

# Mobilní aplikace pro rozpoznávání registračních značek vozidel

Stanislav Král

Vedoucí práce: Ing. Kamil Ekštein, PhD.

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

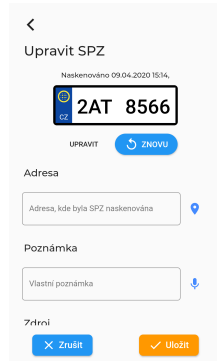
2020

# Obsah prezentace

- 1 Úvod
- 2 Implementace
- 3 Oprava chyby ve frameworku Flutter
- 4 Výsledky

- nedostatek volně dostupného SW pro rozpoznávání registračních značek vozidel **mobilními telefony**
- nabídnout veřejnosti možnost postavit SW stavějící na této funkcionalitě
  - evidence vozidel v autoservisech, placených parkovištích,...
  - monitorování pohybu vozidel v dočasně zřízených kontrolních stanovištích
  - pořizování důkazního materiálu při pojistných událostech


- mobilní aplikace pro platformy **Android** a **iOS** vytvořená s využitím frameworku **Flutter**
- kontinuální snímání scény fotoaparátem
- rozpoznávání českých RZ ve scéně realizováno pomocí OCR s kontrolou formátu identifikátoru
- naskenované RZ a dodatečné informace ukládané do lokální **SQLite** databáze
  - adresa místa pořízení (bud' ručně zadaná nebo automaticky dohledaná dle GPS souřadnic)
  - vlastní textová nebo hlasová poznámka




<


Upravit SPZ

Naskenováno 09.04.2020 15:14,


 **2AT 8566**

UPRAVIT  ZNOVU



Adresa

Adresa, kde byla SPZ naskenována 

Poznámka

Vlastní poznámka 

Zdroj

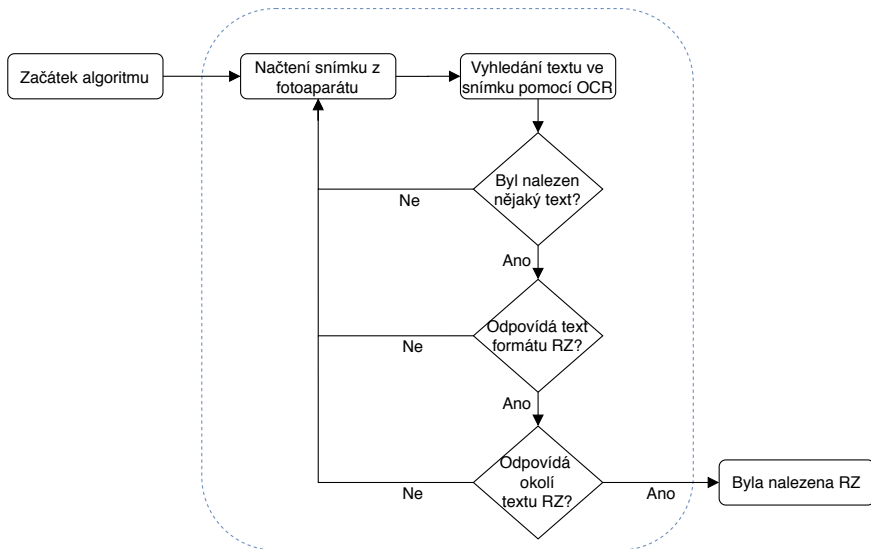
 Zrušit  Uložit



- jednořádkové RZ
  - klasické
  - historické
  - VIP
- dvouřádkové RZ
  - o rozměrech 340 × 200 mm
  - RZ motocyklů



# Diagram rozpoznávání RZ



# Oprava chyby ve frameworku Flutter

- během vývoje objevena chyba v balíčku camera
  - snímání scény pomocí fotoaparátu se ne vždy chovalo korektně
  - tato chyba se projevovala zřídka, čímž bylo její hledání ztížené
  - někdy zapříčinila, že při skenování dvou různých aut byla v obou případech rozpoznána RZ prvního auta
- bylo třeba analyzovat balíček camera z frameworku Flutter a chybu odhalit
- následně provést její opravu, a vše řádně zdokumentovat
- popis včetně opravy předán vývojářům tohoto frameworku pomocí služby **GitHub**<sup>1</sup>

---

<sup>1</sup>GitHub pull request - <https://github.com/flutter/plugins/pull/2655>

- rozšiřitelná **multiplatformní** mobilní aplikace rozpoznávající RZ
- implementovaná metoda rozpoznávání je rychlá a přesná
- poradí si i se **zhoršenými světelnými podmínkami**



- rozšiřitelná **multiplatformní** mobilní aplikace rozpoznávající RZ
- implementovaná metoda rozpoznávání je rychlá a přesná
- poradí si i se **zhoršenými světelnými podmínkami**

Galerie	P	FN	FP	$a$	$\bar{a}$	$t$ [ms]	$\bar{t}$ [ms]	Přesnost [%]
Slunečno	21	0	0	33	1,57	6769	322	100
Zataženo	5	0	0	7	1,40	1288	258	100
Podvečer	17	1	0	30	1,67	6184	344	94,45
Večer	19	3	1	52	2,26	10093	439	82,60

**Tabulka:** Výsledky testování provedené na zařízení Oneplus 6.

- rozšiřitelná **multiplatformní** mobilní aplikace rozpoznávající RZ
- implementovaná metoda rozpoznávání je rychlá a přesná
- poradí si i se **zhoršenými světelnými podmínkami**

Galerie	P	FN	FP	$a$	$\bar{a}$	$t$ [ms]	$\bar{t}$ [ms]	Přesnost [%]
Slunečno	21	0	0	33	1,57	6769	322	100
Zataženo	5	0	0	7	1,40	1288	258	100
Podvečer	17	1	0	30	1,67	6184	344	94,45
Večer	19	3	1	52	2,26	10093	439	82,60

**Tabulka:** Výsledky testování provedené na zařízení OnePlus 6.

- P** – počet scénářů, ve kterých byla správně rozpoznána RZ
- FN** – počet scénářů, ve kterých nebyla rozpoznána žádná RZ
- FP** – počet scénářů, ve kterých byla nalezena nesprávná RZ
- $a$  – počet pokusů, které byly provedeny napříč scénáři
- $t$  – celkový čas rozpoznávání během celého testu

Děkuji vám za pozornost

# 1. otázka z posudku oponenta

*Jak složité by bylo rozpoznávat více značek v jednom obrázku (např. při fotografii z parkoviště by bylo vhodné rozpoznat i přehledovou fotografii)?*

- rozpoznávání nyní vybere RZ, která je nejbližší středu obrázku
  - ostatní RZ zahodí
- bylo by třeba:
  - upravit rozhraní `VrpFinder` a jeho implementaci `VrpFinderImpl` tak, aby metoda `findVrpInImage()` vracela kolekci rozpoznaných RZ
  - náležitě upravit UI aplikace
  - dle další specifikace této funkcionality upravit databázi

## 2. otázka z posudku oponenta

*Jaké úpravy by bylo potřeba učinit, aby aplikace rozpoznávala zahraniční registrační značky?*

- rozpoznávání RZ je závislé na kontrole formátu identifikátoru RZ
- bylo by třeba:
  - implementovat rozhraní `VrpValidator`, které validuje rozpoznané textové bloky ve snímku (kontrola formátu RZ)
  - po implementaci tohoto rozhraní přidat jeho instanci do kolekce používaných validátorů ve třídě `VrpFinderImpl`
  - dle další specifikace této funkcionality upravit UI

### 3. otázka z posudku oponenta

*V textu práce je zmiňována multiplatformnost řešení. Jaké další kroky by bylo potřeba podniknout?*

- aplikace je připravená k sestavení na obě hlavní mobilní platformy **Android** a **iOS**
- aplikace byla důkladně otestována na široké škále zařízení s Android OS
- kvůli koronavirové situaci nebylo možné použít ve školní laboratoři zařízení s **macOS**, provést sestavení pro iOS a následně zde aplikaci otestovat
  - aplikace by však v aktuálním stavu měla být zcela funkční i na platformě iOS
  - bylo by třeba otestovat, zdali korektně funguje zpracování snímků ve formátu používaným na platformě iOS
  - ostatní funkční kód je identický pro obě platformy, a tudíž by nebylo třeba provádět další úpravy

## 4. otázka z posudku oponenta

*Provedl jste průzkum trhu, zda neexistují nějaké podobné aplikace pro Android?*

- ve službě **Google Play** mnoho aplikací určených k rozpoznávání RZ není
  - aplikace **Mobile LPR**, podporující RZ spoustu zemí včetně ČR
  - aplikace **Automatic License Plate Recognition Feature** podporující některé státy USA a Evropy (**ČR nepodporována**)
  - většina aplikací vyhledává informace o vozidle dle **ručně zadané RZ**
- komerční placená aplikace **Licence Plate Reader** od firmy **Z&G**
- další mobilní aplikace pro tuto platformu se mi nepodařilo najít