

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Mobilní aplikace pro včasné varování řidičů při průjezdu
vozidel záchranných složek



2022

Vedoucí práce:
Mgr. Jiří Balun

Jiří Šesták

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Jiří Šesták
Název práce: Mobilní aplikace pro včasné varování řidičů při průjezdu vozidel záchranných složek
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2022
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Jiří Balun
Počet stran: 44
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Jiří Šesták
Title: Mobile application for a swift warning of drivers when an ambulance or a fire truck is about to pass by
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2022
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Jiří Balun
Page count: 44
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Závěrečná práce se zabývá včasným varováním řidičů v provozu, jehož cílem je umožnit rychlou reakci řidičů a vytvoření prostoru na silnicích pro vozidla záchranných a bezpečnostních složek na výjezdu. Základem systému je mobilní aplikace klienta (řidiče) a webový server záchranných složek. Aplikace komunikuje s webovým serverem, který reaguje na polohu zařízení řidiče a polohy záchranných vozidel na výjezdu.

Synopsis

The thesis focuses on a swift warning of drivers to speed up their reaction and to help them clear the road faster for emergency vehicles. Thesis stands on a client's (driver's) mobile application and emergency services' web server. Client's application communicates with the web server, which reacts to the device location of the client and locations of any emergency vehicles in action.

Klíčová slova: varování řidičů; záchranné složky; mobilní aplikace; webový server; poloha vozidel

Keywords: warning of drivers; emergency services; mobile application; web server; vehicle's location

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
2	Systém používaný Zdravotním operačním střediskem Olomouckého kraje	9
2.1	Národní informační systém	9
2.2	Mapa	9
2.3	Probíhající události	9
2.4	Obecné okno s detaily událostí a podrobnostmi o posádkách sanitních vozů	11
2.4.1	Událost	11
2.4.2	Naléhavost	12
2.4.3	Indikace	12
2.4.4	Popis posádky sanitního vozu	13
2.4.5	Příznaky k události:	13
2.4.6	Horská služba	13
2.4.7	Mobilní aplikace Záchranka	13
3	Uživatelská dokumentace	14
3.1	Mobilní aplikace	14
3.1.1	Instalace	14
3.1.2	Notifikace	15
3.2	Webový server	15
3.2.1	Mapa	18
3.2.2	Filtry	18
3.2.3	Tlačítko refresh	18
3.2.4	Okno simulace	19
3.2.5	Zadávací formulář pro událost	19
3.2.6	Zadávací formulář připojení záchranného vozidla	19
3.2.7	Uzavírací formuláře pro událost a záchranné vozidlo	19
3.2.8	Tabulka aktivních událostí	19
3.2.9	Tabulka záchranných vozidel	20
4	Programátorská dokumentace	22
4.1	Mobilní aplikace	22
4.1.1	ClientApp	22
4.1.1.1	MainPage.xaml	22
4.1.1.2	MainPage.xaml.cs	22
4.1.1.3	StartSending	23
4.2	Databáze	23
4.2.1	RescueCarsTable	23
4.2.2	CarsTable	24
4.2.3	EventsTable	24
4.3	Webový server	24

4.3.1	Models	25
4.3.2	Controllers	26
4.3.3	Views	27
4.3.4	Fungování webového serveru	27
5	Testování	30
5.1	Varování	30
5.2	Upozornění	30
5.3	Rozhodování	31
5.4	Tesovací simulace na serveru	31
5.4.1	Hlavní testovací scénáře	31
5.4.2	První testovací scénář	31
5.4.2.1	Auto 1 v simulačním scénáři 1	32
5.4.2.2	Auto 2 v simulačním scénáři 1	32
5.4.2.3	Auto 3 v simulačním scénáři 1	32
5.4.3	Druhý testovací scénář	32
5.4.3.1	Auto 1 v simulačním scénáři 2	35
5.4.3.2	Auto 3 v simulačním scénáři 2	35
5.4.3.3	Auto 5 v simulačním scénáři 2	35
5.4.4	Třetí testovací scénář	35
5.4.4.1	Auto 1 v simulačním scénáři 3	37
5.4.4.2	Auto 2 v simulačním scénáři 3	37
5.4.4.3	Auto 3 v simulačním scénáři 3	37
5.4.5	Auto 4	37
6	Google Maps API	39
6.1	Maps Javascript API	39
6.2	The Directions API	39
6.3	Geocoding API	39
6.4	Problémy využití Google Maps API	39
	Závěr	41
	Conclusions	42
	A Obsah přiloženého CD/DVD	43
	Literatura	44

Seznam obrázků

1	Mapa operátora v systému Dispečer	10
2	Okno probíhajících událostí v systému Dispečer	10
3	Okno s detaily událostí v systému Dispečer	11
4	Číselník pro rychlé vytáčení kontaktů dispečera	12
5	Výchozí obrazovka v neaktivním/aktivním režimu	14
6	Varování a upozornění	16
7	Vyskakovací okno	17
8	Mapa	20
9	Okno simulace, zadávací a uzavírací formuláře	21
10	Tabulka pro aktivní události a tabulka pro sanitky	21
11	ADO.NET entity data model	25
12	Trasy klientských aut	32
13	Začátek simulačního scénáře 1	33
14	Scénář 1 - Auto 3 dostává varování	33
15	Scénář 1 - Auto 3 je z dosahu sanitky	33
16	Scénář 1 - Auto 1 dostává upozornění	34
17	Scénář 1 - Auto 2 dostává upozornění	34
18	Scénář 1 - Auto 2 dostává varování	34
19	Scénář 1 - Auto 2 je z dosahu sanitky	35
20	Začátek simulačního scénáře 2	35
21	Scénář 2 - Auto 5 dostává varování	36
22	Scénář 2 - Auto 1 dostává upozornění	36
23	Scénář 2 - Auto 3 dostává upozornění	36
24	Začátek simulačního scénáře 3	37
25	Scénář 3 - Auto 2 dostává varování	38
26	Scénář 3 - Auto 2 a 3 jsou z dosahu sanitky	38
27	Scénář 2 - Auto 1 a 3 dostávají upozornění	38

Seznam zdrojových kódů

1	Model pro záchranné vozidlo	25
2	Kontroler MobileAppController	40

1 Úvod

Doba příjezdu vozidel záchranných a bezpečnostních složek je často rozhodujícím faktorem při záchraně lidských životů. Na silnicích lze často vidět, jak se doba příjezdu vozidel záchranných složek prodlužuje kvůli pomalým reakcím jednotlivých účastníků silničního provozu. V dnešní době je reakce ostatních účastníků provozu závislá na zvuku sirény záchranného vozidla. Bohužel záleží na množství faktorů jako je hlasitost rádia v automobilu nebo okolní ruch provozu, jestli řidič sirénu uslyší včas a s předstihem vytvoří záchrannou uličku pro vozidlo na výjezdu. V ideálním světě, kde by vozidlo záchranných složek mělo včas připravený volný průjezd po celé délce trasy k nehodě nebo pacientovi, by se doba příjezdu velmi zkrátila, a tím by se zvýšil počet zachráněných lidských životů. Proto se nabízí otázka, jestli neexistuje jiné efektivnější řešení. Standardním vybavením téměř každého člověka je v moderní době chytrý telefon, který se často používá i při jízdě automobilem například kvůli navigaci. Použití mobilních telefonů pro detekci poloh automobilů a jejich kolizí s trasami a polohami vozidel záchranných složek, je jedno z řešení, které by mohlo při dostatečném počtu uživatelů zkrátit dobu příjezdu vozidel na výjezdu.

V budoucnu by se podobná funkcionalita dala využít v autonomních vozidlech nebo chytrém řízení provozu pomocí semaforů. Komunikací mezi jednotlivými účastníky silničního provozu se zabývá projekt C-Roads, jehož cílem je propojení silničního provozu do komunikující sítě tak, aby se snížil počet dopravních nehod a zefektivnil přesun vozidel záchranných a bezpečnostních složek. Mým hlavním cílem bylo vymyslet systém, který bude založený na komunikaci mobilní aplikace řidiče (klienta) se serverem záchranných složek s využitím *Google Maps API*, jehož cílem bude notifikovat řidiče o blížících se vozidlech na výjezdu tak, aby vytvořili záchrannou uličku s dostatečným předstihem a vozidla na výjezdu měla volný průjezd.

2 Systém používaný Zdravotním operačním střediskem Olomouckého kraje

V rámci sběru dat pro bakalářskou práci, jsem navštívil Zdravotní operační středisko Olomouckého kraje. V dnešní době operační střediska po celé České republice používají systém *Dispečer* od firmy *RCS Kladno*. Operátor v systému vidí probíhající události, mapu s aktuálním pohybem vozů všech složek Integrovaného záchranného systému České republiky spolu s aktivními událostmi a seznam událostí s detailem vybrané události. Dále je systém propojený se systémy ostatních složek integrovaného záchranného systému pomocí Národního informačního systému.

2.1 Národní informační systém

Národní informační systém propojuje systémy všech integrovaných záchranných systémů a umožňuje komunikaci zdravotníků, policistů a hasičů v rámci jednoho vlákna pro danou událost. Jednotlivé složky zde komunikují a vytváří stavy události podle vlastních aktivit vázaných k události. Jedná se o zaznamenávání výjezdů a příjezdů k události, stavu pacienta, celkové situace na místě a různé další podrobnosti.

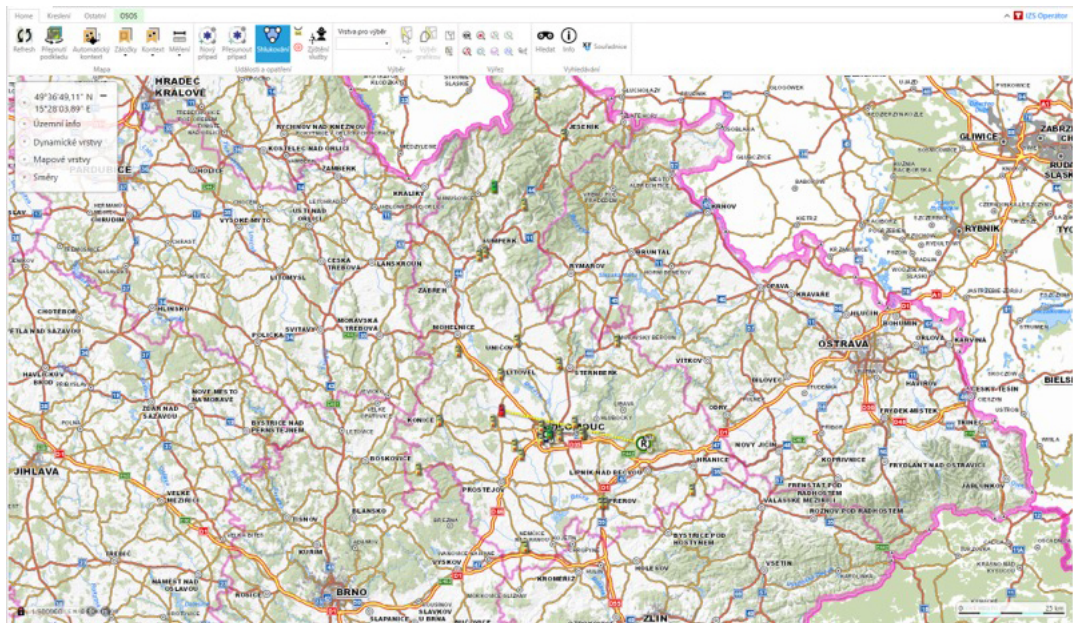
2.2 Mapa

Mapa systému má několik vrstev, které může operátor použít pro lokalizaci pacienta. Základem je mapa zobrazující názvy ulic a čísla popisné. Operátor má k dispozici také turistickou vrstvu, která zobrazuje všechny turistické trasy a cyklotrasy. Dále má k dispozici satelitní mapu, kterou používá pro upřesnění okolí události a popřípadě k určení ideálního místa pro přistání záchranného vrtulníku.

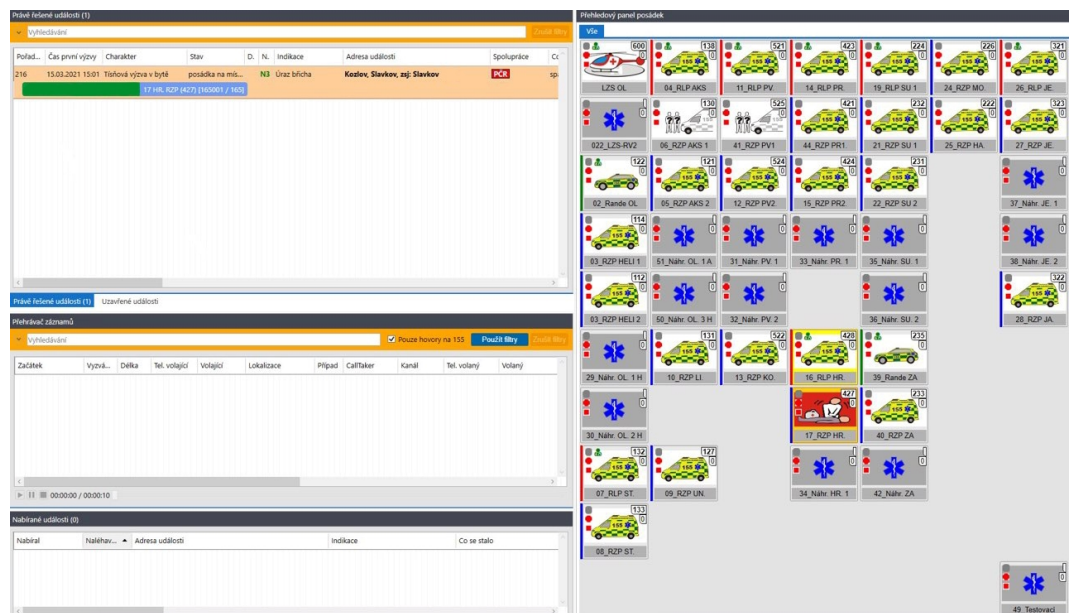
Operátor také vidí přibližnou polohu volajícího s přesností na 2 km. Pokud má volající chytrý telefon s funkcí AML, která posílá přesné souřadnice operátorovi, je zaměření pacienta s přesností desítek metrů.

2.3 Probíhající události

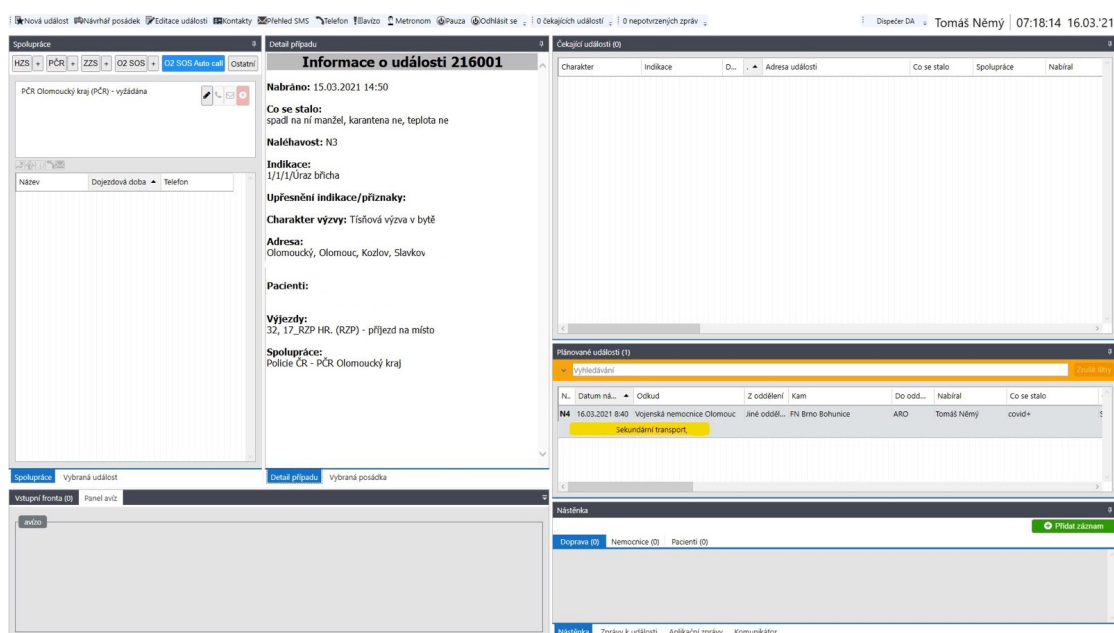
V okně probíhajících událostí vidí operátor seznam událostí a jaká vozidla jsou s událostí spojena. Kromě seznamu událostí má k dispozici seznam všech vozidel integrovaného záchranného systému, které může přetažením napojit na probíhající událost.



Obrázek 1: Mapa operátora v systému Dispečer



Obrázek 2: Okno probíhajících událostí v systému Dispečer



Obrázek 3: Okno s detaily událostí v systému Dispečer

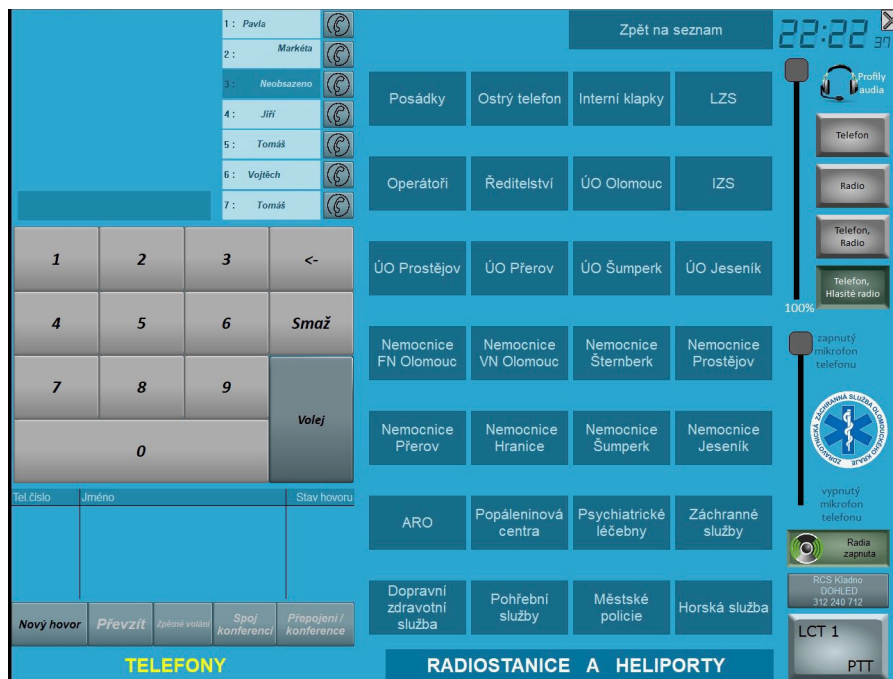
2.4 Obecné okno s detaily událostí a podrobnostmi o posádkách sanitních vozů

Obecné okno obsahuje popis vybrané události a vybrané posádky vozidla. Dále také obsahuje okno s vlákem události Národního informačního systému a chat pro komunikaci.

2.4.1 Událost

Událost obsahuje:

- Id události
- Stav události
- Jméno pracovníka, který událost založil v systému
- Naléhavost
- Indikace
- Popis aktuální situace na místě události
- Oznamovatele
- Telefon oznamovatele
- Adresu události



Obrázek 4: Číselník pro rychlé vytáčení kontaktů dispečera

- Jméno a věk pacienta
- Spolupráci s jinými složkami
- Informaci o umístění pacienta v rámci pater domu kvůli výbavě posádky

2.4.2 Naléhavost

Naléhavost se dělí na 4 typy:

1. N1 - Došlo k selhání životních funkcí, nebo pacient velmi krvácí (Pokud je událost vedena s naléhavostí N1, musí jet k události v rámci posádky doktor).
2. N2 - Pravděpodobně dojde k selhání životních funkcí.
3. N3 - Může dojít ke zhoršení stavu pacienta.
4. N4 - Ostatní

2.4.3 Indikace

Indikace obsahuje zkrácený heslovitý popis události (bezvědomí, nedýchá, zlomenina horní končetiny, urgent...)

2.4.4 Popis posádky sanitního vozu

Popis obsahuje typ auta, které jede k dané události (značka a model auta) a posádku. Ne vždy je v posádce doktor. Ten vyjíždí z důvodu nedostatku doktorů jen na vážné případy.

2.4.5 Příznaky k události:

Každá událost obsahuje příznaky, které slouží ke statistickým účelům.

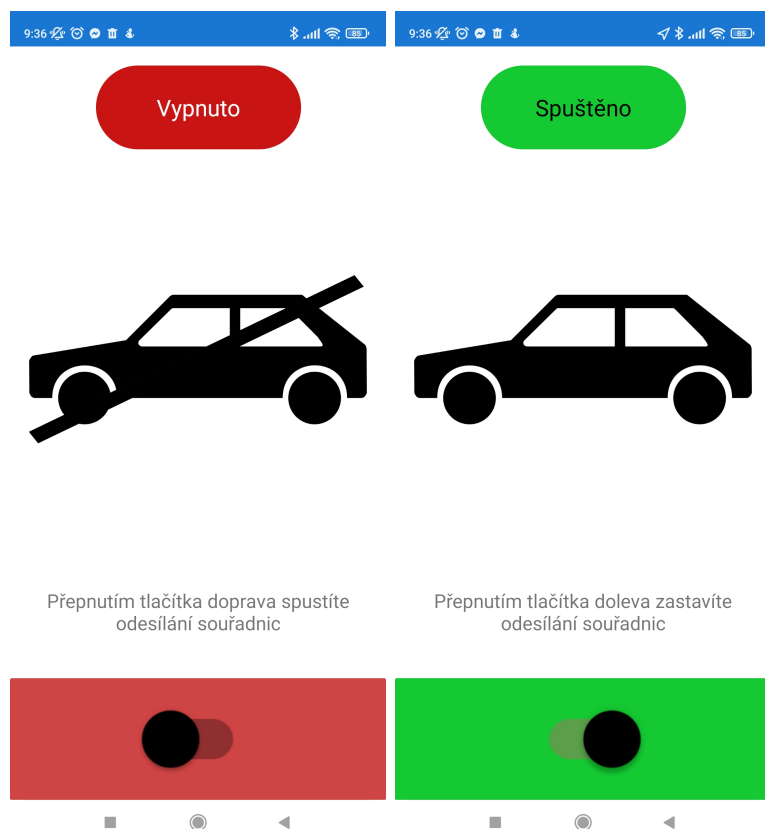
- Telefonická asistovaná neodkladná resuscitace
- Telefonická neodkladná první pomoc
- Pacient pozitivní na Covid
- Porod mimo nemocnici
- Pacient s ukrajinským občanstvím
- Byl použit automatický externí defibrilátor
- Odeslání SMS zprávy o události vedení záchranné služby

2.4.6 Horská služba

Horská služba není zatím propojena s ostatními složkami integrovaného záchranného systému, ale má svůj systém *Horská služba* od firmy *Gina*.

2.4.7 Mobilní aplikace Záchranka

Operační středisko pracuje i s oznámeními, které přichází z aplikace *Záchranka*. Pokud uživatel aplikace *Záchranka* stiskne nouzové tlačítko, operátorovi se rozezní tón upozornění a na odděleném monitoru se mu zobrazí místo pacienta spolu s jeho jménem a příjmením, věkem, stavem signálu a baterie telefonu a další podrobnosti.



Obrázek 5: Výchozí obrazovka v neaktivním/aktivním režimu

3 Uživatelská dokumentace

3.1 Mobilní aplikace

Mobilní aplikace je určena pro uživatele (řidiče). Při spuštění se aplikace nachází v neaktivním režimu (vypnuto), ve kterém čeká na přepnutí do aktivního režimu (zapnuto) uživatelem, ve kterém zasílá polohu zařízení webovému serveru a přijímá jeho odezvu. Pro správný běh aplikace je potřeba mít své zařízení připojené k Internetu (zapnutá mobilní data) a mít zapnuté a povolené sdílení polohy zařízení. Pokud není nějaká z těchto podmínek splněna, aplikace se přepne do neaktivního režimu. Aplikace je zaměřena na operační systém Android. Minimální podporovaná verze systému je 5.0 (Lollipop).

Výchozí obrazovka je rozdělena na ukazatel režimu, ikonu režimu, popis a tlačítko přepnutí do/z aktivního/neaktivního režimu.

3.1.1 Instalace

Při stažení instalačního souboru se může uživateli zobrazit upozornění, které říká, že z bezpečnostních důvodů nelze aplikaci z neznámých zdrojů instalovat. V tomto případě je nutné v *Nastavení* povolit instalaci z neznámých zdrojů.

Po spuštění instalačního souboru se objeví vyskakovací okno *Blokováno funkcí Play Protect*. Uživatel zde musí dát možnost *Přesto nainstalovat*. Po instalaci se může objevit vyskakovací okno *Odeslat aplikaci ke kontrole* u kterého musí vybrat uživatel možnost *Neposílat*. Poté je aplikace připravena k používání.

3.1.2 Notifikace

Mobilní aplikace přijímá 3 typy notifikací:

1. Varování
2. Upozornění
3. Vyskakovací okna

Varování je notifikace s vyšší prioritou a znamená, že zařízení je méně než 200 metrů od aktivního záchranného vozidla na výjezdu a zároveň je přímo na jeho trase. Reakcí řidiče na příjem varování by mělo být okamžité vytvoření záchranářské uličky podle pravidel silničního provozu. Po přijetí varování jej lze potvrdit tlačítkem „rozumím“. Pokud to uživatel neudělá, je varování uzavřeno automaticky po 30 vteřinách.

Upozornění je notifikace s nižší prioritou a znamená, že zařízení je méně než 200 metrů od aktivního záchranářského vozidla, ale není přímo na jeho trase. Při přijetí upozornění by měl řidič zvýšit svoji pozornost, aby mohl reagovat na změnu aktuální situace, pokud by záchranářské vozidlo změnilo svoji trasu. Stejně jako u varování lze upozornění zavřít pomocí tlačítka „rozumím“, nebo jej zavře aplikace automaticky po 30 vteřinách.

Vyskakovací okna jsou notifikace týkající se chodu aplikace, která jsou reakcí na neaktivní přístup k Internetu a vypnuté sdílení polohy zařízení.

3.2 Webový server

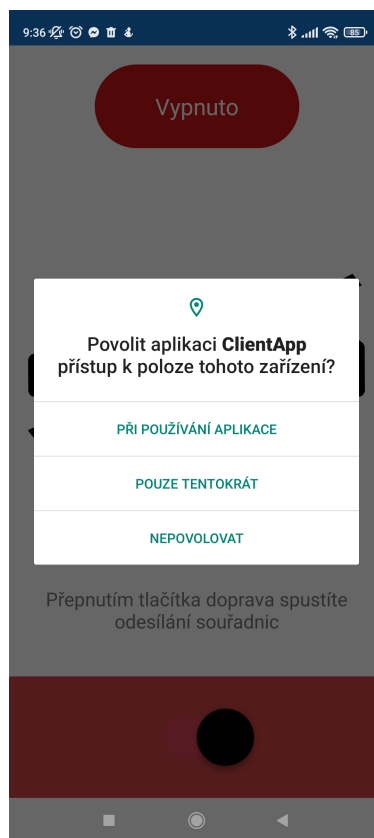
Webový server je určen k simulaci zjednodušeného systému, se kterým pracuje operátor ve zdravotním operačním středisku.

Hlavní stránka webového serveru obsahuje:

1. Mapu
2. Filtry
3. Zadávací formulář pro událost
4. Zadávací formulář pro záchranné vozidlo
5. Tabulku aktivních událostí
6. Tabulku záchranných vozidel v systému








Obrázek 6: Varování a upozornění



Obrázek 7: Vyskakovací okno

3.2.1 Mapa

Mapa zobrazuje záchranná vozidla, aktivní události a pro potřeby testování také polohy řidičů s mobilní aplikací v aktivním režimu. Podržením klávesy Ctrl a točením kolečka myši, lze mapu přiblížit/oddálit. Na mapě se zobrazují ikony a vypočtená trasa záchranného vozidla na výjezdu. Po přejetí myši na sanitní vůz nebo ikonu události na mapě se zobrazí informace o dané sanitce nebo události.

1. Ikona záchranného vozidla 
2. Ikona události 
3. Trasa záchranného vozidla je vykreslena na mapě modrou čarou.
4. Ikona vozidla s mobilní aplikací 
5. Vozidlo po přijetí varování 
6. Vozidlo po přijetí upozornění 

3.2.2 Filtry

Filtry slouží operátorovi k filtrování sanitek. Pro potřeby testování je přidáno tlačítko pro spuštění/zastavení obnovování mapy.

Typy filtrů:

1. Podle *Id* záchranného vozidla – zobrazí pouze vozidlo s daným *Id*.
2. Podle stavu záchranného vozidla – zobrazí pouze aktivní/ neaktivní vozidla.
3. Podle stavu výjezdu záchranného vozidla – zobrazí vozidla, která jedou k pacientovi nebo se vrací do nemocnice.
4. Podle *Id* aktivní události – zobrazí záchranná vozidla připojená k dané události.

3.2.3 Tlačítko refresh

Tlačítko refresh simuluje spuštění serveru, kdy server začne porovnávat polohy sanitek a uživatelských aut, vykreslovat ikony na mapu a rozesílat upozornění a varování. Pokud je tlačítko červené, znamená to, že server je zastavený. Pokud je tlačítko zelené, server je spuštěn. Po každém zastavení se server vrací do konzistentního stavu kvůli testování a simulacím. Konzistentní stav obsahuje 3 neaktivní sanitky pro testování a 3 události pro testování. Všechny ostatní sanitky, události a uživatelská vozidla jsou odstraněny.

3.2.4 Okno simulace

Okno simulace obsahuje výběr ze tří testovacích sanitek a pěti testovacích uživatelských vozidel. Kvůli omezení pro použití neplacené verze Google Maps API, je simulace sanitek omezena na jednu sanitku v dané simulaci. Jakmile je aktivních sanitek více, může dojít k překročení limitu dotazů za sekundu a k nekorrektnímu chování serveru. Omezení se netýká uživatelských aut, takže je možné si vybrat jakoukoliv kombinaci aut pro simulaci.

3.2.5 Zadávací formulář pro událost

Zadávací formulář pro událost simuluje jak reálně funguje aktuálně používaný systém v operačních střediscích. Operátor přijme hovor, sepiše informace do události a založí ji.

Operátor zadává:

1. Adresu (město, ulice, číslo popisné, nebo zeměpisné souřadnice zem. délka a zem. šířka)
2. Naléhavost (viz 2.4.2)
3. Popis události

Popis události je předvyplněný informacemi, které by měl operátor doplnit.

3.2.6 Zadávací formulář připojení záchranného vozidla

Po vzniku události připojí operátor k dané události záchranné vozidlo, které v reálném provozu vyjíždí k události.

Operátor zadává:

1. *Id* události
2. Popis vozidla

Popis vozidla je předvyplněný informacemi, které by měl operátor doplnit.

3.2.7 Uzavírací formuláře pro událost a záchranné vozidlo

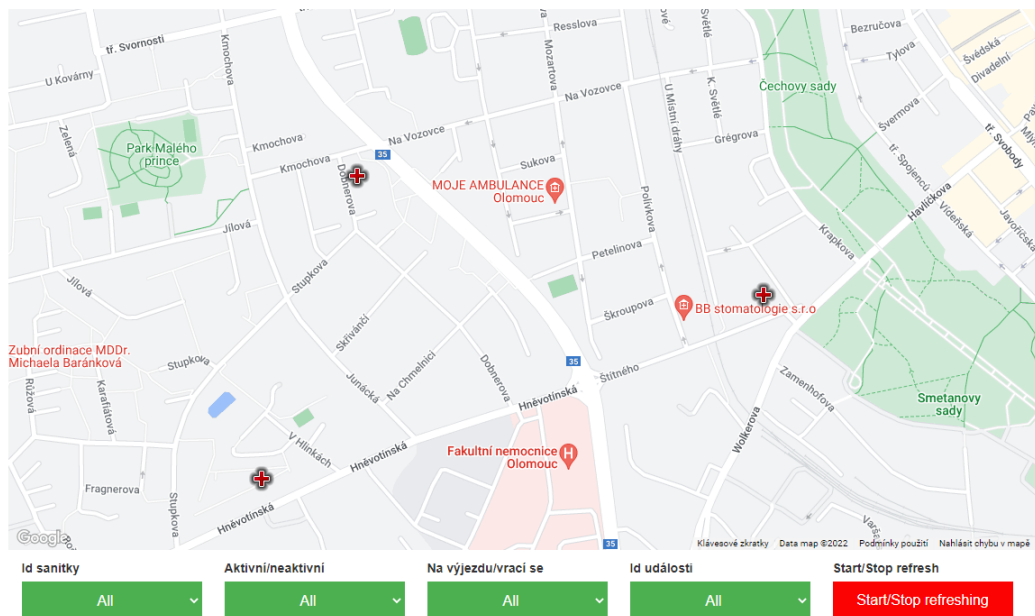
Operátor může uzavřít událost, na kterou nejsou vázané žádné aktivní sanitky. Pro potřeby testování operátor nemůže uzavřít události 1-3, které slouží pro simulační účely.

Operátor může zneaktivnit sanitky kromě sanitek 1-3, které stejně jako události slouží k simulačním účelům.

3.2.8 Tabulka aktivních událostí

V tabulce operátor vidí aktivní události a jejich id, adresu, čas vzniku události, naléhavost a popis události.

Přehled aktivních vozidel



Obrázek 8: Mapa

3.2.9 Tabulka záchranných vozidel

V tabulce operátor vidí vozidla spojená s aktivní událostí a její id a popis vozidla.

Simulace

- Sanitka 1
- Sanitka 2
- Sanitka 3
- Osobní auto 1
- Osobní auto 2
- Osobní auto 3
- Osobní auto 4
- Osobní auto 5

Založení události

Adresa:

Naléhavost:

Popis události:

Založil:
 | Indikace:
 | Co se stalo:
 | Poschodí:
 | Oznamovatel:
 | Telefon:
 | Jméno pacienta:
 | Věk pacienta:

Uložit

Vyslání sanitky

Událost číslo:

Popis vozidla:

Typ auta:
 | Posádka:

Uložit

Uzavření události

Událost číslo:

Smazat

Vymazání sanitky

Sanitka číslo:

Smazat

Obrázek 9: Okno simulace, zadávací a uzavírací formuláře

Aktivní události

Id	Adresa události	Čas vytvoření	Naléhavost	Popis
1	Olomouc Štítného 10	18:30:4 4/4/2022	1	Založil: Jiri Šesták Indikace: Bezvedomí Co se stalo: Pacient leží na zahradě v bezvedomí Poschodí: ne Oznamovatel: Pavel Novák Telefon: 777 777 777 Jméno pacienta: Kristián Dufek Věk pacienta: 85
2	Olomouc Dobnerova 718/26	18:31:4 4/4/2022	2	Založil: Jiri Šesták Indikace: Bezvedomí Co se stalo: Pacientovi se udelalo špatně a upadli do bezvedomí. Podarilo se jej přivést ke smyslům, ale nekomunikuje. Poschodí: 2. patro Oznamovatel: Pavel Dvorák Telefon: 555 555 555 Jméno pacienta: Patrik Jašek Věk pacienta: 66
3	Olomouc Hnevotínská 521	18:31:21 4/4/2022	4	Založil: Jiri Šesták Indikace: Zlomenina horní končetiny Co se stalo: Pacient spadol ze schodu Poschodí: ne Oznamovatel: Tomáš Kovár Telefon: 666 666 666 Jméno pacienta: Patrik Balog Věk pacienta: 35

Sanitky

Id	Aktivní	Id události	Popis
1	Neaktivní		
2	Neaktivní		
3	Neaktivní		

Obrázek 10: Tabulka pro aktivní události a tabulka pro sanitky

4 Programátorská dokumentace

4.1 Mobilní aplikace

Mobilní aplikace byla vyvíjena pomocí .NET Framework a Xamarin s využitím jazyka C#. .NET Framework je prostředí potřebné pro běh aplikací nabízející rozhraní a potřebné knihovny[1]. Xamarin je open source platforma pro vytváření aplikací pro iOS, Android a Windows s využitím C#, která rozšiřuje .NET Framework pro práci s mobilními zařízeními[2]. Projektová složka mobilní aplikace se skládá z projektů *ClientApp*, *ClientApp.Android* a *ClientApp.iOS*. Projekt *ClientApp.iOS* je určen pro vývoj aplikací s operačním systémem iOS, na který se práce nezaměřuje.

4.1.1 ClientApp

Projekt *ClientApp* obsahuje definici rozložení aplikace *MainPage.xaml*, která je sdílená pro oba operační systémy a její třídu *MainPage.xaml.cs*. Dále obsahuje třídu *ClientCarsModelClass*, která se využívá pro vytváření objektů a jejich odesílání pomocí HTTP dotazů serveru.

4.1.1.1 MainPage.xaml

Rozložení mobilní aplikace je definováno v XAML a obsahuje definice jednotlivých prvků, ze kterých se rozložení skládá. XAML je značkový jazyk založený na XML, který byl vyvinut společností Microsoft pro .NET Framework a Windows Presentation Foundation. Základem použitého rozložení je *Grid layout* rozdělený do 5 řádků a 3 sloupců. Prvky rozložení aplikace:

1. *mode_frame*: Rámec obalující popisek, který zobrazuje, v jakém režimu aktuálně aplikace běží.
2. *image_frame*: Rámec obalující obrázek.
3. *message_frame*: Rámec obalující popisek, který zobrazuje dodatečné informace k funkčnosti aplikace.
4. *switch_frame*: Rámec obalující přepínač, kterým se přepíná aplikace do/z aktivního/neaktivního režimu.
5. *popupWarning* a *popupCaution*: *Contentview* (třída rozložení), které zobrazují varování nebo upozornění při aktivním běhu aplikace.

4.1.1.2 MainPage.xaml.cs

Po spuštění aplikace se nastaví uživatelské rozhraní na neaktivní režim, ve kterém čeká na přepnutí uživatelem do aktivního režimu. Po přepnutí přepínače do aktivní polohy kontroluje metoda *OnToggled* oprávnění přístupu k poloze zařízení a připojení zařízení k Internetu. Poté spouští metodu *StartSending*.

4.1.1.3 StartSending

Metoda *StartSending* provádí velkou část funkcionality aplikace. Hlavní částí je cyklické odesílání HTTP dotazů webovému serveru. Po prvním spuštění pošle HTTP POST s objektem třídy *ClientCarViewModel*, ve které jsou uloženy souřadnice zařízení. Po odezvě z webového serveru si ukládá přidělené *Id* a případně reaguje na odpověď serveru, pokud je v odpovědi varování nebo upozornění. Jakmile má metoda přidělené *Id* serverem, tak pošle své souřadnice serveru v HTTP PUT s daným *Id*. Aplikace si nastavuje interval odesílání souřadnic serveru podle vzdálenosti k nejbližšímu záchrannému vozidlu. V reakci na upozornění nebo varování jej zobrazuje v uživatelském rozhraní aplikace. Pokud aplikace pošle více než 10 stejných souřadnic (zařízení se nehýbe), nebo je aplikace přepnuta do neaktivního režimu, je odeslán HTTP DELETE serveru.

Délka intervalu odesílání souřadnic podle vzdálenosti od nejbližšího aktivního záchranného vozidla:

1. Vzdálenost je větší než 5 km – 30 vteřin
2. Vzdálenost je menší než 5 km a větší než 3 km – 15 vteřin
3. Vzdálenost je menší než 3 km a větší než 1 km – 5 vteřin
4. Jinak - 2 vteřiny

4.2 Databáze

Databáze *CarLocationsDB* je vytvořena v Microsoft SQL Server Management Studio. Obsahuje tabulky *RescueCarsTable* pro záchranná vozidla, *CarsTable* pro zařízení uživatelů a *EventsTable* pro události.

4.2.1 RescueCarsTable

Obsahuje:

1. *Id* – Primární klíč
2. *Latitude* – Aktuální zeměpisná šířka
3. *Longitude* – Aktuální zeměpisná délka
4. *FinishLatitude* – Cílová zeměpisná šířka
5. *FinishLongitude* – Cílová zeměpisná délka
6. *InAction* – Aktivní záchranné vozidlo (true/false)
7. *Description* – Popis
8. *IdEvent* – *Id* události na kterou je vozidlo navázané

4.2.2 CarsTable

Obsahuje:

1. *Id* – Primární klíč
2. *Latitude* – Aktuální zeměpisná šířka
3. *Longitude* – Aktuální zeměpisná délka
4. *OnRoute* – Zařízení se nachází na trase nějakého záchranného vozidla (true/false)
5. *IdRescueCarByRoute* – *Id* záchranného vozidla, na jehož trase se zařízení nachází
6. *DistanceFromNearestRescueCar* – Vzdálenost od nejbližšího aktivního záchranného vozidla
7. *IdNearestRescueCarByDistance* – *Id* nejbližšího aktivního záchranného vozidla
8. *UpdateTime* – Čas poslední aktualizace souřadnic zaslané serveru
9. *UUID* – unikátní identifikátor ve formátu xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx obsahující znaky a-z, A-Z a čísla 0-9

4.2.3 EventsTable

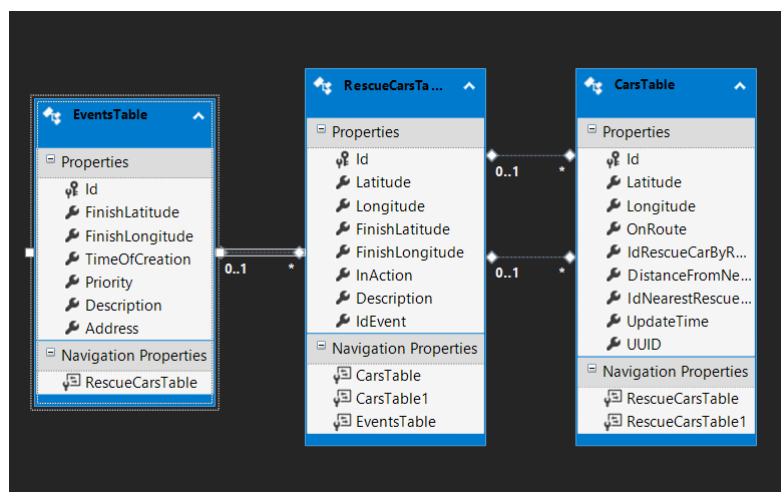
Obsahuje:

1. *Id* – Primární klíč
2. *FinishLatitude* – Cílová zeměpisná šířka
3. *FinishLongitude* – Cílová zeměpisná délka
4. *TimeOfCreation* – Čas vytvoření události
5. *Priority* – Priorita
6. *Description* – Popis
7. *Address* – Adresa

4.3 Webový server

Webový server je implementován pomocí ASP.NET Web API 2, což je framework umožňující implementaci webových služeb založených na HTTP dotazech, které jsou přístupné z různých platform a aplikací. Server je spojený s databází *CarLocationsDB* pomocí webové služby *RescueCarsDataAccess*, která umožňuje webovému serveru komunikaci s databází.

Hlavními částmi webového serveru jsou Controllers, Models a Views.



Obrázek 11: ADO.NET entity data model

4.3.1 Models

Models obsahují ViewModels třídy pro každou tabulku databáze a používají se v kontrolerech pro přístup k databázi.

1. *RescueCarsViewModel* – pro záchranná vozidla
2. *ClientCarViewModel* – pro uživatelská zařízení
3. *EventsViewModel* – pro události

```

1 using System;
2
3 namespace RescueCarsServer.Models
4 {
5     //třída pro vytváření objektů záchranných vozidel
6     public class RescueCarsViewModel
7     {
8         public int Id { get; set; }
9         public double Latitude { get; set; }
10        public double Longitude { get; set; }
11        public double? FinishLatitude { get; set; }
12        public double? FinishLongitude { get; set; }
13        public bool InAction { get; set; }
14        public String Description { get; set; }
15
16        public int? IdEvent { get; set; }
17    }
18 }

```

Zdrojový kód 1: Model pro záchranné vozidlo

4.3.2 Controllers

Kontrolery definují jakým způsobem bude server odpovídat na HTTP dotazy, které mu přijdou z uživatelských aplikací. V projektu *RescueCarsServer* existuje 5 kontrolerů:

1. *HomeController*
2. *RescueCarsController*
3. *ClientCarsController*
4. *EventsController*
5. *StandartController*
6. *MobileAppController*

HomeController implementuje GET metody pro každou z tabulek databáze. Vrací pro danou tabulku databáze výpis všech záznamů tabulky.

RescueCarsController implementuje HTTP metody pro tabulku *RescueCarsTable*.

- Metoda *GetAll()* je HTTP GET metoda, která vrací všechny záznamy z tabulky *RescueCarsTable*.
- Metoda *GetAllRescueCars()* je HTTP GET metoda, která vrací všechny záznamy z tabulky *RescueCarsTable*, které odpovídají zadaným filtrům na hlavní stránce webového serveru.
- Metoda *PutRescueCar(RescueCarsViewModel car)* je HTTP PUT metoda, která aktualizuje data v databázi pro instanci *RescueCarsViewModel* daného *Id*.
- Metoda *PostRescueCar(RescueCarsViewModel car)* je HTTP POST metoda, která ukládá instanci *RescueCarsViewModel* a vrací vygenerované *Id*, pokud se uložení zdařilo.
- Metoda *DeleteRescueCar (int id)* je HTTP DELETE metoda, která z databáze maže záznam s daným *Id*.

ClientCarsController implementuje HTTP metody pro tabulku *CarsTable*.

- Metoda *GetOldCars()* je HTTP GET metoda, která vrací všechny záznamy z tabulky *CarsTable*, které mají v atributu *UpdateTime* čas starší než 30 minut.

- Metoda *PutCar(ClientCarViewModel car)* je HTTP PUT metoda, která aktualizuje data v databázi pro instanci *ClientCarViewModel* daného *Id*. Pokud je vzdálenost mezi zařízením a aktivním záchranným vozidlem menší než 200 metrů a zařízení je na trase tohoto záchranného vozidla, tak metoda vrací varování, pokud zařízení není na trase daného vozidla, vrací upozornění.
- Metoda *PostClientCar(ClientCarViewModel car)* je HTTP POST metoda, která ukládá objekt *ClientCarViewModel* a vrací vygenerované *Id*, pokud se uložení zdařilo.
- Metoda *DeleteClientCar (int id)* je HTTP DELETE metoda, která z databáze maže záznam s daným *Id*.

EventsController implementuje HTTP metody pro tabulku *EventsTable*.

- Metoda *GetAll()* je HTTP GET metoda, která vrací všechny záznamy z tabulky *EventsTable*.
- Metoda *PostEvent(EventsViewModel newEvent)* je HTTP POST metoda, která ukládá objekt *EventsViewModel*.
- Metoda *DeleteEvent (int id)* je HTTP DELETE metoda, která z databáze maže záznam s daným *Id*.

MobileAppController je pomocný kontroler, který implementuje HTTP GET metodu *GetAllCars()*, která vrací všechny záznamy z tabulky *CarsTable*.

StandartController je pomocný kontroler, který implementuje HTTP GET metodu *GetEvent()*, která vrací událost podle daného *Id* *CarsTable*.

4.3.3 Views

Views obsahují soubory formátu *cshtml*, které utváří design a logiku hlavní stránky webového serveru.

- *Layout.cshtml* – obsahuje html definici stránky
- *Index.cshtml* – obsahuje logiku zapsanou v Javascriptu

4.3.4 Fungování webového serveru

Soubor *Index.cshtml* obsahuje skripty, které se vykonávají po spuštění webového serveru.

- Skripty pro načtení dat z databáze a naplnění tabulek pro události a Záchrannářská vozidla.
- Skripty pro naplnění filtrů tak, aby se dala filtrovat aktuální data na mapě.

- Skripty pro zadávací formuláře události záchranného vozidla.
- Hlavní skript webového serveru obsluhující mapu a vnitřní logiku serveru.

Hlavní skript s funkcí *initMap()*

Funkce *initMap()* zajišťuje načítání dat z databáze pomocí kontrolerů, výpočet tras záchranných vozidel, porovnávání poloh záchranných vozidel a uživatelských zařízení a jejich vykreslování do mapy. K zobrazení mapy, vykreslení ikoněk, tras a výpočtu jejich umístění využívá logika webového serveru funkcionalit *Google Maps API*.

Proměnné

- *rescueCarsMarkers* – pole ikoněk záchranných vozidel k vykreslení
- *rescueCarsDescriptions* – pole definic popisů záchranných vozidel
- *rescueCarsInfos* – pole hodnot popisů záchranných vozidel
- *carsMarkers* – pole ikoněk uživatelských zařízení
- *messageArray* – pole pro uchovávání zasláných zpráv zařízením
- *arrayOfRescueCars* – pomocné pole pro uchování záchranných vozidel

Průběh *initMap()*

Funkce *initMap()* inicializuje mapu webového serveru. Funkce dále volá funkci *setRescueCars()* a po jejím dokončení funkci *setCars*. Jelikož obě metody používají asynchronní HTTP dotazy, je potřeba je volat pomocí *Promises*.

Funkce *setRescueCars()* si uloží hodnoty zadané ve filtrech mapy a pošle HTTP dotaz kontroleru *RescueCars*. Pokud je request úspěšný a vrácena data nejsou prázdná, začíná v cyklu tato data procházet. Každé záchranné vozidlo vkládá do pomocného pole *arrayOfRescueCars*, které bude později použito ve funkci *setCars()*. Poté kontroluje jestli má daný vůz nastavenou cílovou polohu a jestli je přidělen k aktivní události. Pokud se vůz dostane do vzdálenosti menší než 10 metrů od cílové destinace, posílá HTTP POST webovému serveru s nastavenou cílovou lokací na výchozí nemocnici, protože dorazil k místu dané události a bude se vracet zpět. Následně vykreslí do mapy svoji ikonku na základě aktuální lokace a aktuální trasy do cílové lokace.

Funkce *setCars()* se spouští po dokončení funkce *setRescueCars()*. Pošle HTTP GET request kontroleru *ClientCars*, který mu vrátí všechna aktivní uživatelská zařízení v provozu. Pokud není *arrayOfRescueCars* (které se plní ve funkci *setRescueCars()*) prázdné a zároveň data vrácená requestem také nejsou prázdná, pokračuje funkce cyklem, který prochází *arrayOfRescueCars*. Pokud ale data vrácená requestem obsahují aktivní zařízení a zároveň nejsou žádné aktivní záchranné vozy, je přeskočeno veškeré porovnávání se záchrannářskými vozy a jsou pouze vykresleny ikonky zařízení.

Pro každé záchranářské vozidlo si funkce vypočítá jeho aktuální trasu a ve vnořeném cyklu prochází data (pole aktivních uživatelských zařízení), kde jako první nastaví zařízení *DistanceFromNearestRescueCar* a *IdNearestRescueCarByDistance*, což jsou vzdálenost od nejbližšího aktivního záchranářského auta a jeho Id. Poté funkce pokračuje jednou ze čtyřech možností:

1. Zařízení je na trase záchranného vozu a zároveň už má nastaveno, že je na trase jiného aktivního záchranného vozu. Vypočítá se tedy, daný záchranný vůz blíže než ten, na jehož trase už zařízení je. Potom se uloží *Id* bližšího záchranného vozu do *IdRescueCarByRoute* zařízení a pošle se HTTP POST webovému serveru s aktualizovanými daty zařízení. Na základě vrácených dat z webového serveru se potom zařízení nastaví ikonka pro varování, upozornění, nebo se nechá původní ikonka značící pouze aktivní zařízení.
2. Zařízení je na trase záchranného vozu, ale není na trase žádného jiného aktivního záchranného vozu. Pošle se tedy HTTP POST webovému serveru s aktualizovanými daty zařízení, kde se přidá aktuální záchranný vůz do *IdRescueCarByRoute*. Na základě vrácených dat z webového serveru se potom zařízení nastaví ikonka pro varování, upozornění, nebo se nechá původní ikonka značící pouze aktivní zařízení.
3. Zařízení není na trase záchranného vozu a zároveň už má nastaveno, že je na trase jiného záchranného vozu. Pošle se tedy HTTP POST webovému serveru s aktualizovanými daty.
4. Zařízení není na trase záchranného vozu a není na trase žádného jiného aktivního záchranného vozu. Pošle se tedy HTTP POST webovému serveru s aktualizovanými daty.

Funkce *setCars()* tedy ve vnějším cyklu prochází aktivní záchranná vozidla a ve vnitřním cyklu aktivní uživatelská zařízení, kterým nastavuje nejbližší aktivní záchranné vozidlo a popřípadě vozidlo, na jehož trase se zařízení nachází. Z důvodu omezení volání *Google Maps API* se prochází záchranná vozidla, kvůli kterým se volá funkce *Google Maps API* na vypočtení a vykreslení trasy. Tím se minimalizuje počet dotazů *Google Maps API*.

Funkce *refreshRescueCars()* volá použitím *Promises setRescueCars* a *setCars* a je cyklicky volána po stisknutí tlačítka *Start refresh* a její volání je ukončeno stiskem tlačítka *Stop refresh*.

Vykreslování ikoněk uživatelských zařízení, stejně jako tlačítka pro spuštění a zastavení obnovování mapy jsou součástí webového serveru čistě z testovacích důvodů a lepší vizualizace.

5 Testování

Aktivní uživatelské zařízení přijímá dva typy notifikací, což jsou varování a upozornění. Server se rozhoduje jakou notifikaci poslat na základě hodnot uložených v attributech uživatelského zařízení, které si načítá z databáze.

Jsou to:

- *OnRoute* – nachází/nenachází se na trase nějakého záchranného vozidla
- *IdRescueCarByRoute* – pokud se nachází na trase vozidla, jaké je jeho *Id*
- *DistanceFromNearestRescueCar* – vzdálenost od nejbližšího vozidla
- *IdNearestRescueCarByDistance* – *Id* nejbližšího vozidla

5.1 Varování

Server zasílá varování aktivnímu uživatelskému zařízení, pokud se *IdRescueCarByRoute* a *IdNearestRescueCarByDistance* zařízení rovnají. To znamená, že je zařízení na trase záchranného vozidla na výjezdu, které je zároveň nejbližším vozidlem na výjezdu a vzdálenost mezi uživatelským zařízením a záchranným vozidlem je kratší než 200 metrů. Uživatel (řidič) tedy dostává varování s dostatečným předstihem a vytvoří záchrannářskou uličku včas.

Při provozu mohou nastat dva typy situací:

1. Uživatelské zařízení je v dosahu a na trase pouze jednoho vozidla na výjezdu – Server uživatelskému zařízení pošle varování.
2. Uživatelské zařízení je v dosahu a na trase více než jednoho vozidla na výjezdu – Server uživatelskému zařízení pošle varování a uloží bližší vozidlo a vzdálenost od něj do *DistanceFromNearestRescueCar* a *IdNearestRescueCarByDistance*.

Při druhé variantě situace se může stát, že v jednom okamžiku bude řidič v dosahu a na trasách dvou (nebo více) záchranných vozů. Dostane varování a bližší vozidlo projede a vzdálí se. V této situaci dostane řidič druhé varování od druhého vozidla na výjezdu, pokud bude stále v jeho dosahu. Nemůže se tedy stát, že by řidič vytvořil prostor pro bližší vozidlo na výjezdu a následně způsobil dopravní nehodu kvůli druhému vozidlu na výjezdu, které kolem něj ještě neprojelo.

5.2 Upozornění

Server zasílá upozornění aktivnímu uživatelskému zařízení, pokud je vzdálenost mezi uživatelským zařízením a nejbližším záchranným vozidlem kratší než 200 metrů. Upozornění nepožaduje po řidiči vytvoření záchrannářské uličky, protože se uživatel nenachází přímo na trase vozidla na výjezdu, ale nabádá ho ke zvýšení

pozornosti, pokud by došlo ke změně trasy záchranného vozidla. V tomto případě je řidič s dostatečným předstihem informován o pohybu vozidla na výjezdu v jeho blízkém okolí a je tím pádem schopný pohotově reagovat a případně vytvořit prostor pro průjezd záchranného vozidla.

5.3 Rozhodování

Poslední situací, kterou může nastat, je aktivita dvou nebo více vozidel na výjezdu ve vzdálenosti kratší než 200 metrů, a zároveň kolize uživatelského zařízení s trasou pouze jednoho z nich. V tomto případě dostává uživatelské zařízení notifikaci bližšího vozidla na výjezdu, protože se předpokládá, že to projede dříve. Uživatel tedy dostane upozornění, pokud je blíže vozidlo na výjezdu, na jehož trase uživatelské zařízení není. Varování dostane, pokud je blíže vozidlo na výjezdu, na jehož trase se uživatel nachází.

5.4 Tesovací simulace na serveru

Na domovské stránce serveru je okno *Simulace*, které slouží pro otestování funkčnosti posílání varování a upozornění. V simulaci jsou dostupné 3 sanitky a 5 klientských aut. V každé simulaci lze spustit pouze jedna sanitka a libovolný počet klientských aut. Počet sanitek je omezený z důvodu použití bezplatné verze Google Maps API, která je omezená na počet přístupů, výpočet tras a lokací. Pokud je překročen tento limit, server nefunguje korektně, protože dotazy na Google Maps API nevrací potřebná data. Pro korektní průběh simulace také doporučuji uzavřít vytvořené události a sanitky tak, aby při simulaci byly na serveru pouze 3 testovací sanitky a 3 události. Nebo stačí spustit refresh na pár sekund a znovu ho zastavit. Server se se zastavením refreshu vrací do konzistentního stavu pro simulaci.

Každý testovací scénář se skládá ze sekvence 30 souřadnic, které se postupně odesílají a simulují pohyb vozidel.

5.4.1 Hlavní testovací scénáře

- Sanitka 1, Vozidlo 1,2,3
- Sanitka 2, Vozidlo 1,3,5
- Sanitka 3, Vozidlo 1,2,3

5.4.2 První testovací scénář

Sanitka 1 začíná na ulici Hněvotínská u výjezdu z areálu Fakultní nemocnice Olomouc. Auto 1 začíná na ulici Kmochova, auto 2 na ulici V Hlinkách a auto 3 na ulici Dobnerova. 13.



Obrázek 12: Trasy klientů aut

5.4.2.1 Auto 1 v simulačním scénáři 1

Auto se v tomto testovacím scénáři nedostane na trasu sanitky. Při vjezdu na ulici Foerstrova (na mapě silnice číslo 35) dostane upozornění, že se nachází aktivní sanitka v okruhu 200 metrů 16. Auto má v simulačním scénáři až do konce simulace ikonku upozornění, protože jeho vzdálenost od sanitky je stále méně než 200 metrů. V reálné situaci při provozu, by řidič odklikl upozornění a dbal by zvýšené pozornosti, protože se nachází v okolí aktivní sanitky.

5.4.2.2 Auto 2 v simulačním scénáři 1

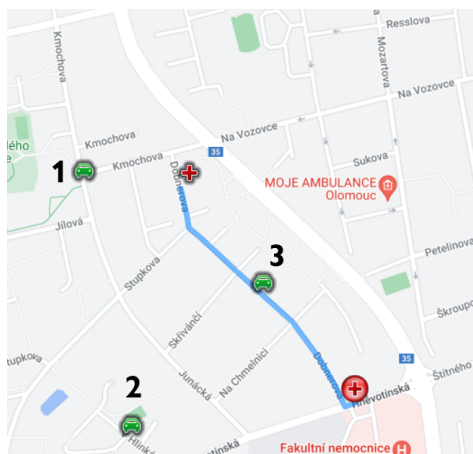
Auto dostává upozornění, když se blíží ke křižovatce ulic Na Chmelnici a Dobnerova 17. Vzápětí při vjezdu do křižovatky dostává auto varování k vytvoření záchranářské uličky 18. Poté pokračuje po ulici Dobnerova, kde se jeho ikonka změní na zelenou (bez upozornění/varování), jakmile se dostane dál než 200 metrů od aktivní sanitky 19.

5.4.2.3 Auto 3 v simulačním scénáři 1

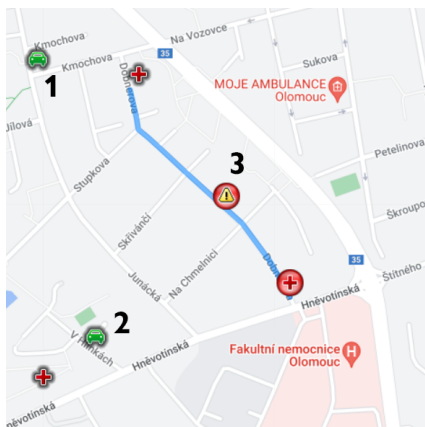
Auto dostává krátce po startu simulace varování, protože je na trase sanitky 14. Jakmile se dostane z dosahu 200 metrů, změní se jeho ikona na zelenou 15.

5.4.3 Druhý testovací scénář

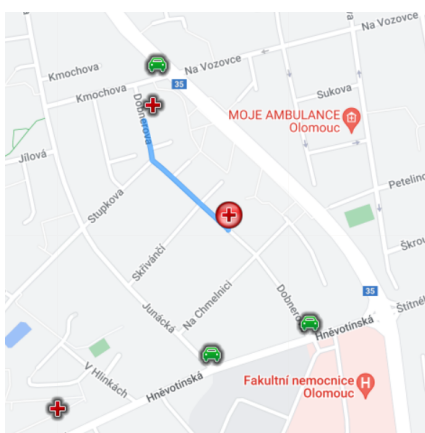
Sanitka 2 začíná na ulici Hněvotínská u výjezdu z areálu Fakultní nemocnice Olomouc. Auto 1 začíná na ulici Kmochova, auto 2 na ulici V Hlinkách a auto 3 na ulici Dobnerova. 20.



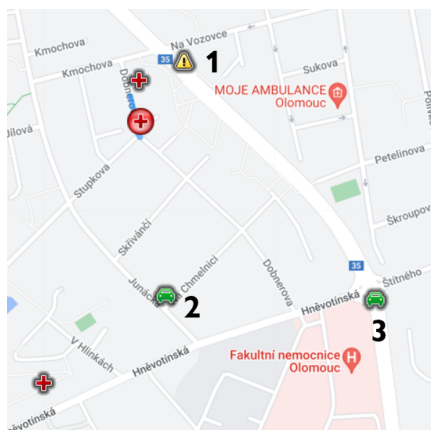
Obrázek 13: Začátek simulačního scénáře 1



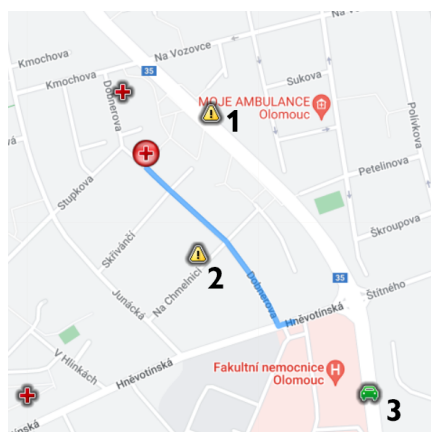
Obrázek 14: Scénář 1 - Auto 3 dostává varování



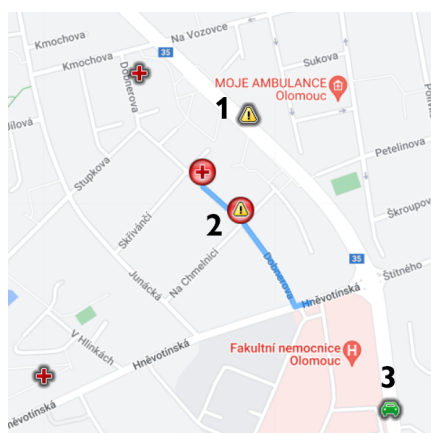
Obrázek 15: Scénář 1 - Auto 3 je z dosahu sanitky



Obrázek 16: Scénář 1 - Auto 1 dostává upozornění



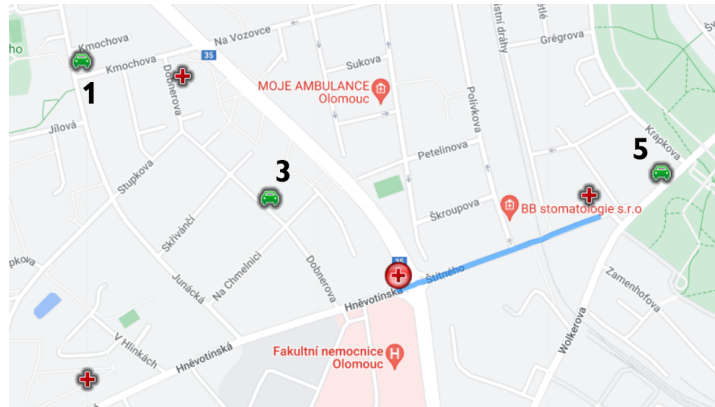
Obrázek 17: Scénář 1 - Auto 2 dostává upozornění



Obrázek 18: Scénář 1 - Auto 2 dostává varování



Obrázek 19: Scénář 1 - Auto 2 je z dosahu sanitky



Obrázek 20: Začátek simulačního scénáře 2

5.4.3.1 Auto 1 v simulačním scénáři 2

Auto dostává upozornění až ke konci simulace, když přijíždí na křižovatku ulic Foerstrova, Štítného a Hněvotínská. 22.

5.4.3.2 Auto 3 v simulačním scénáři 2

Auto dostává upozornění stejně jako auto 1 až při příjezdu ke křižovatce silnic. 23.

5.4.3.3 Auto 5 v simulačním scénáři 2

Auto dostává varování jakmile se dostane na trasu sanitky a je blíže než 200 metrů 21.

5.4.4 Třetí testovací scénář

Sanitka 1 začíná na ulici Hněvotínská u výjezdu z areálu Fakultní nemocnice Olomouc. Auto 1 začíná na ulici Kmochova, auto 2 na ulici V Hlinkách (dostává



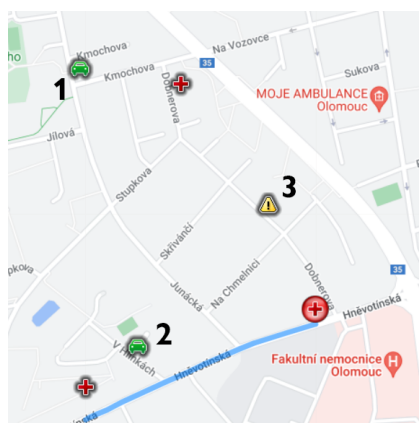
Obrázek 21: Scénář 2 - Auto 5 dostává varování



Obrázek 22: Scénář 2 - Auto 1 dostává upozornění



Obrázek 23: Scénář 2 - Auto 3 dostává upozornění



Obrázek 24: Začátek simulačního scénáře 3

hned na začátku simulace upozornění) a auto 3 na ulici Dobnerova. 27.

5.4.4.1 Auto 1 v simulačním scénáři 3

Auto dostává upozornění na konci simulace, když se blíží na křižovatku ulic Foerstrova, Štítného a Hněvotínská. 22.

5.4.4.2 Auto 2 v simulačním scénáři 3

Auto dostává varování, protože leží na trase sanitky 25. Jakmile se dostane z jejího dosahu, přepne se jeho ikonka na zelenou 26.

5.4.4.3 Auto 3 v simulačním scénáři 3

Auto dostává upozornění hned na začátku simulace. Jakmile se vzdálí z dosahu sanitky, přepne se jeho ikona na zelenou 26. Na konci simulace se auto dostává opět do dosahu sanitky a dostává upozornění. 27.

5.4.5 Auto 4

Auto 4 má záměrně trasu mimo dosah všech sanitek, aby šlo otestovat, že se mu neposílají žádná upozornění ani varování.



Obrázek 25: Scénář 3 - Auto 2 dostává varování



Obrázek 26: Scénář 3 - Auto 2 a 3 jsou z dosahu sanitky



Obrázek 27: Scénář 2 - Auto 1 a 3 dostávají upozornění

6 Google Maps API

V závěrečné práci je využita neplacená licence Google Maps API určená pro nekomerční využití a vývoj, což se pojí s omezeními na výkon a množství dotazů.

6.1 Maps Javascript API

Javascript API umožňuje vykreslovat mapy na webových stránkách, serverech a mobilních zařízeních. Webový server využívá Javascript API ke zobrazení mapy na hlavní stránce serveru a vykreslování ikonek (markerů) na mapě.[3]

6.2 The Directions API

The Direction API je webová služba *Google Maps API*, která je založena na HTTP dotazování a vrací vygenerované trasy ve formátu JSON nebo XML. Webový server využívá Direction API při generování a vykreslení tras jednotlivých záchranných vozů do mapy.[4]

- *DirectionsService* – vrací vygenerovanou trasu
- *DirectionsRenderer* – vykresluje vygenerovanou trasu do mapy

6.3 Geocoding API

Geocoding API umožňuje vyhledávat lokace podle zadané adresy a pracovat s danými souřadnicemi. Webový server využívá Geocoding API pro získání souřadnic ze zadané adresy operátorem při zadávání nové události.[5]

6.4 Problémy využití Google Maps API

Využití bezplatné omezené licence Google Maps Api pro vývoj a nekomerční užití, se ukázalo jako problémové při větším počtu záchranných vozů a uživatelských zařízení k vykreslení do mapy. Licence pro vývoj je omezená počtem dotazů poslaných Google Maps API, což může vést k chybovosti vykreslování na mapě. Druhým problémem je obsáhlé použití asynchronních volání, která nelze ošetřit jinak, než čekáním na dokončení všech volání na úkor rychlosti reakcí serveru. Jelikož je vykreslování poloh uživatelských zařízení účelově pouze pro vizualizaci a testování, tak je dána přednost plynulosti práce webového serveru a rychlosti odezvy na varování a upozornění.

```

1 using RescueCarsDataAccess;
2 using System;
3 using System.Linq;
4 using System.Web.Http;
5
6 namespace RescueCarsServer.Controllers
7 {
8     public class MobileAppController : ApiController
9     {
10         public IHttpActionResult GetAll()
11         {
12             try
13             {
14                 using (CarLocationsDBEntities entities = new
15                     CarLocationsDBEntities())
16                 {
17                     var carsData = entities.CarsTable.Select(e => new
18                     {
19                         Id = e.Id,
20                         Latitude = e.Latitude,
21                         Longitude = e.Longitude,
22                         DistanceFromNearestRescueCar = e.
23                             DistanceFromNearestRescueCar,
24                         IdNearestRescueCarByDistance = e.
25                             IdNearestRescueCarByDistance,
26                         OnRoute = e.OnRoute,
27                         IdRescueCarByRoute = e.IdRescueCarByRoute,
28                         UpdateTime = e.UpdateTime,
29                         UUID = e.UUID
30                     }).ToList();
31                     return Json(carsData);
32                 }
33             }
34             catch (Exception ex)
35             {
36                 return InternalServerError(ex);
37             }
38         }
39     }
40 }

```

Zdrojový kód 2: Kontroler MobileAppController

Závěr

Funkční řešení problému včasného varování řidičů při průjezdu vozidel záchranných a bezpečnostních složek, které by bylo v provozu, zatím neexistuje. Cílem této práce proto bylo zaměřit se na možná řešení tohoto problému. Jako logické se jevílo řešení pomocí mobilní aplikace jakožto klientského zařízení a zjednodušeného webového serveru, imitujícího server záchranných složek. Bylo očividné, že toto řešení nebude možné použít ve velkém měřítku na skutečný provoz. Z praktického hlediska by záležel dopad tohoto systému na průjezd vozidel záchranných složek dopravou na počtu aktivních uživatelů aplikace a jejich ochotě před každou jízdou aplikaci zapnout. Z právního hlediska by bylo obtížné napojení aplikace na servery záchranných a bezpečnostních složek. Proto jsem se při implementaci tohoto řešení zaměřil hlavně na základní koncept řešení problému s včasným varováním řidičů, který by mohl sloužit jako základní myšlenka pro budoucí použití. Samotná logika, kterou jsou porovnávány polohy uživatelů a vozidel na výjezdu, spolu s logikou posílání notifikací, by se v budoucí době dalo využít například pro nasazení v autonomních vozidlech za použití 5G sítě nebo chytrých křižovatkách.

Conclusions

The working solution for a swift warning of drivers when an emergency car is about to pass by does not exist at this time. The goal of this thesis was to focus on possible solutions of this problem. As a logical approach seemed a solution in a form of mobile application as a client device and a simplified web server imitating emergency services' server. It was obvious, that this solution could not be used on real traffic at such a large scale. From a practical standpoint, the efficiency of the system to assist the passage of emergency vehicles through traffic would depend on the number of active users and their willingness to turn on the application before driving. From a legal standpoint, it would be difficult to connect the application to the emergency services' servers. That is why I have focused my efforts during implementation of the solution on the basic concept of the swift warning of drivers, which could serve as a basis for future use. The logic itself by which the user locations and locations of emergency vehicles are compared, as well as notification sending logic, could be used in the future for usage in autonomous vehicles using 5G network or at smart intersections.

A Obsah přiloženého CD/DVD

Source/

Kompletní zdrojové soubory mobilní aplikace ClientApp a projektové složky RescueCarsServer obsahující zdrojové soubory RescueCarsServer a RescueCarsDataAccess.

ClientApp.apk

Instalační soubor mobilní aplikace ClientApp.

readme.txt

Instrukce ke spuštění všech programů a další informace.

Bakalářská práce.zip

Text práce v zip archivu.

Bakalářská práce.pdf

Text práce v pdf.

Literatura

- [1] *Desktopová příručka pro .NET, .NET Core a .NET Framework. : Dokumentace k Windows Presentation Foundation.* 2021. Dostupný z: [⟨https://docs.microsoft.com/cs-cz/dotnet/desktop/wpf/?view=netdesktop-5.0⟩](https://docs.microsoft.com/cs-cz/dotnet/desktop/wpf/?view=netdesktop-5.0).
- [2] *Dokumentace pro Xamarin. : Dokumentace ke Xamarin.Forms.* 2021. Dostupný z: [⟨https://docs.microsoft.com/cs-cz/xamarin/xamarin-forms/⟩](https://docs.microsoft.com/cs-cz/xamarin/xamarin-forms/).
- [3] *Google Maps Platform Documentation. : Maps Javascript API.* 2021. Dostupný z: [⟨https://developers.google.com/maps/documentation/javascript/overview⟩](https://developers.google.com/maps/documentation/javascript/overview).
- [4] *Google Maps Platform Documentation. : The Directions API.* 2021. Dostupný z: [⟨https://developers.google.com/maps/documentation/directions/overview⟩](https://developers.google.com/maps/documentation/directions/overview).
- [5] *Google Maps Platform Documentation. : Geolocation API.* 2021. Dostupný z: [⟨https://developers.google.com/maps/documentation/geolocation/overview⟩](https://developers.google.com/maps/documentation/geolocation/overview).
- [6] *Desktopová příručka pro .NET, .NET Core a .NET Framework. : Dokumentace k Windows Presentation Foundation.* 2021. Dostupný z: [⟨https://docs.microsoft.com/cs-cz/dotnet/desktop/wpf/?view=netdesktop-5.0⟩](https://docs.microsoft.com/cs-cz/dotnet/desktop/wpf/?view=netdesktop-5.0).