

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-XXXX-XXXXX

Šimon Drienik

Zefektívnenie verejnej správy s využitím moderných aplikácií

Bakalárska práca 1

Vedúci záverečnej práce: Ing. Tomáš Kováčik, PhD.

Január 2021

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-XXXX-XXXXX

Šimon Drienik

**Zefektívnenie verejnej správy s využitím
moderných aplikácií**

Bakalárska práca 1

Študijný program: Informatika

Študijný odbor: 9.2.1 Informatika

Školiace pracovisko: Fakulta informatiky a informačných technológií , FIIT STU,
Bratislava

Vedúci záverečnej práce: Ing. Tomáš Kováčik, PhD.

Január 2021

ZADANIE BAKALÁRSKEHO PROJEKTU

Meno študenta: **Drienik Šimon**
Študijný odbor: Informatika
Študijný program: Informatika
Názov projektu: **Zefektívnenie verejnej správy s použitím moderných aplikácií**

Zadanie:

Verejná správa a samospráva sú prostredím, kam inovácie prichádzajú len postupne. Pritom by moderné technológie na najrôznejších úrovniach mohli zjednodušiť, zrýchliť a zlepšiť fungovanie verejnej správy. Príkladom môžu byť správcovia ciest, ktorí nemusia zostať pri papierových zápisníkoch pracovných čát, ale môžu sa posunúť k používaniu moderných aplikácií na zaznamenávanie problémov s poškodenými cestami, značkami, zvodidlami, prerastenou či nebezpečnou zeleňou atď. Hlásenia problémov by mohla zadávať aj verejnosť, aplikácia by mohla pracovať s GIS, v ktorých majú správcovia zaznamenané úseky ciest.

Identifikujte problémy, ktoré správcovia ciest nižších tried riešia. Analyzujte možnosti využitia moderných aplikácií na ich zaznamenávanie, zverejňovanie, pridelovanie pracovníkom na vyriešenie a sledovanie vyriešenia týchto problémov ich nadriadenými. Navrhnite, implementujte a otestujte systém, v ktorom takáto aplikácia zefektívni plánovanie a prácu správcu ciest.

Práca musí obsahovať:

- Anotáciu v slovenskom a anglickom jazyku
- Analýzu problému
- Opis riešenia
- Zhodnotenie
- Technickú dokumentáciu
- Zoznam použitej literatúry
- Elektronické médium obsahujúce vytvorený produkt spolu s dokumentáciou

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU, Bratislava
Vedúci projektu: Ing. Tomáš Kováčik, PhD.

Termín odovzdania práce v zimnom semestri : 7.12.2020

Termín odovzdania práce v letnom semestri : 11.5.2021

Bratislava 21.9.2020

doc. Ing. Valentino Vranič, PhD.
poverený riadením ÚISI

Čestné prehlásenie

V Bratislave, 3.1.2021

.....

Šimon Drienik

Pod'akovanie

Anotácia

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Študijný program: Informatika

Autor: Šimon Drienik

Bakalárska práca 1: Zefektívnenie verejnej správy s využitím moderných aplikácií

Vedúci projektu: Ing. Tomáš Kováčik, PhD.

Január 2021

text

Annotation

Slovak University of Technology in Bratislava

Faculty of informatics and information technologies

Degree Course: Informatics

Author: Šimon Drienik

Bachelor's thesis 1: Streamlining public administration with the use of modern applications

Supervisor: Ing. Tomáš Kováčik, PhD.

January 2021

Obsah

1	Úvod do problematiky	1
1.1	Oči na ceste -mobilná verzia pre Android	2
2	Analýza	5
2.1	Analýza webovej platformy Oči Na Ceste	5
2.1.1	Použité technológie na vývoj	5
2.1.2	Používateľské rozhranie a funkcionality	9
2.1.3	Databáza	13
2.1.4	Stavový model problému	17
2.1.5	Analýza zimplementovaného riešenia	18
2.2	Analýza technológií na vývoj android verzie	22
2.2.1	Programovacie prostredie	22
2.2.2	Programovací jazyk	23
2.2.3	Androidové sieťové knižnice a závislosti	24
2.3	Analýza existujúcich zahraničných riešení	26
2.3.1	WCRC fix it	26
2.3.2	UDOT Click ‘n Fix	28
2.3.3	Find It, Fix It	28
2.4	Záver analýzy	30
3	Návrh riešenia	33

4	Implementácia	35
5	Testovanie	37
6	Záver	39
6.1	Zhodnotenie projektu	40
6.2	Možné vylepšenia	40
A	Harmonogram práce	
B	Digitálne médium	

Zoznam obrázkov

2.1	Tabuľka oprávnení používateľských rolí	9
2.2	Domovská stránka webovej platformy	10
2.3	Stránka vytvorenia hlásenia	11
2.4	Tabuľka všetkých problémov s filtrovaním	13
2.5	Schéma vytvorenej databázy	16
2.6	Stavový model problému	17
2.7	Zoznam hlásení	27
2.8	Mapa hlásení	27
2.9	Mapka pre zameranie markera	29
2.10	Screen hlavného menu	29
2.11	Screen vytvorenia hlásenia	30

Zoznam použitých skratiek

API	Application Programming Interface
PC	Počítač
TTSK	Trnavský samosprávny kraj
PHP	HyperText Preprocessor
HTTP	Hyper-Text Transfer Protocol
SSH	secure shell
XML	eXtensible Markup Language
IMAP	Internet Message Access Protocol
JSON	JavaScript Object Notation
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
MySQL	Maria SQL
ID	Identification
OS	Operating System
POP3	Post Office Protocol, version 3
REST	Representational state transfer
SPZ	Štátna poznávacia značka
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
INT	Integer
PK	Primary key
FK	Foreign key
CSRF	Cross site request forgery
IDE	Integrated Development Environment
NDK	Native Development Kit xxi
POJO	Plain Old Java Object

Kapitola 1

Úvod do problematiky

V dnešnej dobe sa čím ďalej viac využívajú všadeprítomné smartfóny na aplikácie, ktoré nám majú zlepšovať a zjednodušovať život. Ich výhoda je, že používatelia môžu pristupovať k dátam kdekoľvek aj mimo PC. Veľmi často sa aplikácie využívajú aj na vytvorenie mobilných verzií webových aplikácií pre prehľadnejšie používateľské rozhranie na menších zariadeniach a teda mobiloch, poprípade tabletoch. Táto skutočnosť je zameraním tejto práce. Správa a údržba ciest Trnavského samosprávneho kraja zefektívnila spôsob zhromažďovania informácií o Závadách na svojich verejných komunikáciách prostredníctvom webovej aplikácie, kde používatelia môžu zaznamenávať na mape rôzne Závady, na ktoré natrafili. Najlepším spôsobom ako maximálne využiť potenciál tohto systému a tohto nápadu, je dať ľuďom možnosť nahlásiť problém verejnej komunikácie hneď ako ho objaví. Teda prostredníctvom mobilnej aplikácie, ktorá im uľahčí vyhľadávanie miesta na mape prostredníctvom ich aktuálnej polohy. Je viac pravdepodobné, že užívateľ bude mať pri sebe na cestách smartfón ako laptop a preto je zavedenie mobilnej verzie tohto systému na mieste. Áno, dá sa na webovú aplikáciu pristupovať aj prostredníctvom webového vyhľadávača na smartfóne ale je to používateľsky neprívetivé a

neje tam možnosť zobrazenia aktuálnej polohy používateľa. Tieto nahlásené problémy musí riešiť zamestnanec Trnavského samosprávneho kraja, ktorý sa musí na označené miesto dostať. Pomocou mobilnej aplikácie bude pre neho jednoduchšie navigovať sa na miesto nahláseného problému vďaka jeho aktuálnej polohe na mape v mobilnej aplikácii. Daná opisovaná aplikácia bude dostupná pre zariadenia s operačným systémom Android.

1.1 Oči na ceste -mobilná verzia pre Android

Mobilná aplikácia Oči Na Ceste bude spĺňať všetky funkčné a aj nefunkčné požiadavky jej webovej verzie. Mobilná a aj webová aplikácia budú pracovať s tou istou databázou údajov a to čo bude dostupné na webovej aplikácii tak bude dostupné aj na mobilnej aplikácii a zase naopak. Cieľom tejto práce je zlepšenie prístupu do aplikácie aj mimo desktopových zariadení. Aplikácia slúži na zber dát o poškodených cestách verejnej komunikácie v Trnavskom samosprávnom kraji pre lepší prehľad o stave ciest zamestnancom TTSK. Tieto údaje môže pridávať ktokoľvek prostredníctvom webovej alebo mobilnej aplikácie. Údaje môže pridávať aj zaregistrovaný alebo nezaregistrovaný občan, s tým rozdielom, že zaregistrovaný občan môže sledovať stav všetkých svojich nahlásených problémov vo svojom profile. Zaregistrovaný občan bude môcť mazať svoje hlásenia. Zamestnanci TTSK budú môcť editovať hlásenia. Budú môcť pridávať informácie o stave a popise riešenia problému. Na riešenie hlásení je dostupná aj logistika prác zamestnancov. Administrátor bude môcť pridelovať riešenie problémov jednotlivým zamestnancom.

Kapitola 2

Analýza

Kapitola obsahuje následné analýzy. Analýza webovej platformy Oči Na Ceste pozostávajúca z použitých technológií a výsledných funkcionalít, z ktorých pozostáva webová platforma. Analýza existujúcich zahraničných riešení danej problematiky. Analýzy technológií vhodných na použitie pre vývoj mobilnej Android verzie projektu Oči Na Ceste. Použité technológie na vývoj danej aplikácie.

2.1 Analýza webovej platformy Oči Na Ceste

V tejto sekcii sú detailne popísané a vymenované technológie, použité na vývoj webovej platformy Oči Na Ceste, používateľské rozhranie, funkcionality a prípady použitia všetkých užívateľských rolí.

2.1.1 Použité technológie na vývoj

Webová aplikácia beží na SSH serveri s operačným systémom Debian 9, kde boli nainštalované nasledujúce programy. Prostredie Nano na editáciu súborov. Manažér závislostí Composer. Skriptovací jazyk PHP 7.4 s rozšíreniami mbstring a

XML. Databáza na uchovávanie dát MySQL. Web server NGINX a php rámec Laravel na komunikáciu s databázou a aplikáciou. Balík od Laravelu Laravel Auth na prihlasovanie a registráciu užívateľov. Google Maps api pre prácu s markermi

2.1.1.1 Composer

Composer, ako bolo povedané je manažér závislostí v PHP. Deklaruje knižnice v projekte a dokáže ich inštalovať a aktualizovať. Zaoberá sa knižnicami a inštaluje ich do adresárov na základe vášho projektu. Tento program zistí, ktoré verzie balíkov je potrebné nainštalovať a následne ich nainštaluje.

2.1.1.2 PHP 7.4 s rozšírením mbstring a XML

PHP v celom názve Hypertext Preprocessor je široko používaný skriptovací jazyk, špeciálne navrhnutý pre webové aplikácie a môže byť jednoducho implementovaný do html. PHP je používaný najmä na server-side skriptovanie. Jednou z najvýznamnejších funkcií PHP je jeho podpora pre širokú škálu databáz. V tomto prípade sa využíva databáza MySQL a s využitím PHP je jednoduché vytvoriť webovú aplikáciu prepojenú s databázou MySQL. PHP funguje ako komunikátor medzi databázou a webovou aplikáciou a pre túto úlohu je najlepšou voľbou, tak ako vo webovej platforme nášho Projektu.

MBstring poskytuje viac bajtové reťazcové funkcie, ktoré sú vhodné pri práci s viac bajtovým kódovaním v PHP. Zvládne prevod kódovania znakov medzi dvojicami kódovania. Je navrhnutý tak, aby spracovával kódovanie na báze Unicode.

Veľa krát, keď píšeme kód v PHP vyžadujeme reťazec napísaný v XML. Rozšírenie XML pre PHP nám umožňuje uľahčiť si prácu tým, že namiesto stáleho zapisovania celého XML reťazca znova a znova, nám tento XML reťazec uloží do súboru, ktorý jednoducho len zahrnieme do každého príkladu kde je potrebný.

2.1.1.3 MySQL

MySQL je otvorený systém riadenia relačnej databázy s modelom klient-server. Je to služba používaná na vytváranie a správu databáz na základe relačného modelu. Databázový systém na uchovávanie dát na serveri od spoločnosti Oracle. V tomto prípade sa využíva na správu, manipuláciu a ukladanie informácií o užívateľoch a záznamoch o cestných chybách, teda markeroch.

2.1.1.4 Web server NGINX

Open-source softvér NGINX je serverový systém pre webové služby, reverzné proxy, ukladanie do vyrovnávajúcej pamäte, vyrovnávanie zaťaženia, streamovanie médií a mnoho iného. Funguje ako funkcia HTTP servera, funkcia servera proxy pre e-mail (IMAP, POP3 a SMTP) a reverzný server proxy a nástroj na vyrovnávanie záťaže pre servery HTTP, TCP a UDP.

2.1.1.5 LARAVEL

Laravel je najpoužívanější PHP rámec pre budovanie akýchkoľvek webových aplikácií. Poskytuje veľmi dobrú API, ktorá je využívaná aj v tomto projekte a je napojená aj na webovú platformu, ktorá beží na rovnakom serveri ako API a aj na Android mobilnú platformu, ktorá beží na zariadeniach užívateľov. V tomto prípade je celá webová platforma postavená na Laraveli vrátane serverovej logiky a používateľského rozhrania. Po zadaní platnej URL na doméne webovej platformy projektu Laravel skontroluje zadané "routes", teda cesty a keď nájde route s danou url a daným Http requestom, tak sa spustí funkcia k nemu priradená "controller". V tomto controlleri sa spracujú požiadavky užívateľa a ako návratová hodnota sa vráti užívateľovi web stránka "view"s požadovanými informáciami získanými z databázy alebo v inom prípade vloženými informáciami do databázy.

2.1.1.6 LARAVEL Auth

Rozšírenie rámca Laravel, Laravel Auth uľahčuje správu a implementáciu overovania užívateľov, registráciu a prihlásenia. V jadre rámcového rozšírenia sú zariadenia zložené z "guards" a "providers". Guards definujú spôsob autentifikácie používateľov pre každú požiadavku. Laravel sa dodáva s ochranou relácie, ktorá pomocou cookie a úložiska relácie udržiava svoj prihlásený stav - session management. Pri používaní webového prehľadávača používateľ uvedie svoje prihlasovacie údaje do formulára. Ak sú tieto údaje správne, aplikácia uloží informácie o autentifikovanom užívateľovi v relácii používateľa. Následne cookie vydaný prehliadačom obsahuje ID relácie, aby následne žiadosti o aplikáciu mohli priradiť užívateľa k správnej relácii. Po prijatí súboru cookie relácie aplikácia načíta údaje relácie na základe ID relácie, zistí, že overené informácie boli uložené v relácii a považuje užívateľa za overeného. Providers získavajú užívateľov z trvalého úložiska, teda z databázy. Laravel má podporu načítania používateľov pomocou nástroja Eloquent. To je základ Laravel Auth a je možné si zdefinovať ďalších providers podľa potreby projektu.

2.1.1.7 Google Maps API

Pre implementáciu Google maps do aplikácie a následnú manipuláciu a úpravu markerov je potrebné založiť si účet na Google Maps API, kde povolíte rozšírenie pre Places API a Roads API aby sa dali zobrazovať adresy a verejné komunikácie a získavať súradnice z vytvorených markerov. Obsah vytvorených markerov sa dá meniť ľubovoľne podľa potreby a aj vzhľad markeru a aj informačného okna. Po vytvorení účtu a implementácie mapy do aplikácie je potrebné do kódu vložiť aj API Key, ktorý bol vygenerovaný na Google účte. Spoločnosť Google pomocou tohto kľúča sleduje frekvenciu užívania mapy a podľa toho môže fakturovať jej používanie

2.1.2 Používateľské rozhranie a funkcionality

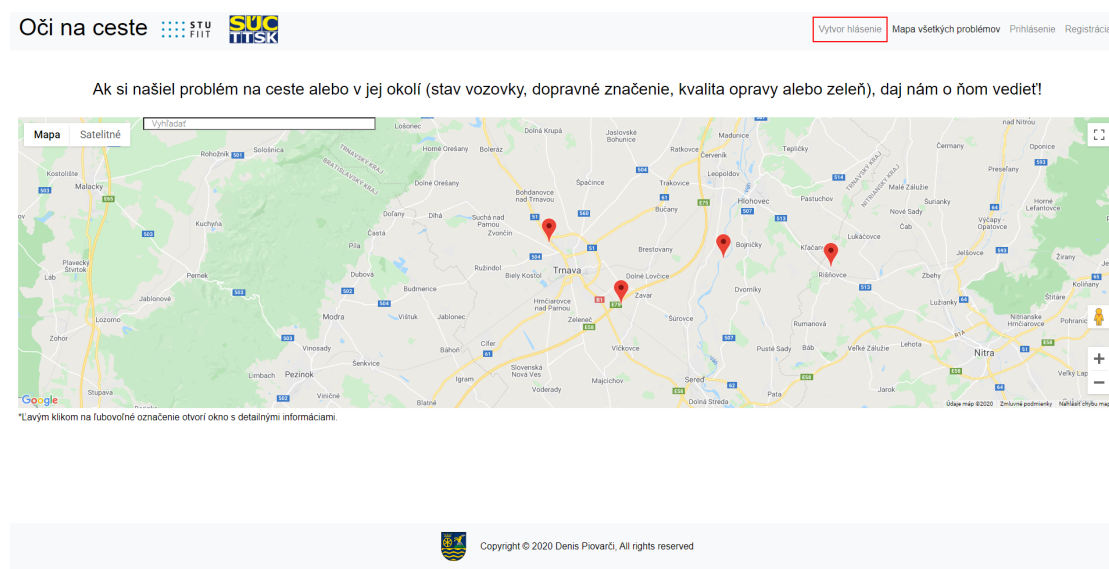
V kapitole Používateľské rozhranie a funkcionality sú detailne opísané všetky používateľské roly a ich oprávnenia na vykonávanie jednotlivých úloh. Detailne sú popísané všetky vykonateľné úlohy a užívateľské rozhranie, na ktorom sú vykonávané.

Oprávnenie/Rola	Nezaregistrovaný občan	Zaregistrovaný občan	Dispečer	Manažér	Admin
Registrácia používateľov	✓	✓	✓	✓	✓
Prihlásenie používateľov		✓	✓	✓	✓
Editácia údajov používateľského konta		✓	✓	✓	✓
Vytváranie označení na mape	✓	✓	✓	✓	✓
Vyhľadávanie nad mapou	✓	✓	✓	✓	✓
Vytvorenie hlásenia problému	✓	✓	✓	✓	✓
Zobrazenie všetkých problémov na mape	✓	✓	✓	✓	✓
Zobrazenie všetkých problémov v tabuľke			✓	✓	✓
Zobrazenie mojich problémov		✓	✓		
Detailný náhľad na problém		✓	✓	✓	✓
Vymazanie problému		✓	✓	✓	✓
Zobrazenie histórie vykonaných zmien			✓	✓	✓
Editácia atribútov problému			✓	✓	✓
Filtrovanie nad problémami			✓	✓	✓
Zmena stavu riešenia problému			✓	✓	✓
Pridanie popisu stavu riešenia problému			✓	✓	✓
Pridelenie problému zamestnancovi				✓	✓
Vymazanie používateľa zo systému					✓
Zobrazenie používateľských účtov					✓
Zmena role používateľa					✓

Obr. 2.1: Tabuľka oprávnení používateľských rolí

2.1.2.1 Domovská stránka

Domovská stránka webovej aplikácie obsahuje mapu so všetkými hláseniami v podobe markerov. Po kliknutí na marker sa zobrazia bežne dostupné informácie pre neregistrovaného používateľa. V menu má neregistrovaný užívateľ na výber prihlásenie, registráciu, mapa všetkých problémov alebo vytvorenie nového hlásenia. Zaregistrovaný užívateľ má na výber ešte Moje hlásenia. Dispečer a Manažér má navyše tabuľku Všetky hlásenia. Admin má v ponuke navyše tabuľku všetkých používateľov.



Obr. 2.2: Domovská stránka webovej platformy

2.1.2.2 Stránka vytvorenia problému

Problém vytvoríme tak, že na mape vyberieme pozíciu, následne sa súradnice presú do formulára. Ďalej vo formulári vyberieme z možností Kategóriu problému a stav problému. V poli Popis problému voľne a stručne popíšeme problém a odošleme. Táto stránka je dostupná aj z profilu Registrovaného používateľa alebo akéhokoľvek zamestnanca a z role neregistrovaného užívateľa.

Oči na ceste STU FIIT SÚC ITSK Vytvor hlásenie Mapa všetkých problémov Prihlásenie Registrácia

Vyhľadaj

Map Satellite

Univerzita sv. Cyrila a Metoda v Trnave Lidl

Katedrálly chrám sv. Jána Krstiteľa Poliklinika Družba DRUŽBA

Bazilika sv. Mikuláša Synagoga café

Mestská veža Bernolákov sad

Kostol sv. Jakuba

City Arena McDonald's

Hotel SPI

Map data ©2020 Terms of Use Report a map error

*Ľavým klikom na mapu sa vytvorí označenie miesta problému.

Vytvorenie hlásenia

Poloha *

48.377878,17.593492

Kategória *

Stav vozovky ▼

Stav problému *

Poškodená ▼

Popis problému *

V smere na City Arénu pri krajnici je hlboký výmol.

Vytvorit

Obr. 2.3: Stránka vytvorenia hlásenia



2.1.2.3 Prihlásenie, registrácia, zmena údajov konta

Registrácia prebieha jednoducho vo formulári, kde zadáme Meno a Priezvisko, platný e-mail, heslo a potvrdíme heslo. Prihlásenie prebieha podobne, kde len vložíme e-mail a heslo.

Zmena používateľských údajov konta prebieha v sekcii Nastavenia po kliknutí na meno prihláseného užívateľa a je možné zmeniť akýkoľvek údaj zadávaný pri registrácii. Nie je overovaná platnosť e-mailu, pretože po registrácii neobdržíte e-mail o potvrdení platnosti e-mailu. Administrátor má následne právo a možnosť meniť používateľské roly u registrovaných užívateľov a tak z nich spraviť Dispečera, Admina alebo manažéra.

2.1.2.4 Tabuľka problémov

Pre verejného registrovaného používateľa sa jedná o zoznam ním vytvorených problémov s atribútmi: Poloha, Vytvorené dňa, Kategória problému, Stav problému a Stav riešenia a pre akéhokoľvek zamestnanca to znamená zoznam úplne všetkých problémov s atribútmi: ID, Poloha, Zadané, Kategória, Stav problému, Stav Riešenia, Priorita, Používateľ, Zamestnanec, Vozidlo. S možnosťou filtrovania nad zvoleným atribútom. Verejný registrovaný používateľ môže zobraziť detail problému alebo problém vymazať kliknutím na ikonu "trash". Všetci zamestnanci môžu editovať aj mazať záznamy, zobraziť detail hlásenia o rozšírené zamestnanecké atribúty: Priorita, Priradený zamestnanec, Priradene vozidlo, Vytvoril. Verejný registrovaný užívateľ môže zobraziť len základné atribúty: ID, Poloha, Popis, Zadané dňa, Kategória, Stav Problému, Stav riešenia problému a Popis stavu riešenia problému. Okrem detailov je pre zamestnancov k dispozícii aj História stavu riešenia problému. Dispečer môže editovať atribúty Poloha, Kategória, Stav Problému, Stav Riešenia Problému, Priradené vozidlo, Popis a Priorita. Manažér a Admin môže navyše editovať Priradeného zamestnanca.

Oči na ceste  

Vytvor hlásenie Všetky hlásenia Mapa Používatelia Šimon Drienik ▾

Všetky hlásenia

[Filtruj](#)

#	ID	Poloha	Zadané Vyber ▾	Kategória Vyber ▾	Stav problému Vyber ▾	Stav riešenia Vyber ▾	Priorita Vyber ▾	Používateľ Vyber ▾	Zamestnanec Vyber ▾	Vozidlo Vyber ▾	Edit	Detail	Vymaž záznam
1	1	48.392184,17.562421	2020-06-02 12:29:36	Stav vozovky	Poškodená	Prijaté	Nepiradená	Dalibor	Nepiradený	Nepiradené			
2	2	48.343830,17.647565	2020-06-02 12:29:49	Kvalita opravy	Zle viditeľná	Prijaté	Nepiradená	Dalibor	Nepiradený	Nepiradené			
3	4	48.379992,17.768147	2020-10-07 10:02:35	Stav vozovky	Chýbajúca	Prijaté	Nepiradená	Šimon Drienik	Nepiradený	Nepiradené			
4	5	48.373104,17.895080	2020-11-06 08:11:37	Stav vozovky	Poškodená	Prijaté	Nepiradená	Nezaregistrovaný občan	Nepiradený	Nepiradené			
5	6	48.855250,17.973334	2020-11-16 10:17:43	Kvalita opravy	Zle viditeľná	Prijaté	Nepiradená	Šimon Drienik	Nepiradený	Nepiradené			
6	7	48.760501,21.270977	2020-11-20 08:10:42	Zeleň	Vyblednutá	Prijaté	Nepiradená	Šimon Drienik	Nepiradený	Nepiradené			

Obr. 2.4: Tabuľka všetkých problémov s filtrovaním

2.1.2.5 Tabuľka všetkých používateľov

Administrátor má možnosť zobrazíť tabuľku všetkých užívateľov po kliknutí v hornej lište na "Používatelia". Tabuľka ponúka možnosť vymazania účtu, podobne ako pri vymazaní problému alebo zmenu používateľskej role kliknutím na "Upraviť rolu". Administrátor má nad množinu scenárov a povolení všetkých typov užívateľov. Tabuľka obsahuje atribúty o užívateľovi: ID, Meno, Email, Rola, Vytvorené dňa.

2.1.3 Databáza

Uchovávanie dát je sprostredkované pomocou MySQL databázy. Hlavným prvkom bude tabuľka problém, ktorá uchováva atribúty ako id problému, polohu vytvoreného problému uloženú ako TEXT a uchováva súradnice oddelené čiarkou, ďalej je to popis problému v hodnote TEXT. Nasledovné atribúty sú ako cudzie kľúče odkazujúce na iné tabuľky.

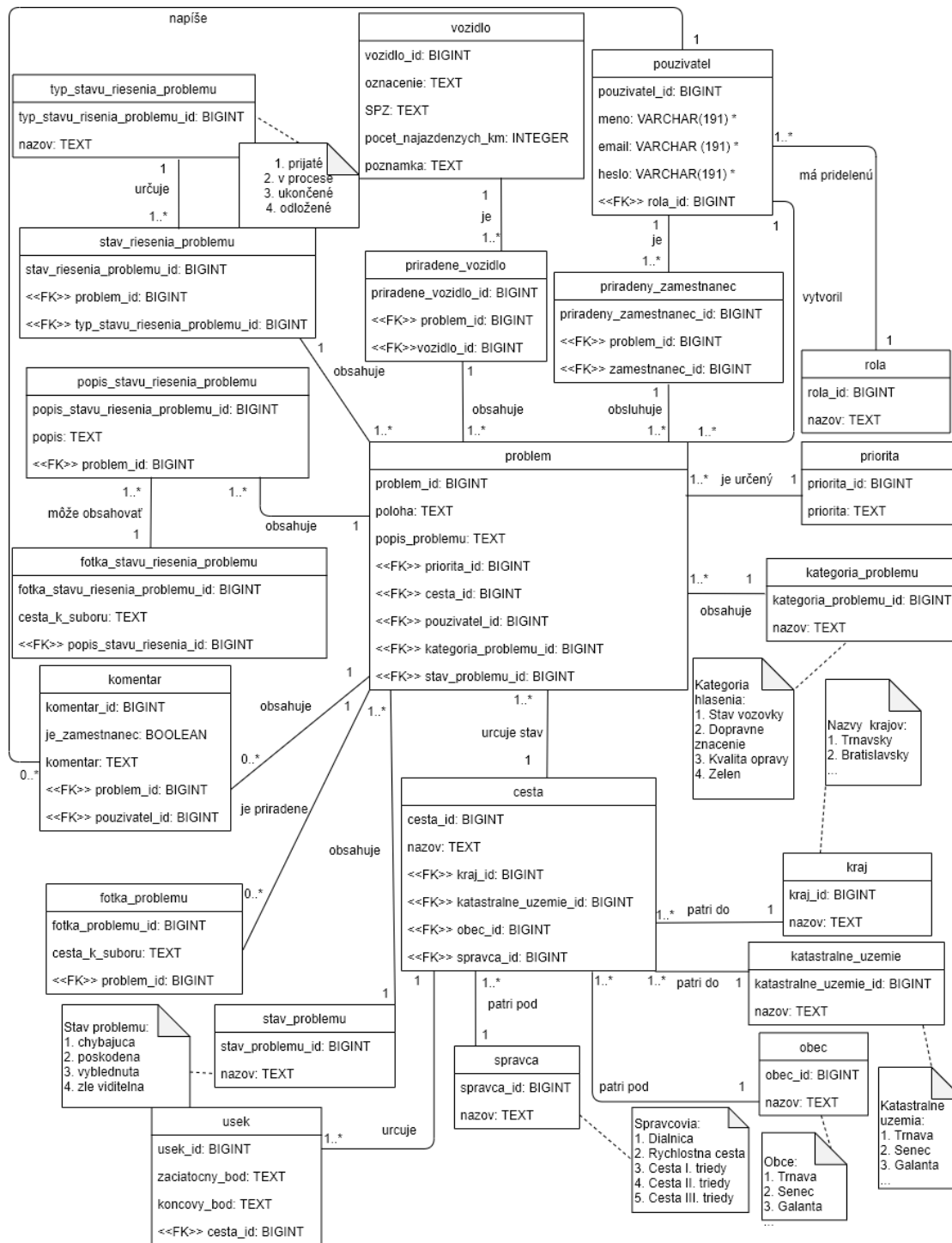
Priorita id je ID záznamu v tabuľke Priorita, ktorá obsahuje atribúty priorita id a priorita s hodnotou text.

Cesta id obsahuje ID záznamu v tabuľke cesta, ktorá obsahuje atribúty ako cesta id, názov v hodnote TEXT, cudzí kľúč kraj id, ktorý odkazuje na číselník kraj. Cudzí kľúč katastrálne územie id, ktorý odkazuje na číselník katastrálne územie. Cudzí kľúč obec id, ktorý odkazuje na číselník obec a cudzí kľúč správca id, ktorý určuje prideleného typu správcu v číselníku správca. Záznam v tabuľke cesta je určený tabuľkou úsek, ktorá obsahuje atribúty úsek id, začiatkový bod ako TEXT, koncový bod ako TEXT a cudzí kľúč cesta id, odkazujúci na záznam v tabuľke cesta.

Používateľ id odkazuje na záznam v tabuľke používateľ a určuje, kto problém vytvoril. V tabuľke používateľ sa nachádzajú atribúty používateľ id, meno v hodnote varchar(191), email v hodnote varchar(191), heslo v hodnote varchar(191), ktoré je ale samozrejme zaťažované pomocou funkcií Bcrypt a Argon2 od balíku Authentication rámca Laravel. Cudzí kľúč rola id, ktorý odkazuje na tabuľku rola a tým určuje akú rolu má pridelenú. Tabuľka rola obsahuje okrem id len názov v hodnote TEXT. tabuľka používateľ je pomocou používateľ id prepojená na tabuľku komentár a určuje kto ho napísal. Tabuľka komentár obsahuje okrem svojho id a id užívateľa BOOLEAN hodnotu je zamestnanec a TEXT komentáru a cudzí kľúč problém id, ktorý odkazuje na tabuľku problém a určuje, ktorý problém obsahuje tento komentár. Tabuľka priradený zamestnanec, okrem svojho id obsahuje cudzí kľúč problém id, ktorý určuje problém na obsluhu a cudzí kľúč zamestnanec id, ktorý odkazuje na záznam v tabuľke používateľ.

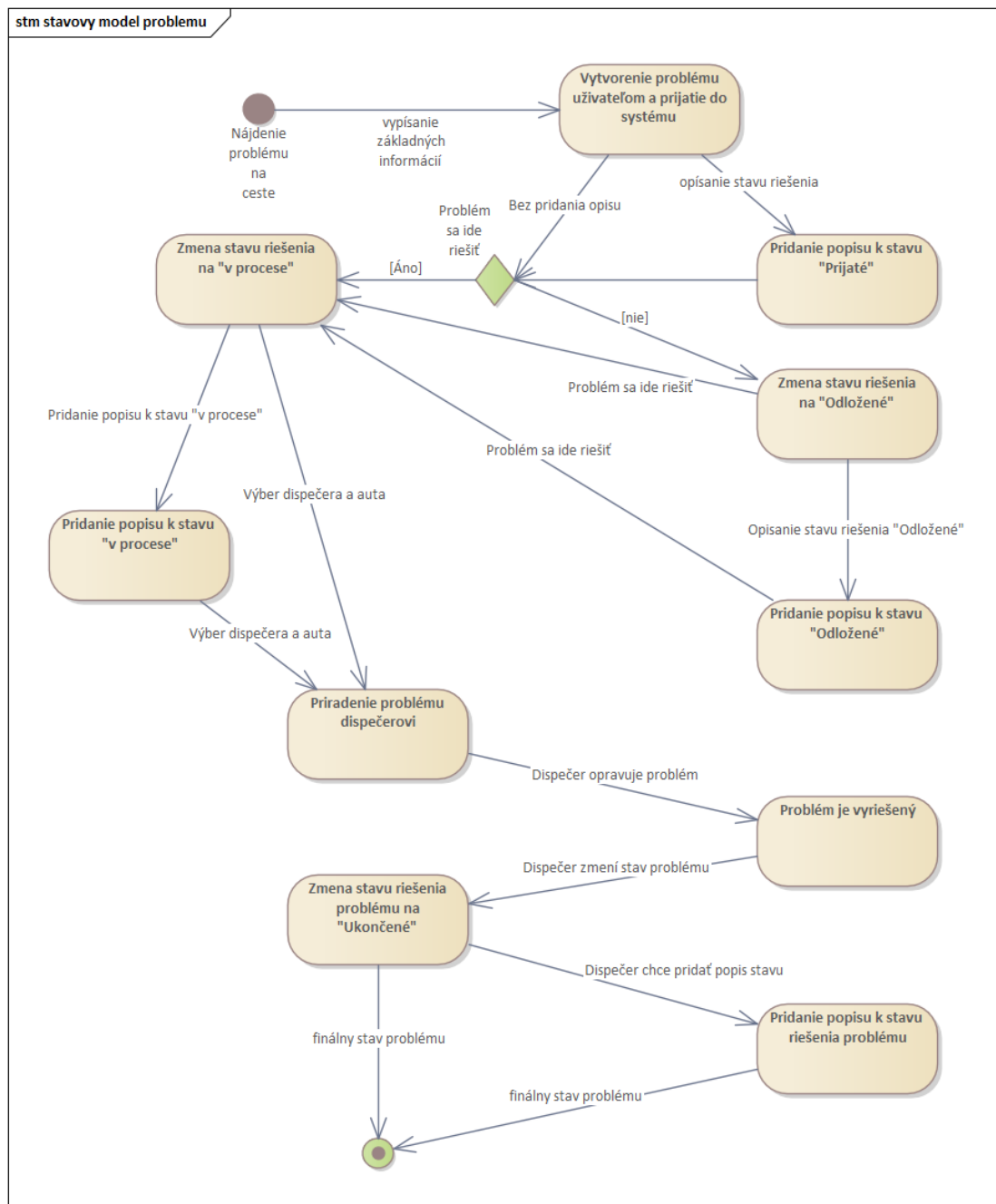
Cudzí kľúč kategória problému id odkazuje na záznam v číselníku kategória problému. Cudzí kľúč Stav problému id odkazuje na záznam v číselníku stav problému.

Nasledovné tabuľky sú prepojené na záznam v tabuľke problém pomocou cudzieho kľúča problém id. Fotka problému, ktorá okrem iného obsahuje cestu k fotke. Popis stavu riešenia problému, obsahujúca popis v hodnote TEXT a na túto tabuľku odkazuje pomocou cudzieho kľúča tabuľka fotka stavu riešenia problému, ktorá okrem id popisu stavu riešenia problému obsahuje cestu k súboru. Ďalej je to stav riešenia problému, ktorá okrem id problému, svojho id, odkazuje pomocou cudzieho kľúča na číselník typ stavu riešenia problému. Nakoniec sa pomocou tabuľky priradené vozidlo určuje, ktoré vozidlo patrí k danému problému, obsahuje ešte cudzí kľúč vozidlo id, ktorá odkazuje na tabuľku vozidlo, kde je označenie v hodnote TEXT, SPZ v hodnote TEXT, poznámka v hodnote TEXT a počet najazdených km v INT.



Obr. 2.5: Schéma vytvorenej databázy

2.1.4 Stavový model problému



Obr. 2.6: Stavový model problému

2.1.5 Analýza zimplementovaného riešenia

V tejto časti sú opísané najdôležitejšie zanalyzované alebo využité časti zdrojového kódu webovej platformy, ktorá je tvorená pomocou frame-worku Laravel. Webová platforma je kompletne naprogramovaná v Laraveli, ktorý ja nainštalovaný na serveri. Pre komunikáciu Android aplikácie a databázy je využitá časť API tohto frame-worku. Do určitých častí boli pridané funkcie, ktoré využíva Android aplikácia.

2.1.5.1 Models

Laravel framework obsahuje triedu Model, ktorá slúži ako komunikátor medzi danou tabuľkou v databáze a aplikáciou. Každá tabuľka má priradený svoj vlastný model. Vďaka týmto modelom môžeme vkladať, načítat a aktualizovať záznamy v tabuľkách. Tieto modely sa nachádzajú v priečinku app/Models. Každý z týchto modelov obsahuje základné informácie o tabuľke, na ktorú odkazuje. To zahŕňa pole "fillable" obsahujúce všetky atribúty tabuľky. "table", ktorej hodnota je názov tabuľky v databáze. "PrimaryKey", premenná, ktorej hodnota je názov stĺpca pre primárny kľúč záznamu. Niektoré modely majú pole "attributes", kde sú zavedené defaultné hodnoty, niektorých atribútov, ktorým sa hneď nepriraduje hodnota. Ak tabuľka obsahuje cudzí kľúč, tak je priradená do modelu funkcia, ktorej návratová hodnota je záznam, ktorého id sa zhoduje s daným cudzím kľúčom. Tento záznam sa vyhľadáva v určenej tabuľke vo funkcií. Pre využitie API pre Android aplikáciu nebolo potrebné zasahovanie alebo doimplementácia do týchto modelov.

2.1.5.2 Controllers

Po tom ako je vyvolaný nejaký Http request pomocou URL linkov sa musí niečo vykonať. Môže to byť interakcia s databázou alebo zobrazenie stránky. To je vykonávané pomocou funkcií zdefinovaných v controlleroch, ktoré sa nachádzajú v

časti app/Http/Controllers. Na všetky interakcie a zobrazenia by nám kludne stačil jeden PHP súbor ale bol by veľmi veľký a neprehľadný. Preto na prácu s každou väčšou entitou je vytvorený samostatný PHP súbor, teda controller, ktorý obsahuje vyššie spomenuté funkcie. Controller slúži ako komunikátor medzi requestom a Modelom. Vykonáva všetky hlavné funkcionality aplikácie.

Pre mobilnú Android verziu tohto projektu sú tieto kontrolléry dôležité a preto bolo potrebné vytvoriť nové, ktoré budú slúžiť na vykonávanie requestov volaných z mobilnej aplikácie.

V controlleri ProblemController.php je doimplementovaná funkcia showAllProblemsAndroid(). Pomocou modelu Problém sú do premennej uložené všetky záznamy. Údaje, ktoré sú získavané pomocou FK sa získavajú pomocou funkcií na interakciu s tabuľkou, na ktorú odkazuje FK v databáze. Je porovnávaná hodnota FK z tabuľky Problém a PK zo sekundárnej tabuľky. Ak sa tieto hodnoty rovnajú, tak z daného záznamu je do premennej uložená požadovaná hodnota na zobrazenie. Pri získavaní hodnoty z FK stav riešenia problému, uložíme do premennej hodnotu FK typ stavu riešenia problému id z tabuľky Stav Riešenia Problému a toto ID ďalej porovnáваме do tabuľky Typ Stav Riešenia Problému, kde pri zhode vyberieme požadovanú hodnotu na zobrazenie. Pre vyhľadanie popisu stavu riešenia problému porovnáваме id problému s atribútom problém id v tabuľke popis stavu riešenia problému. Tieto záznamy vkladáme do listu a vraciame ako json formát.

V controlleri unregisteredAddRecordAndroid slúžiacom na vytváranie záznamov neregistrovaným používateľom, máme ako parametre polohu, popis problému, kategóriu problému a stav problému. Pri kategórii a stave, vkladáme pomocou porovnávania parametrov a názvov v číselníkoch do záznamu iba ID daných názvov z číselníkov, kde sa nám tieto dve hodnoty zhodujú. Hodnota používateľ id je na-

stavená na 1, keďže sa jedná o neregistrovaného užívateľa. Po vytvorení záznamu, získame jeho id a do tabuľky StavRieseniaProblemu vložíme záznam s atribútom problém id na dané id a typ stavu riešenia problému na 1, keďže sa jedná o Prijaté.

Zatiaľ sú doimplementované iba dve funkcie ale pre dokončenie projektu budú potrebné ďalšie a v iných Controller súboroch.

2.1.5.3 Auth Controllers

Auth Controllers sú kontroléry nachádzajúce sa v priečinku Controllers/Auth a majú na starosti Verifikáciu užívateľov, potvrdzovanie hesiel, funkciu zabudnutého hesla, prihlásenie používateľa, registrovanie a resetovanie hesla. Majú na starosti aj veľmi dôležité hešovanie hesiel, aby neboli voľne dostupné v databáze. Zisťujú aj rolu prihlasovaného užívateľa aby vedeli, ktoré časti má prístupné. Jedná sa o rozšírenie Authentication rámca Laravel.

2.1.5.4 Middlewares

Middleware je funkcia, ktorá filtruje prichádzajúce Http požiadavky na API. Nachádzajú sa v podadresári `\app\Http\Middleware`. `Authenticate.php` overuje autentizáciu používateľa. Ak užívateľ nie je autentifikovaný, middleware ho presmeruje na prihlasovaciu obrazovku a ak je autentifikovaný, tak je žiadosti umožnené pokračovať. V `EncryptCookies.php` deklaruujeme výnimky, teda súbory cookie, ktoré nebudú šifrované. `CheckForMaintenanceMode.php` nám definuje URI dostupné počas údržby aplikácie. `CheckRole.php` kontroluje rolu užívateľa, ktorý vyvolal požiadavku na server a v prípade neoprávneného vstupu vypíše kód 401, `RedirectIfAuthenticated.php` presmeruje užívateľa ak bol oprávnený. `TrimStrings.php` súbor atribútov, ktoré nechceme upravovať. V tomto prípade sa jedná o password a password confirmation. `TrustProxies.php` nám deklaruje dôveryhodné proxies pre túto

aplikáciu a header, ktorý proxies deteguje. `VerifyCsrfToken.php` slúži na generovanie CSRF tokenu pre každú aktívnu session používateľa. Zisťuje sa, či je autentifikovaný používateľ osobou, ktorá podáva žiadosti do aplikácie. Tento token je uložený v session používateľa a mení sa pri každej regenerácii. Dajú sa ta aj zdefinovať výnimky žiadostí, ktoré nemusia byť overené tokenom.

2.1.5.5 Routes

Routes alebo cesty, ktoré sú v súbore `web.php` presmerúvajú `Http` requesty, ktoré sú volané pomocou URL. Tieto URL sú zdefinované v daných routes. Ku `Http` requestu je priradená URL, pomocou ktorej má byť vykonaný a ten môže byť vykonaný ak je sublink z requestu bez base URL zdefinovaný v niektorej z ciest v súbore `web.php`. Ešte sa v laraveli nachádza jeden súbor pre routes a to je `api.php` a ten má takú istú funkcionality, akurát pre dosiahnutie ciest v tomto súbore, musí byť v request linku pridaný prefix `/api`. Môžeme implementovať akékoľvek `Http` requesty. V tomto programe sú zatiaľ implementované requesty typu `GET`, `POST`, `RESOURCE`. Presmerovanie funguje tak, že ak sa postfix URL requestu zhoduje s niektorým postfixom v zdefinovanej ceste tak sa vykoná priradená funkcia v danom kontroléri. Funkcia je zdefinovaná takto: "NázovSúboruKontroléru@NázovFunkcievKontroléri", ak URL obsahuje aj údaje tak s týmito údajmi sa dá pracovať vo funkcií tak, že sú vložené ako parameter. Pre android aplikáciu boli vytvorené zatiaľ tieto routes pre vybavovanie `GET` requestov. `/showallAndroid` napojená na kontrolér funkciu `ProblemController@showAllProblemsAndroid` a `/unregisteredPostAndroid/poloha/popisProblemu/kategoriaProblemu/stavProblemu` napojená na funkciu v kontroléri `ProblemController@unregisteredAddRecordAndroid`. Časom ich bude treba ešte veľa pre ďalšie funkcionality aplikácie a requesty.

2.2 Analýza technológií na vývoj android verzie

V tejto kapitole sú opísané niektoré technológie vhodné na vývoj android verzie projektu Oči Na Ceste a predovšetkým tie, ktoré boli vybrané a uznané za najvhodnejšie. API na komunikáciu s databázou na serveri a API na zobrazenie, úpravy a interakcie s mapou Google Maps nie sú v tejto kapitole opísané, na koľko boli už v kapitole Analýza webovej platformy Oči Na Ceste a pri vývoji android aplikácie sú použité tie isté API nástroje, teda Google Maps API a LARAVEL API.

2.2.1 Programovacie prostredie

Na vývoj android aplikácií sa v minulosti využíval najmä Eclipse ako úradník IDE pre Android, ktorý bol štandardom pre prácu v jazyku JAVA. Neskôr sa do popredia dostalo IDE od JetBrains - Java IDE IntelliJ IDEA. Po čase Google oznámil vývoj nového programátorského prostredia Android Studio, ktoré je v podstate základ IntelliJ IDEA od JetBrains rozšírené o doplnky. Vývojom Android Studio sa dostal Eclipse do veľkého ústrania a pre vývoj android aplikácií je už zastaralý a oproti Android Studio veľmi nepopulárny a pracovať v ňom je omnoho zdĺhavejšie a používateľsky neprívetivé oproti Android Studio, ktoré Vám už vygeneruje základnú štruktúru priečinkov a základné súbory pre vývoj aplikácie a tak isto je aj veľmi intuitívne. Navyše Android Studio má širokú škálu používateľov, takže pre prípadné riešenie chýb alebo hľadanie návodov na internete je omnoho ľahšie ako pri už nepopulárnom Eclipse. Toto boli hlavné dôvody, prečo bolo vybrané pre tento projekt Android Studio a nie Eclipse.

2.2.2 Programovací jazyk

2.2.2.1 JAVA

Java, zatiaľ najpopulárnejší jazyk pre vývoj Android aplikácií a tak isto oficiálny vo vývojovom prostredí Android Studio. Je to objektovo orientovaný jazyk, obsahujúci mnoho funkcií vo forme triednych konštruktorov. Java je síce zložitejšia ale pri dodržaní konvencií je aj veľmi ľahko čitateľná. V Android Studio sa pri práci s JAVA nevyužívajú len programátorské časti a triedy ale aj XML manifesty. Tieto manifesty slúžia pre automatické montážne systémy Gradle, Maven. Na tvorbu používateľského rozhrania sa používa XML. Vzhľadom na to, že JAVA je najpopulárnejší a najpoužívanější medzi Android vývojármi, bol vybraný ako prvá voľba pre tento projekt.

2.2.2.2 C++

C++ je jazyk z nižšej úrovne ale podporovaný prostredím Android Studio pomocou Java NDK (Native Development Kit). Výhodou je možnosť písania natívnych aplikácií, ktoré sú užitočné pri tvorbe hier a aplikácií náročnejších na zdroje. To, že je aplikácia napísaná v jazyku C s podporou NDK znamená, že kód nie je spustený v Java Virtual Machine ako pri písaní v JAVA, ale prostredníctvom pripojeného zariadenia. Je to výhodnejšie ako práca v emulátore, pretože pracujete s aplikáciou priamo v zariadení a môžete si odskúšať senzory, gestá a mnoho iného. Avšak táto možnosť je dostupná aj pre JAVA aplikácie. Práca v C++ je náročnejšia ako v JAVA a tak isto je náročnejšia konfigurácia. Odporúča sa preto len na spracovanie a vykresľovanie grafiky, videa, 3D modelov prípadne iných programátorských modulov, potrebných na rýchle vykonanie zložitých operácií. Náročnosť práce v tomto jazyku a nepopulárnosť medzi vývojármi Android aplikácií boli hlavné dôvody, prečo nebol vybraný na prácu v tomto projekte.

2.2.2.3 Kotlin

Kotlin je veľmi mladý programátorský jazyk, vydaný v máji 2017 a už v roku 2019 ho Google uznala za vhodný pre vývoj aplikácií pre Android systém. Postupne sa JAVA tlačí do ústrania a preto nové štandardné nástroje ako knižnice alebo funkcie pre Android Studio budú zamerané predovšetkým pre Kotlin. Na platforme Live Typing píše Android vývojári prevažne v jazyku Kotlin. Začalo sa aj prepisovanie starších JAVA kódov do kódu Kotlin. Android Studio ponúka aj možnosť, že ak je do Kotlin projektu vložený blok kódu napísaný v JAVA, tak Android Studio sa Vás opýta, či chcete tento kód automaticky preložiť do Kotlin. Kotlin je považovaný ako jeden z najlepších moderných programovacích jazykov ako JAVA, Scala, TypeScript. Kotlin má veľmi veľa výhod, ktoré ho robia user-friendly, ako napríklad automatická identifikácia typov údajov, podpora funkčnej paradigmy a rozšírenie funkcií. Do popredia sa dostal vďaka eliminovaniu otravných chýb ako NullPointerException. Riešenie chýb je intuitívne a ľahké vďaka tomu, že pri prípadnej chybe Vám Android nevypíše len kód chybovej hlášky ale rovno vysvetlí prečo tam ten problém je a ako je možné ho odstrániť. Kotlin je postavený na engine JAVA, takže dá sa povedať, že zbral to najlepšie z populárnej JAVA a ostatné vylepšil. Kvôli týmto prevažujúcim kladným vlastnostiam, Kotlin bol vybraný pre prácu na tomto projekte.

2.2.3 Androidové sieťové knižnice a závislosti

Tieto rozšírenia uľahčujú vytvorenie prepojenia medzi android aplikáciou na zariadení užívateľa a API na serveri. Riešia aj odosielanie a prijímanie Http requestov bez zdĺhavého písania kódu. Tieto nástroje sú buď vložené do projektu ako externá knižnica alebo ako závislosť do súboru build.gradle. Gradle je nástroj na automatizáciu zostavovania pre vývoj softvéru. Riadi proces vývoja v úlohách kompilácie

a balenia až po testovanie a publikovanie. Medzi podporované jazyky patrí práve používaný Kotlin.

2.2.3.1 OKHTTP

HTTP je spôsob komunikácie moderných aplikácií v sieti a teda spôsob vymieňania údajov a médií po sieti. OkHttp je klient na vytvorenie HTTP spojenia. Používanie OkHttp je jednoduché. Podporuje synchrónne blokujúce volania a asynchrónne volania so spätnými volaniami. OkHttp je veľmi používaná Http klient knižnica od square.github.io komunity pre Java a Android developerov.

2.2.3.2 ION

ION je tak isto jednoduchá, ale výkonná knižnica Http pre Android, vybavená rozhraním Fluent Chainable API. Je to vhodné na opätovné použitie pripojení, zrušenie príliš dlhých požiadaviek a zobrazovanie obrázkov.

2.2.3.3 Retrofit

Retrofit je najobľúbenejšia Http knižnica. Retrofit je považovaný ako typovo-bezpečný klient Http pre Java a Android vytvorená komunitou Square. Je veľmi vhodná na konzumáciu JSON a XML údajov, ktoré sa parsujú do starých objektov Java (POJO). Pomocou anotácií je veľmi jednoduché vytvárať Http žiadostí. Vďaka obľúbenosti medzi vývojármi a jednoduchému vytváraniu Http žiadostí a odpovedí vo formáte JSON bola táto knižnica zvolená pre prácu na tomto projekte.

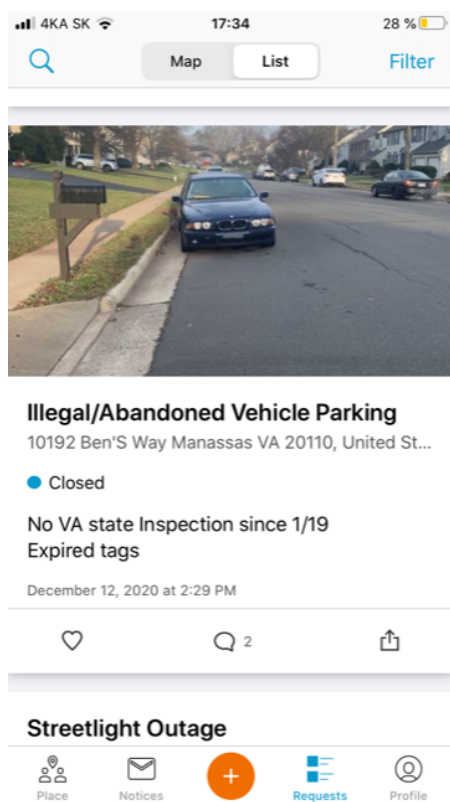
2.3 Analýza existujúcich zahraničných riešení

V tejto sekcií sú vymenované a opísané podobné aplikácie zo zahraničia zamerané na hlásenie poškodenej infraštruktúry.

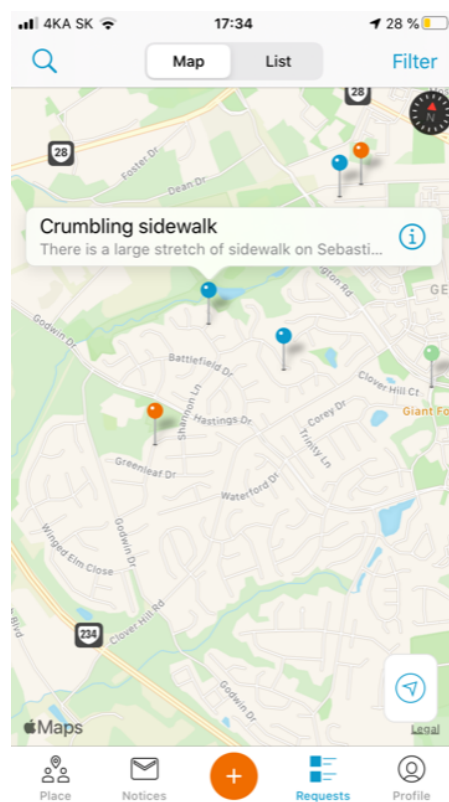
2.3.1 WCRC fix it

Washtenaw County Road Commision fix it je aplikácia z Washtenaw County štátu Michigan v USA, ktorá ponúka možnosť nahlásiť nenúdzové cestné poruchy priamo na úrad Washtenaw County Road Commision. Užívatelia môžu sledovať stav riešenia problému a pri vyriešení problému sú o tom upozornení. Hlásenia sú viditeľné na mape ako markery. Pri vzdialení zoom-u mapky sa markery, ktoré sú pri sebe spoja do jedného a pri ďalšom oddialení sa zase spoja do jedného a pri približovaní sa markery rozbalia do jednotlivých. Markery sú načítavané postupne vždy keď sa zmení pozícia na mape alebo zoom. Markery a teda hlásenia sú dostupné vo verzii mapky alebo zoznamu. K dispozícií je aj vyhľadávací filter podľa statusu, dátumu vytvorenia, dátumu uzatvorenia alebo typu. Je možnosť registrácie a sledovania svojich hlásení. Po kliknutí na marker je k dispozícií okno s názvom a popisom hlásenia a možnosťou rozkliknutia detailov. Po kliknutí na detail uvidíte fotku problému, nadpis, ulicu, stav riešenia, popis a prípadné komentý k hláseniu.

Postup pridávania hlásení: V spodnom menu stlačíme na ikonu "+ä ďalej vyberieme možnosť pridať hlásenie s fotografiou alebo bez fotografie. Po výbere sa zobrazí mapka s červeným krížikom uprostred, ktorý nasmerujeme na danú pozíciu. Ďalej klikneme v pravom hornom rohu na tlačidlo next. Tu vyberieme kategóriu problému. Po ďalšom potvrdení sa zobrazí screen s názvom ulice pridaného hlásenia, možnosť pridania popisu a možnosť pridania kontaktu na Vás pre prípadný dotazu zo strany úradu. Tento kontakt nebude verejne dostupný.



Obr. 2.7: Zoznam hlásení



Obr. 2.8: Mapa hlásení

2.3.2 UDOT Click ‘n Fix

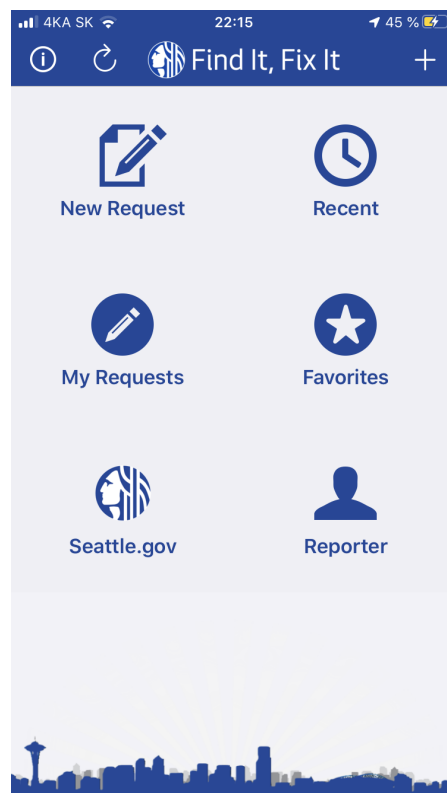
UDOT Click and Fix je aplikácia, ktorá umožňuje nahlásiť problémy a poškodenia na cestách a diaľniciach v štáte Utah v Spojených Štátoch, kde sú tieto problémy nahlasované priamo na Úrad dopravy Utah. Jednoducho položíte marker na mapke, kde ste zaznamenali Závadu. Keď je problém nahlásený, tak je posunutý dispečerom na odstránenie Závady a užívateľovi je odstránenie oznámené. Dizajn a rozhranie je veľmi podobné ba priam identické ako pri aplikácií WCRC Fix It, pretože je od tej istej spoločnosti SeeClickFix. S jedným rozdielom, že problémy nahlásené v tejto aplikácii sú odosielané na úrad štátu Utah a riešené sú problémy len v oblasti štátu Utah ale v aplikácii je vidieť, že sú problémy vytvorené aj za hranicami štátu aj za hranicami USA. Toto by malo byť obmedzené len na oblasť štátu, ktorý tieto problémy rieši aby nebol v markeroch chaos. Výhodou tejto aplikácie aj aplikácie WCRC je, že keď zvolíte nejaké súradnice, tak sa k nim priradí aj názov ulice.

2.3.3 Find It, Fix It

Ak chcete nahlásiť výmoľ, poškodenú cestu, neprehľadnú značku a iné poruchy na cestách v meste Seattle štátu Washington v USA tak je k dispozícii aplikácia Find It, Fix It. V hlavnom menu je k dispozícii možnosť vytvorenia hlásenia a zobrazenie vlastných hlásení aj bez prihlásenia. Avšak nebolo k dispozícii zobrazenie všetkých hlásení. Pri zvolení možnosti Nové hlásenie sa zobrazí zoznam kategórií. Po vybraní kategórie sa zobrazí screen s možnosťou pridania fotky, vybrania lokácie na mapke, popisu, detailov lokácie a ak chcete tak vaše Meno, priezvisko a kontakt pre prípadne odpovedanie o vyriešení problému. Nevýhodou tejto aplikácie je nemožnosť zobrazenia všetkých hlásení na mapke a moc zdĺhavé preklikávanie pri vytváraní hlásenia. Aplikácia pochádza od spoločnosti City of Seattle mobile dev. Rovnako



Obr. 2.9: Mapka pre zameranie markera



Obr. 2.10: Screen hlavného menu

ako v predchádzajúcich aplikáciách je k vybraným súradniciam priradená aj ulica. To je veľkou výhodou pre lepšiu orientáciu dispečerov, pretože namiesto súradníc uvidia názov ulice. Ďalšou nevýhodou je možnosť pridania markera aj za hranice štátu Washington, kde už úrad prepojený s touto aplikáciou nerieši problémy na cestách.

4KA SK 22:15 45 %

< Clogged Storm Drain Submit

Add Photo

For emergencies, call [206-386-1800](tel:206-386-1800)

Location >

Description >

Location Details >

Causing property damage? >

Reporter Anonymous >

Obr. 2.11: Screen vytvorenia hlásenia

2.4 Záver analýzy

V prvej časti analýzy sme sa zamerali na porozumenie a oboznámenie sa s dokumentom a programom webovej platformy Oči Na Ceste. Analyzovali sme použité

technológie a používateľské rozhranie so všetkými užívateľskými roľami. V tejto časti sme zanalyzovali aj databázu a podrobne opísali všetky tabuľky. Hlavnou položkou projektu je "problémä preto sme zostrojili jeho stavový model. Na záver tejto časti sme zanalyzovali jednotlivé časti kódu a opísali naše rozšírenia. V druhej časti sme sa venovali technológiám na vývoj Android verzie projektu. Opísali sme vybrané programovacie prostredie. Dostupné programovacie jazyky a sieťové knižnice dostupné na zrealizovanie projektu a odôvodnenie výberu daného jazyka a knižnice. V tretej časti sme hľadali a analyzovali zahraničné verzie nášho projektu. Podarilo sa nám ich nájsť veľa z USA ale veľa z nich boli od spoločnosti SeeClickFix a tak boli podobné ale podarilo sa nájsť aj aplikáciu zo štátu Washington, ktorá je odlišná a od odlišnej spoločnosti. V ďalšej kapitole sa dočítate o špecifikáciách a návrhu projektu

Kapitola 3

Návrh riešenia

Kapitola 4

Implementácia

Kapitola 5

Testovanie

Kapitola 6

Záver

Práca na projekte v zimnom semestri bola rozdelená na časti. Ako prvé sme sa oboznámili s webovou platformou Oči Na Ceste pre najlepšie pochopenie a zistenie, na čo je projekt zameraný, aké sú tam používateľské role a prípady použitia. V projekte je niekoľko rolí a zistiť čo jednotlivé roly majú sprístupnené. Nám pomohlo vytvoriť si účet v každej jednej z nich. To sme urobili až ku koncu semestra. Neskôr sme získali prístup na server, kde sa nachádza API, databáza a celá webová platforma aby sme mohli pridať vlastné funkcionality a zanalyzovať zdrojový kód. V ďalšej časti sme sa venovali analýze dostupných nástrojov na vybudovanie požadovanej aplikácie a nastavili sme plán práce pre zimný semester. Zistili sme, že použijeme API, ktorá sa už nachádza na serveri a rozšírime ju o potrebné funkcionality. Pre zobrazovanie API Google Maps sme museli vytvoriť vlastný účet na Google Maps. V prevej polovici semestra sme implementovali mapu a knižnicu na komunikáciu s API na serveri a vykonali kompletnú zálohu servera. V druhej polovici semestra sa podarilo zobrazovať hlásenia na mapke v mobile a začali sme s písaním analýzy. Ku koncu semestra sa podarilo implementovať funkcionality na pridávanie hlásení z mobilu do db. Po vyriešení zobrazovania a pridávania hlásení

neregistrovaným používateľom, bola práca zameraná len na písanie dokumentu až do konca semestra.

6.1 Zhodnotenie projektu

Plán práce na projekte v zimnom semestri nebol úplne dodržaný ale základné funkcionality a analýza sa podarilo dodržať. Počas implementácie sa vyskytli problémy ako problém s prihlásením na server. Pri pridávaní funkcionalít sme týždeň riešili ich nefunkčnosť a nakoniec sme zistili, že boli implementované v zlom adresári. Funkcionality boli písané v textovom editore Vim. Počas toho sa naskytli problémy ako hlášky 500 error, ktoré sa zobrazovali na webe namiesto stránok. Preto sme povolili zobrazovanie error výpisov namiesto len hlášok a následne sme zistili, že pri pridávaní vlastných funkcionalít do API, boli nedopatrením znehodnotené funkcionality webovej platformy. Tento problém bol vyriešený prepísaním funkcionalít zo zálohy servera. V druhom semestri bude práca pokračovať na funkcionalitách pridelených jednotlivým rolám, dopísavanie dokumentu, vylepšovanie používateľského rozhrania, registrácie a prihlasovanie a ostatné potrebné záležitosti.

6.2 Možné vylepšenia

V doposiaľ vykonanej práci sú dostupné niekoľké vylepšenia. Upraviť používateľské rozhranie pre lepšie používanie a pre lepšie pridávanie hlásení. To môže byť vykonané testovaním nezainteresovanou osobou a následné úpravy v dizajne a v prípadoch použitia. Kategórie pri vytváraní hlásenia sú na tvrdo pridané v zdrojovom kóde namiesto toho aby boli dynamicky ťahané priamo z tabuľky v db. Bolo by to efektívnejšie, pretože keď sa pridá ďalšia funkcionalita v db tak bude

automaticky dostupná aj v mobile.

Bibliografia

Dodatok A

Harmonogram práce

A.1 Zimný semester

1 st -4 th	
5 th -6 th	
7 th	
8 th -10 th	
11 th -12 th	

A.2 Letný semester

1 st -2 nd	
3 rd -6 th	
7 th -9 th	
10 th	
11 th -12 th	

Dodatok B

Digitálne médium

Evidenčné číslo práce v informačnom systéme: FIIT-XXXX-XXXXX

Link digitálnej časti: (GitHub repozitár):

https://github.com/SimonDrienik/Digitalizacia_spravy_ciest