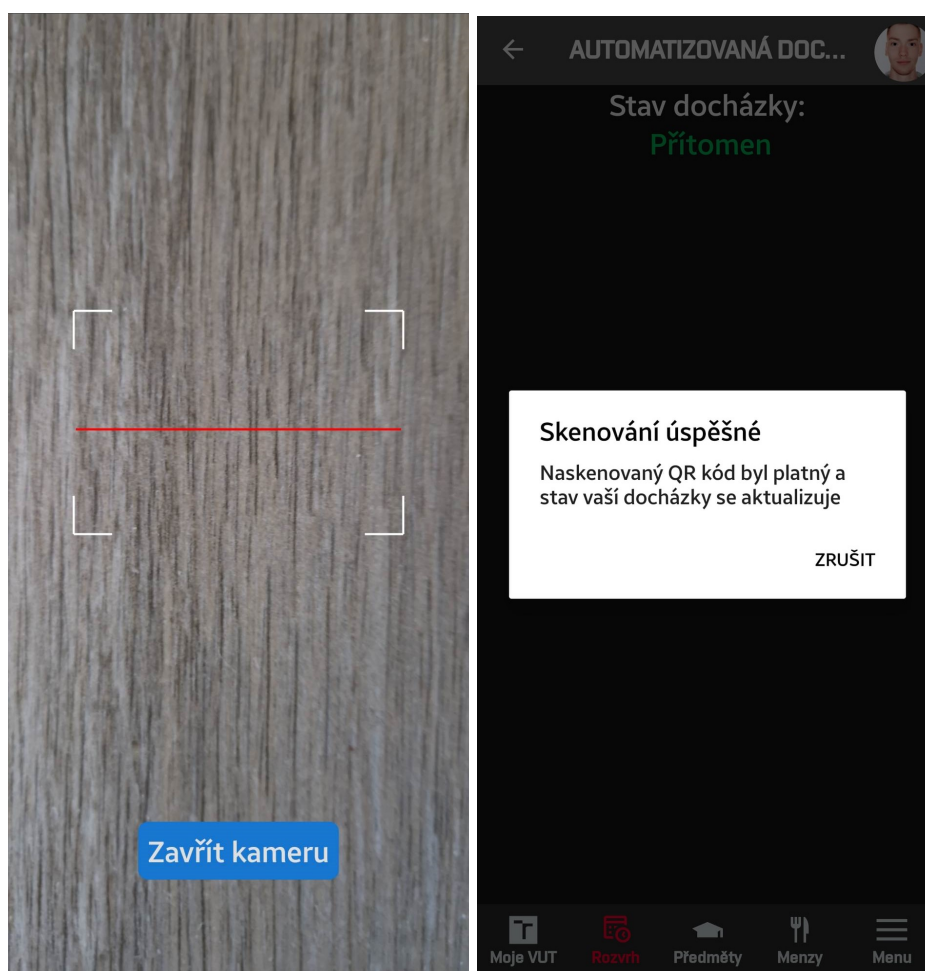


Obr. 5.5: Zobrazenie QR kódu v mobilnom zariadení vyučujúceho

Študent

Aby bol študent schopný QR kód naskenovať, bol vytvorený komponent **QRCodeScanner**. Tento komponent prekreslí všetky ostatné komponenty na obrazovke a zobrazí obraz zo zadnej kamery zariadenia. Ak študent po prvýkrát spustí kameru na skenovanie QR kódu, zobrazí sa mu kontextové okno s požiadavkou o prístup k kamere. Ak študent neumožní prístup k kamere, nebude možné ju použiť a pri ďalšom pokuse o skenovanie QR kódu sa opäť zobrazí toto kontextové okno.

Aplikácia analyzuje každý snímok z kamery v reálnom čase, kým nenatrafí na platný QR kód. Ak sa QR kód naskenuje, aplikácia ho dekóduje, získa z neho dáta a overí, či sú v očakávanom formáte dochádzkového tokenu. Ak dáta nevyhovujú očakávanému formátu, aplikácia upozorní študenta na neplatný QR kód. Ak sú dáta v platnom formáte, aplikácia odošle požiadavok (PATCH) na server za účelom overenia. Ak server potvrdí platnosť, aktualizuje študentovu dochádzku a odpovedá mu novým stavom jeho dochádzky.



Obr. 5.6: Úspěšné skenování QR kódu v mobilnom zariadení študenta

Kapitola 6

Testovanie

Testovanie implementovaných funkcionalít do aplikácie Moje VUT je dôležitou súčasťou vývoja. V prípade automatizovanej evidencie dochádzky študentov, testovanie je nevyhnutné pre overenie spoľahlivosti a efektívnosti implementovaného systému. V prvom rade je dôležité sa zamerať na testovanie intuitívnosti aplikácie a následne jej funkčnosti, aby sme overili, či implementované technológie splňujú očakávania. Následne sa bude aplikácia testovať priamo v praxi na vyučovacích hodinách s vyšším počtom študentov, aby sa zaručila stabilita a efektivita implementovanej technológie.

Keďže testovanie aplikácie zahŕňa zber dát, bol pre tento účel v aplikácii Moje VUT použitý nástroj **Firebase** od spoločnosti Google. Firebase poskytuje užitočnú funkcionálnu pre získavanie informácií o používaní a chybách v aplikácii. Tieto služby sa nazývajú **Firebase Crashlytics** a **Firebase Analytics**.

Firebase Crashlytics je nástroj, ktorý pomáha identifikovať a opraviť chyby v aplikácii. Ak sa v aplikácii vyskytne chyba, Crashlytics ju automaticky zachytí a pošle jej detaily na centralizovaný Google server. Tieto informácie pomáhajú identifikovať príčinu chyby, aby ju bolo možné pri nasledujúcom iteratívnom vylepšovaní aplikácie opraviť. Okrem zberu informácií o chybách poskytuje Firebase aj nástroj na sledovanie používania aplikácie s názvom **Firebase Analytics**. Táto služba umožňuje zistiť, ako používatelia interagujú s aplikáciou, ktoré funkcie aplikácie sa najviac používajú, a ktoré by sa mali vylepšiť. Sledovanie používania aplikácie nám pomáha lepšie pochopiť správanie užívateľov a zlepšiť užívateľskú skúsenosť.

6.1 Distribúcia aplikácie

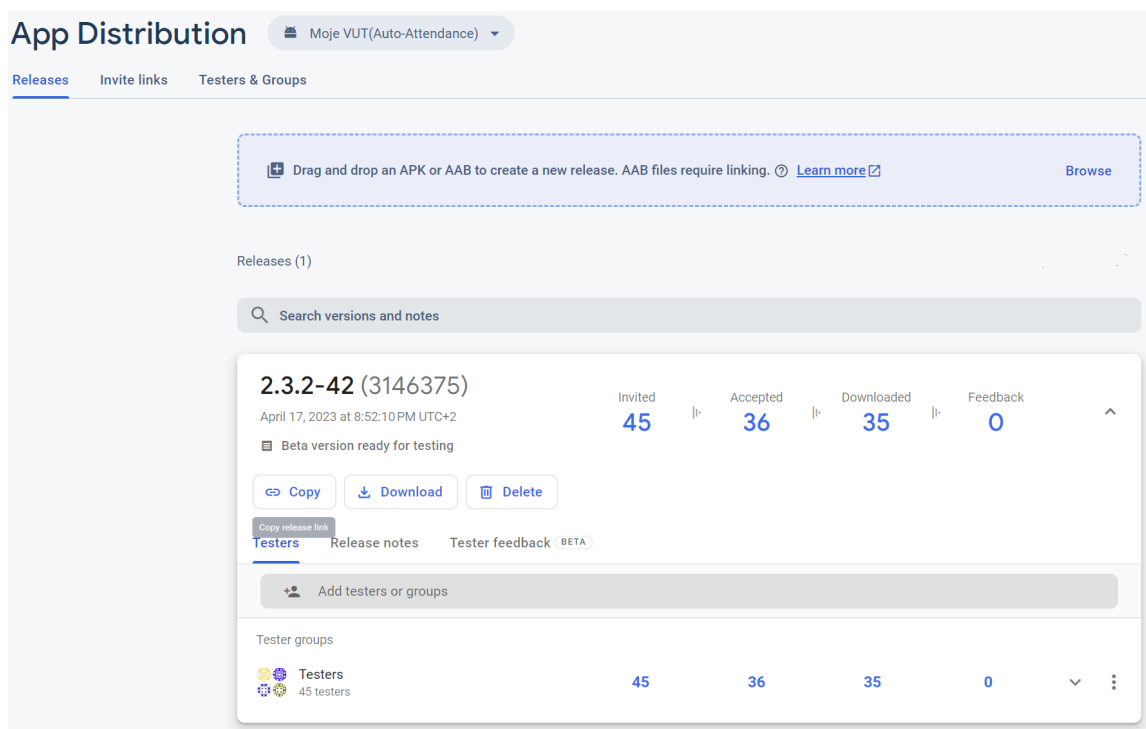
Keďže táto modernizácia ešte nie je zahrnutá v oficiálnej verzii aplikácie Moje VUT, je potrebné zaistiť dostupnú distribúciu samostatne spustiteľnej aplikácie za účelom testovania. Na tento účel bol použitý nástroj **Firebase App Distribution**, ktorý umožňuje distribúciu aplikácie medzi testovacími užívateľmi bez nutnosti zverejnenia aplikácie na oficiálne obchody *Google Play* pre Android alebo *App Store* pre iOS.

Najprv je však potrebné vytvoriť spustiteľný súbor demo aplikácie. Za týmto účelom je možné použiť zabudovaný príkaz “`gradlew assembleDebug`”. Gradlew je zostavovací systém s otvoreným zdrojovým kódom, ktorý slúži na zostavenie zdrojových súborov aplikácie do spustiteľného APK súboru, ktorý môže byť následne nainštalovaný na zariadenie s operačným systémom Android. Nakoľko sa jedná iba o testovaciu demo verziu, nie je potrebné aplikáciu podpisovať a zostavovať ako *release*.

Proces distribúcie aplikácie pomocou Firebase App Distribution je jednoduchý a spočíva v niekoľkých krokoch:

- Vytvorenie účtu na stránke <https://console.firebase.google.com/>
- Vytvorenie nového projektu
- Nahranie spustiteľného súboru APK
- Vytvorenie testovacej skupiny
- Zdieľať distribučný odkaz tejto testovacej skupiny

Po zdieľaní odkazu¹ budú mať užívatelia možnosť stiahnuť demo verziu aplikácie za účelom testovania. Budú automaticky pridaný do vytvorenej testovacej skupiny, čo zaručí prehľadnejšie sledovanie dát o používaní aplikácie. Po finálnych fázach testovania sa počet univerzálnych stiahnutí testovacej aplikácie dostal na **35**.



Obr. 6.1: Distribuovaná testovacia aplikácia v Firebase

¹<https://appdistribution.firebase.dev/i/c776c0cd2232e16d>

6.2 Testovanie v reálnom prostredí

Aby sa zaručila intuitívnosť aplikácie a správna implementácia navrhnutej funkcionality, je potrebné, aby aplikáciu otestovali priamo zamestnanci a študenti VUT v reálnom prostredí. Keďže ale mnoho študentov a zamestnancov používa oficiálnu mobilnú aplikáciu Moje VUT, alebo ju majú aspoň nainštalovanú vo svojom zariadení, bolo potrebné taktiež zmeniť názov a balík aplikácie za účelom testovania demo verzie, aby užívatelia nemuseli oficiálnu aplikáciu odinštalovať. Z tohto dôvodu bol názov aplikácie a jej balík zmenený na **Moje VUT (Auto-Attendance)**.

Toto testovanie prebiehalo dvoma spôsobmi:

1. Distribučný odkaz testovacej aplikácie bol rozposlaný zamestnancom VUT, ktorí prejavili záujem o bližšie informácie vývoja v prieskume (3.1) za účelom zhodnotenia intuitívnosti aplikácie. Bol im taktiež zaslaný dotazník, v ktorom hodnotili intuitívnosť implementácie.
2. Po dohode s vyučujúcim bola demo verzia aplikácie otestovaná priamo na vyučovanej hodine, či už na cvičení alebo prednáške. Týmto spôsobom bolo zabezpečené testovanie nielen intuitívnosti aplikácie zo strany vyučujúceho a študentov, ale aj funkčnosti implementácie a efektívnosti v porovnaní s existujúcimi riešeniami pre evidenciu dochádzky. Okrem toho bolo treba aj zistiť, či implementované technológie zvládnu vyšší nápor užívateľov, pretože počas vývoja bol k dispozícii iba jeden učiteľský a jeden študentský testovací účet.

Bohužiaľ, pri testovaní formou popísanou v bode 1, iba dvaja zamestnanci VUT odpovedali na dotazník prieskumu intuitívnosti užívateľského rozhrania, čo je nedostatočný počet na zhodnotenie kvality implementácie. Oba respondenti však označili navigáciu do modulu automatizovanej evidencie ako intuitívnu, rovnako ako aj interakciu s ním. Prvý respondent ohodnotil intuitívnosť na stupnici 0-5 ako 1, druhý ako 0, čo naznačuje, že aplikácia je dostatočne intuitívna.

Na druhej strane, testovanie formou popísanou v bode 2 bolo veľmi úspešné a prínosné. Prvé veľké testovanie sa uskutočnilo na prednáške hodiny *Tvorba webových stránok*, kde testovaciu aplikáciu nainštaloval do svojho mobilného zariadenia vyučujúci a 17 študentov prítomných na prednáške. Následne, po krátkom vysvetlení funkcionality aplikácie, vyučujúci zahájil automatizovanú evidenciu a v priebehu pár desiatok sekúnd po tom, ako študenti zahájili vysielanie svojho identifikátora pomocou BLE (5.2), bolo úspešne a bez problémov zaevidovaných 16 študentov. Jeden študent však nebol zaznamenaný touto technológiou, ale po chvíli sa zistilo, že to bolo spôsobené obmedzenou správou práv popísaným v 5.3.1. Po vykonaní tohto testovania funkčnosti a efektívnosti implementovanej evidencie dochádzky boli študenti a vyučujúci požiadaní o vyplnenie krátkeho dotazníka s cieľom analyzovať spokojnosť a intuitívnosť aplikácie. Bohužiaľ, na tento dotazník odpovedal iba jeden študent, ktorý však označil, že používanie automatizovanej evidencie je veľmi intuitívne. Podľa jeho slov by však nechcel vidieť túto automatizovanú formu evidencie dochádzok v oficiálnej aplikácii Moje VUT, pretože by to, podľa jeho slov, zťažilo alebo znemožnilo falšovanie dochádzky študentov. Môžeme však predpokladať, že študent to nemyslel vážne a išlo len o vtip.

Druhé testovanie touto formou prebiehalo na cvičení z predmetu *Matematická analýza 2*, kde testovaciu aplikáciu stiahlo 14 študentov. Avšak, pri zahajovaní dochádzky a overaní študentských identifikátorov sa vyskytli problémy, konkrétne server odmietal vytvoriť dochádzkový token z dôvodu zlej kontroly práv vyučujúcich a chybného porovnávania z tabuľky zamestnancov a študentov VUT na strane servera. Tento problém sa podarilo opraviť až na konci cvičenia, kedy väčšina študentov už opustila miestnosť. Namiesto evidencie pomocou BLE sa teda otestovala iba evidencia pomocou QR (5.4) s tromi študentami, ktorá prebehla úspešne a bez akýchkoľvek problémov. Po skončení testovania bola s vyučujúcim diskutovaná intuitívnosť aplikácie, ktorý priznal, že najprv aplikácii moc nerozumel, ale po krátkom vysvetlení povedal, že už mu všetko dáva zmysel a je to zrozumiteľné.

6.3 Budúci vývoj

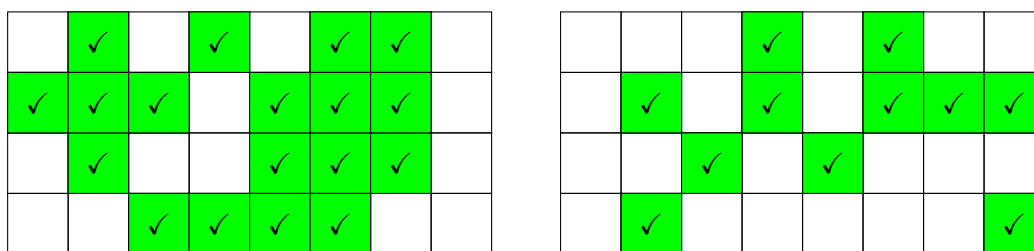
Spätná väzba vyučujúcich a študentov poskytla mnoho možných vylepšení, ktoré by spravili dochádzkový modul aplikácie Moje VUT omnoho použiteľnejším.

Plán učebne a rozmiestnenie študentov

Toto rozšírenie bolo navrhnuté niekoľkými zamestnancami VUT s cieľom zlepšiť identifikáciu študentov v dochádzkovom module aplikácie. Hlavnou výhodou papierovej formy evidencie je, že umožňuje vyučujúcemu vidieť, kde presne študent sedí. Žiaľ, aplikácia Moje VUT túto funkcionality nemá, čo môže spôsobiť problémy pri identifikácii študentov, ktorých mená vyučujúci nepozná.

Existuje však možnosť integrovať do dochádzkového modulu databázu všetkých učební a ich miest na sedenie, ktoré už existujú napríklad na fakulte informačných technológií VUT. Študent by si tak mohol vybrať svoje miesto a vyučujúci by videl kde sedí priamo v aplikácii. Avšak nie všetky fakulty majú takéto dáta dostupné, a preto by bolo potrebné ich najprv vytvoriť.

Ďalšou možnosťou by mohlo byť navrhnutie funkcie vytvárania približného plánu učebne pre vyučujúceho, napríklad pridaním možnosti vytvorenia mriežky reprezentujúcej možné miesta na sedenie.



Kompatibilita s iOS

V aktuálnej fázi vývoja nie je implementovaná funkcionality kompatibilná s operačným systémom iOS z dôvodu konfliktnej knižnice a chybe pri zostavení aplikácie. Táto chyba je však riešiteľná a bude označená ako prioritná, keďže aplikácia Moje VUT je určená pre oba operačné systémy, teda Android aj iOS.

Zvýšení přehlednosti statusu docházky studenta

V současné implementácii musí študent otvoriť detail vyučovacieho predmetu, aby získal informáciu o stave svojej dochádzky. Tento postup môže byť zbytočne zdĺhavý a náročný, ak si chce študent skontrolovať stav dochádzky na viacerých vyučovacích hodinách. Vzhľadom na tieto nedostatky by bolo vhodné implementovať grafický ukazovateľ stavu dochádzky, ktorý by bol zobrazený priamo v rozvrhu predmetov.

	Po 10.10.	Út 11.10.	St 12.10.	Čt 13.10.	Pá 14.10.
7 ⁰⁰					
8 ⁰⁰		Cvičení ... B/N2...	Cvičení ... B/N2...		
9 ⁰⁰		ITU Tvorba už ivatelských rozhraní	ITU Tvorba už ivatelských rozhraní		
10 ⁰⁰	Cvičení ... B/N2...	Cvičení ... B/N2...	Cvičení ... B/N2...		
11 ⁰⁰	ITU Tvorba už ivatelských rozhraní	ITU Tvorba už ivatelských rozhraní	ITU Tvorba už ivatelských rozhraní		
12 ⁰⁰	Cvičení ... B/N2...	Cvičení ... B/N2...		Přednáška B/M1...	
13 ⁰⁰	ITU Tvorba už ivatelských rozhraní	ITU Tvorba už ivatelských rozhraní		ITUe Tvorba už ivatelských rozhraní...	Přednáška B/M1...
14 ⁰⁰	Cvičení ... B/N2...	Cp		ITU Tvorba už ivatelských rozhraní	ITUe Tvorba už ivatelských rozhraní...
15 ⁰⁰	ITU Tvorba už ivatelských rozhraní	ITU	S		
16 ⁰⁰	Cvičení ... B/N2...	Cp			
17 ⁰⁰	ITU Tvorba už ivatelských rozhraní	ITU			

Obr. 6.2: Návrh grafického zobrazenia stavu dochádzky študenta

Zmena implementovanej technológie

Spoločnosť Google nečakane oznámila, že koncom decembra 2023 ukončia podporu technológie Nearby Messages (3.3.1), čo bude mať za následok jej nepoužitelnosť, preto bude potrebné zvoliť inú technológiu, napríklad Nearby Connections (3.3.1), ktorá však vyžaduje náročnejšiu implementáciu z dôvodu neexistujúcich React Native knižníc, preto ak by sa táto technológia mala používať, bude nutné vytvoriť jej podporu v podobe novej knižnice pre React Native.

Stručný návod

Vzhľadom na to, že implementované užívateľské rozhranie nebolo pre všetkých užívateľov okamžite intuitívne, bolo by vhodné integrovať do aplikácie návod na používanie automatizovanej evidencie dochádzky. Tento návod by poskytol užívateľovi potrebné informácie o funkcionalite tejto formy evidencie.

Zabezpečenie proti zneužitiu automatizovanej evidencie pri zmene účtu

Okrem problému druhého zariadenia, ktorý bol opísaný v sekcii 4.2.1, tu existuje aj riziko, že študent sa po úspešnej evidencii prihlási na iný účet, napríklad na účet svojho spolužiaka, ktorý na hodine nie je prítomný, a použije toto zariadenie na jeho evidenciu. Tento problém sa nedá ľahko riešiť, pretože študent môže vlastniť viacero mobilných zariadení, alebo môže mať zariadenie požičané.

Jedným z navrhnutých riešení bolo obmedzenie možnosti použitia automatizovanej evidencie v prípade zmeny účtu v zariadení, no toto opatrenie by bolo možné obísť tak, že študent jednoducho odinštaluje aplikáciu, čím odstráni všetky uložené dáta, a následne ju znova nainštaluje. Preto by overovanie muselo byť implementované na strane servera, čo by si vyžadovalo zbieranie dát, napríklad MAC adresy mobilných zariadení, a ich priradenie konkrétnym študentom. Takáto implementácia by však mohla byť pomerne náročná a z hľadiska GDPR diskutabilná, keďže MAC adresa sa považuje za osobnú informáciu a na jej spracovanie by užívateľ musel poskytnúť súhlas.

6.4 Zhrnutie

Po testovaní (6.2) a spätnej väzbe môžeme usúdiť, že implementovaná funkcionalita nie len splňuje očakávané kritériá automatizovanej evidencie dochádzky, ale je aj pomerne intuitívna na používanie. Testovanie odhalilo rôzne nedostatky, ktoré boli opravené alebo sa ešte musia doladiť, a poskytlo množstvo odporúčaní na budúce vylepšenia a zdokonalenia aplikácie. Zároveň je z testovania zrejmé, že evidencia dochádzky pomocou BLE je menej časovo a námahovo náročná ako iné formy evidencie, čo umožňuje vyučujúcemu viac sa venovať vyučovaniu a uľahčuje jeho administratívnu prácu.

Kapitola 7

Záver

V rámci tejto bakalárskej práce bola vyvinutá a integrovaná funkcionálna automatizovanej evidencie dochádzky študentov do mobilnej aplikácie Moje VUT, ktorá využíva moderné technológie pre evidenciu prítomnosti študentov na vyučovacích hodinách. Toto rozšírenie bolo implementované pomocou multiplatformového frameworku React Native s použitím externých knižníc. Cieľom implementácie bolo zlepšiť efektivitu a zjednodušiť proces administratívnej práce vyučujúceho, ktorý zaznamenáva dochádzku študentov, a tým mu umožniť venovať sa viac vyučovaniu.

Na úvod bol skúmaný záujem o túto funkcionálnu medzi zamestnancami VUT formou dotazníka a následne prebehla analýza súčasných foriem evidencie dochádzky naprieč rôznymi fakultami VUT. Výsledky ukázali, že najväčší záujem o túto funkcionálnu prejavujú vyučujúci prednášok s vysokým počtom študentov, pretože manuálna evidencia dochádzky v takejto situácii je časovo náročná. Taktiež bolo získaných mnoho cenných odporúčaní, ktoré boli použité pri návrhu a následnej implementácii.

Ďalším krokom bolo vytvorenie návrhov funkcionality a užívateľského rozhrania. Po analýze a testovaní dostupných technológií na mobilných zariadeniach bolo zistené, že technológia *Bluetooth Low Energy* spĺňa všetky požadované kritériá pre efektívnu evidenciu dochádzky a preto bola zvolená ako primárna technológia. S ohľadom na možné neočakávané problémy s touto technológiou bola navrhnutá aj sekundárna technológia *QR kódov*, ktorá predstavuje rýchlu a jednoduchú alternatívu pre evidenciu dochádzky. Obe technológie boli navrhnuté tak, aby boli čo najviac bezpečné, spoľahlivé a minimalizovali spotrebu batérie mobilného zariadenia.

Implementovaná funkcionálna v aplikácii Moje VUT umožňuje vyučujúcemu zahájiť evidenciu dochádzky na určitú časovú dĺžku, čím sa zaručí, že študent nebude schopný si dochádzku zaevidovať v iný čas v prípade skoršieho alebo neskoršieho príchodu na vyučovaciu hodinu. Okrem rýchlej formy evidencie dochádzky umožňuje aplikácia študentovi skontrolovať svoj stav dochádzky pre danú hodinu priamo v aplikácii, čím sa zabezpečuje lepšia prehľadnosť evidencie.

Testovanie rozšírenia aplikácie prebehlo dvoma spôsobmi. Intuitívnosť užívateľského rozhrania bola overená poslaním distribučného odkazu testovacej aplikácie rôznym zamestnancom VUT, ktorí aplikáciu nainštalovali a sami sa pokúsili pochopiť, ako sa používa a ako funguje. Následne vyplnili dotazník s spätnou väzbou a návrhmi na vylepšenie. Funkcionálna bola testovaná priamo na vyučovacích hodinách na fakultách VUT, kde ju testovali študenti aj vyučujúci. Toto testovanie odhalilo niekoľko drobných chýb, ktoré boli odstránené a aplikácia sa stala spoľahlivejšou. Taktiež bola otestovaná efektivita tejto formy evidencie dochádzky v porovnaní s inými formami, ako napríklad s evidenciou na papierovú

dochádzkovú listinu, s pozitívnymi výsledkami. Celkovo boli užívatelia s implementáciou automatizovanej evidencie dochádzky spokojný a vyjadrili záujem o integráciu tejto funkcionality do oficiálnej aplikácie.

Hoci má táto implementácia pozitívnu spätnú väzbu, existuje stále veľa možností, ako ju vylepšiť a veľa nedostatkov, ktoré je treba vyriešiť. Práca na tomto rozšírení bola veľmi zaujímavá a poskytla mi mnoho cenných poznatkov o vývoji mobilných aplikácií. Preto plánujem na tomto rozšírení pracovať aj naďalej a pokračovať v jej vývoji, s cieľom úspešne ju integrovať do oficiálnej mobilnej aplikácie Moje VUT.

Literatúra

- [1] AMAZON. *What Is A RESTful API?* [online]. Dostupné z: <https://aws.amazon.com/what-is/restful-api/>.
- [2] ANDROID. *Host-based card emulation overview* [online]. Dostupné z: <https://developer.android.com/guide/topics/connectivity/nfc/hce#secure-element>.
- [3] ARUN, M. *Getting Started With BLE – A Complete Guide* [online]. Dostupné z: <https://www.mindbrowser.com/getting-started-with-ble/>.
- [4] BANSAL, S. *Nearby Connection API* [online]. Dostupné z: <https://proandroiddev.com/nearby-connection-api-b235529e6643>.
- [5] BISDIKIAN, C. An overview of the Bluetooth wireless technology. *IEEE Communications Magazine* [online]. Citeseer. 2001, zv. 39, č. 12. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/968817>.
- [6] BRAY, J. a STURMAN, C. F. *Bluetooth Technology* [online]. 2020. Dostupné z: <https://ece.uwaterloo.ca/~snaik/750/bt.pdf>.
- [7] DS300. *Patch-package* [online]. Dostupné z: <https://github.com/ds300/patch-package>.
- [8] DWIVEDI, H., CLARK, C., THIEL, D. V. et al. *Mobile application security*. McGraw-Hill New York, 2010. ISBN 978-0-07-163357-4.
- [9] EL SEOUD, M. S. A. a TAJ EDDIN, I. A. *Developing an android mobile bluetooth chat messenger as an interactive and collaborative learning aid*. 2. vyd. Springer, 2017. 18–30 s. ISBN 978-3-319-50339-4.
- [10] ELAZAB, M. *Integrated Cooperative Localization in VANETs for GPS Denied Environments*. Dizertačná práca.
- [11] FULBER GARCIA, V. *HTTP: Differences Between PUT and PATCH* [online]. Dostupné z: <https://www.baeldung.com/cs/http-put-vs-patch>.
- [12] GOOGLE. *Nearby Connections API Overview* [online]. Dostupné z: <https://developers.google.com/nearby/connections/overview>.
- [13] GOOGLE. *Nearby Messages API Overview* [online]. Dostupné z: <https://developers.google.com/nearby/messages/overview>.
- [14] HEALTH, I. M. of. *Rn-contact-tracing* [online]. Dostupné z: <https://github.com/MohGovIL/rn-contact-tracing>.

- [15] HEYDON, R. *Bluetooth Low Energy - The Developer's Handbook*. 1. vyd. Pearson Education, Inc, 2012. ISBN 978-0-13-288836-3.
- [16] INNOVEIT. *React-native-ble-manager* [online]. Dostupné z: <https://github.com/innoveit/react-native-ble-manager>.
- [17] INTENT. *React-native-ble-plx* [online]. Dostupné z: <https://github.com/dotintent/react-native-ble-plx>.
- [18] JERRYSHEN. *React-native-qrcode-svg* [online]. Dostupné z: <https://github.com/awesomejerry/react-native-qrcode-svg>.
- [19] KLINOVSKÝ, O. *Mobilní aplikace pro studenty VUT*. 2020. Bakalářská práce. VUT FIT, Vedúci práce Prof. Ing. Tomáš Hruška, CSc.
- [20] KUITUNEN, M. *Cross-Platform Mobile Application Development with React Native*. Tampere, FIN, 2019. Bakalářská práce. Tampere University of Technology. Dostupné z: <https://trepo.tuni.fi/handle/123456789/27139>.
- [21] MANSION, S. *React-native-reanimated* [online]. Dostupné z: <https://github.com/software-mansion/react-native-reanimated>.
- [22] MARSH, J. *UX for Beginners: A Crash Course in 100 Short Lessons*. 1. vyd. O'Reilly Media, Inc., 2015. ISBN 978-1-491-91268-3.
- [23] MASALHA, F. a HIRZALLAH, N. A students attendance system using QR code. *International Journal of Advanced Computer Science and Applications* [online]. Citeseer. 2014, zv. 5, č. 3. Dostupné z: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=288f0459675d41e2d3bbb8b6b65bc927ffe57262>.
- [24] NATIVE, R. *React Native, A framework for building native apps using React* [online]. Dostupné z: <https://reactnative.dev/>.
- [25] OCETNÍK, D. *React-native-background-timer* [online]. Dostupné z: <https://github.com/ocetnik/react-native-background-timer>.
- [26] ROUSAVY, M. *React-native-google-nearby-messages* [online]. Dostupné z: <https://github.com/mrousavy/react-native-google-nearby-messages>.
- [27] ROUSAVY, M. *React-native-vision-camera* [online]. Dostupné z: <https://github.com/mrousavy/react-native-vision-camera>.
- [28] SIDAT, M. *React-native-qrcode-scanner* [online]. Dostupné z: <https://github.com/moaazsidat/react-native-qrcode-scanner>.
- [29] SINGH, S. *Native vs Hybrid vs Cross Platform – What to Choose in 2023?* [online]. Dostupné z: <https://www.netsolutions.com/insights/native-vs-hybrid-vs-cross-platform/>.
- [30] SOLUTIONS, K. *Mobile Application Development: A Comprehensive Guide* [online]. Dostupné z: <https://blog.kms-solutions.asia/an-ultimate-guide-to-mobile-application-development>.

- [31] TESLAMOTORS. *React-native-camera-kit* [online]. Dostupné z: <https://github.com/teslamotors/react-native-camera-kit>.
- [32] TIDWELL, J. *Designing interfaces: Patterns for effective interaction design*. 2. vyd. O'Reilly Media, Inc., 2010. ISBN 978-1-449-37970-4.
- [33] WIKIPEDIA. *Android (operating system)* [online]. Dostupné z: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)).
- [34] WIKIPEDIA. *Global Positioning System* [online]. Dostupné z: https://en.wikipedia.org/wiki/Global_Positioning_System.
- [35] WIKIPEDIA. *IOS* [online]. Dostupné z: <https://en.wikipedia.org/wiki/IOS>.
- [36] WIKIPEDIA. *Near Field Communication* [online]. Dostupné z: https://en.wikipedia.org/wiki/Near_Field_Communication.
- [37] WIKIPEDIA. *QR code* [online]. Dostupné z: https://en.wikipedia.org/wiki/QR_code.
- [38] WORLD rfwireless. *BLE Advertising channels / BLE Data channels list* [online]. Dostupné z: <https://www.rfwireless-world.com/Terminology/BLE-Advertising-channels-and-Data-channels-list.html>.
- [39] WU, W. *React Native vs Flutter, cross-platform mobile application frameworks*. Helsinki, FIN, 2018. Diplomová práca. Metropolia University of Applied Sciences. Dostupné z: <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf>.
- [40] ZACHARIAS, S. a NEWE, T. Technologies and Architectures for Multimedia-Support in Wireless Sensor Network. In: 2010. ISBN 978-953-307-261-6.