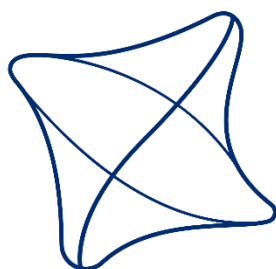


**Žilinská Univerzita v Žiline**

**Fakulta Riadenia a Informatiky**

Informatika



---

**ŽILINSKÁ UNIVERZITA  
V ŽILINE**

**Vývoj aplikácií pre mobilné zariadenia**

**Semestrálna práca**

Peter Hulák

5ZYI26

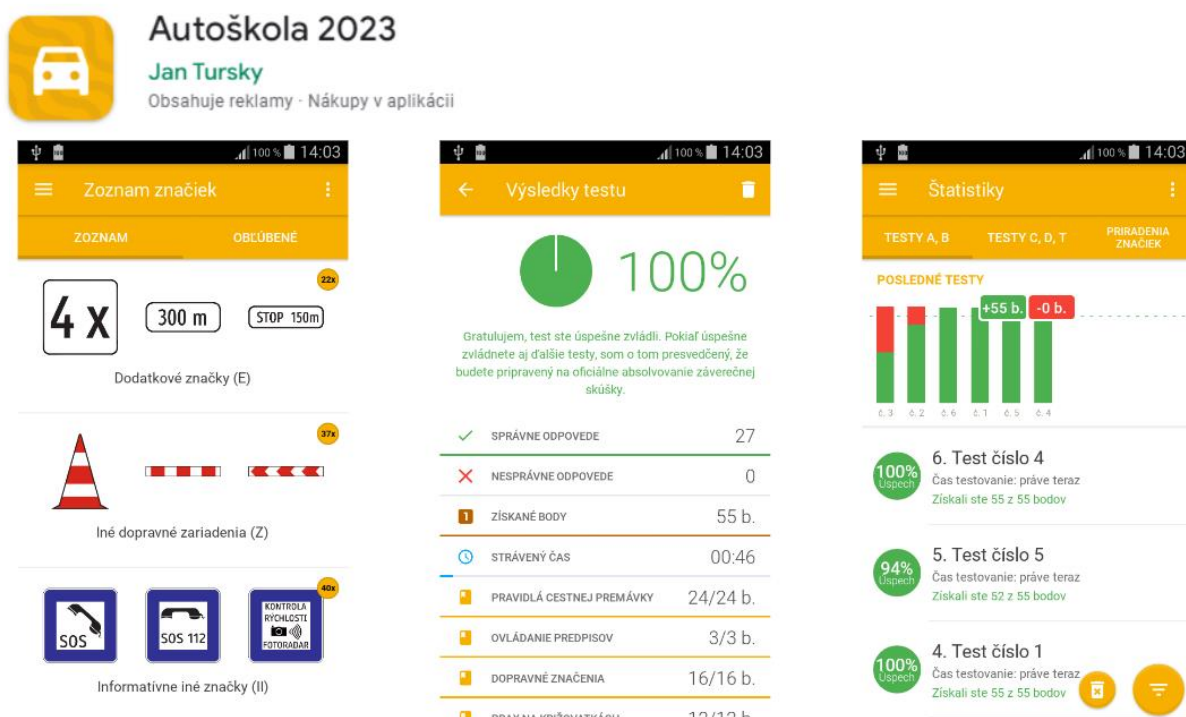
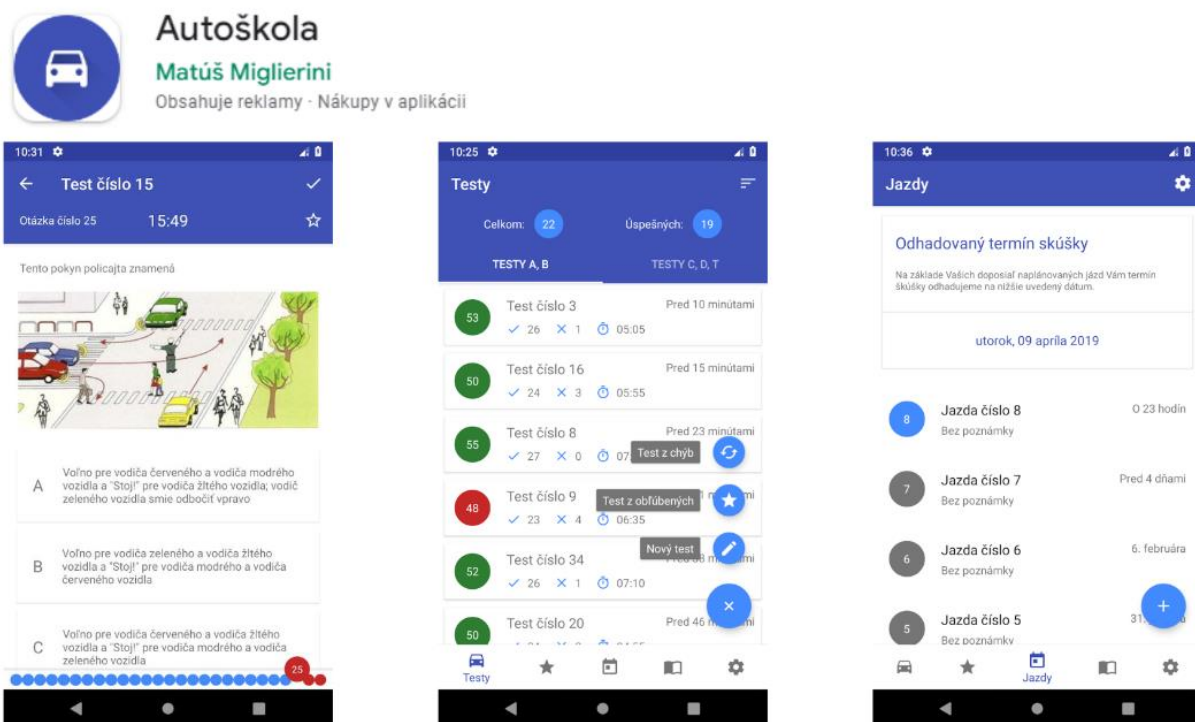
## Obsah

1.	Popis a analýza problému .....	3
2.	Podobné aplikácie .....	4
3.	Návrh aplikácie .....	5
4.	UML Diagram.....	6
5.	Popis Implementácie .....	7
6.	Použité implementácie .....	8
7.	GSON.....	9
8.	Príručka .....	10
9.	Záver .....	11
10.	Použité zdroje.....	12

## 1. Popis a analýza problému

Bude sa jednať o jednoduchú aplikáciu, ktorá bude určená najmä ale nie len pre študentov autoškoly. Bude obsahovať nový oficiálny náhodne generovaný test z autoškôl, ktorý má identický spôsob generovanie, umiestnenia, počtu otázok s tým oficiálnym. Na test bude mať užívateľ stanovený časový limit. Užívateľ si bude môcť vybrať buď oficiálny test alebo test obsahujúci otázky z určitej kategórie. Po skončení testu užívateľovi jeho test vyhodnotí a vypíše jeho výsledok, ktorý sa následne uloží do histórie.

## 2. Podobné aplikácie

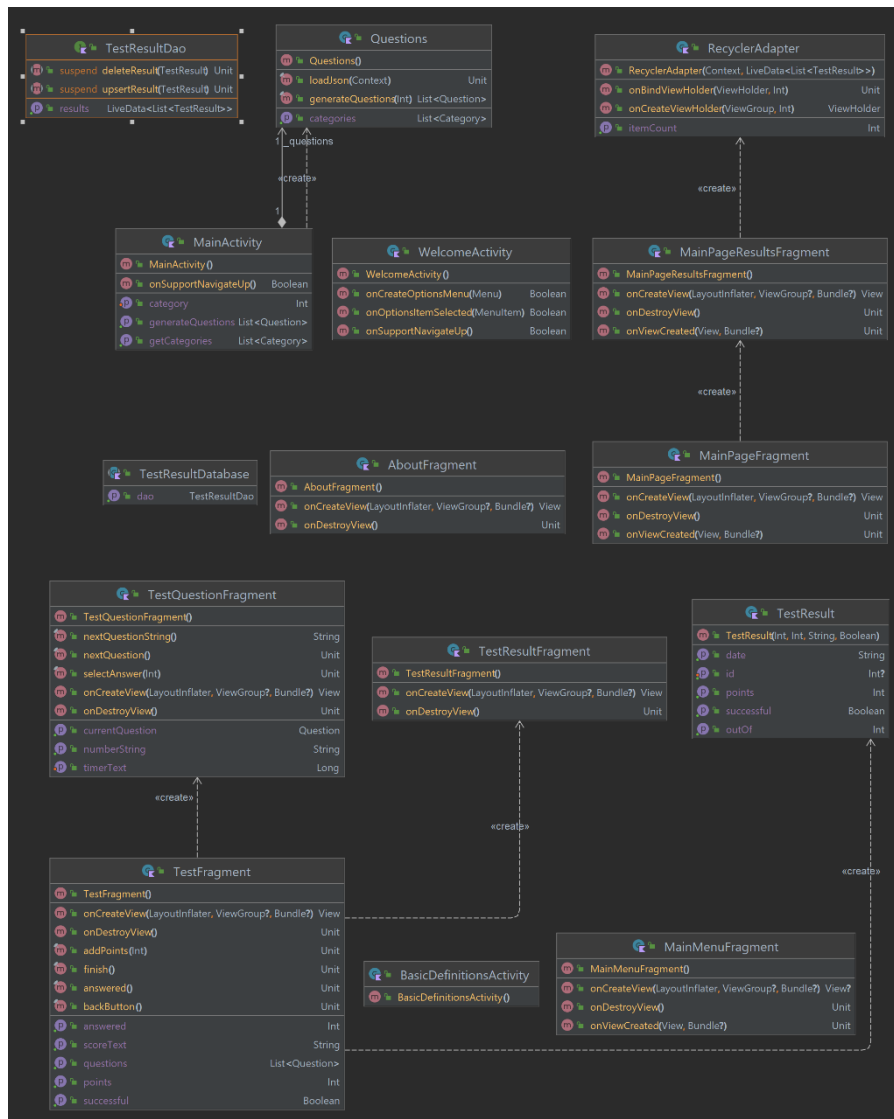


### 3. Návrh aplikácie

Trieda hlavnej aktivity je „WelcomeActivity“. Ďalšie aktivity sú „MainActivity“ a „BasicDefinitionsActivity“. „WelcomeActivity“ sa stará o prechody na ďalšie aktivity. „MainActivity“ obsahuje testy a históriu testov a „BasicDefinitionsActivity“ obsahuje základné poučky.

<b>WelcomeActivity</b> WelcomeActivity() <ul style="list-style-type: none"> <li>onOptionsItemSelected() Boolean</li> <li>onOptionsItemSelected() Boolean</li> <li>onSupportNavigateUp() Boolean</li> </ul>	<b>MainActivity</b> MainActivity() <ul style="list-style-type: none"> <li>onSupportNavigateUp() Boolean</li> <li>category Int</li> <li>generateQuestions() List&lt;Question&gt;</li> <li>getCategories() List&lt;Category&gt;</li> </ul>	<b>TestQuestionFragment</b> TestQuestionFragment() <ul style="list-style-type: none"> <li>nextQuestionString() String</li> <li>nextQuestion() Unit</li> <li>selectAnswer(int) Unit</li> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View</li> <li>onDestroyView() Unit</li> <li>currentQuestion Question</li> <li>numberString String</li> <li>timerText Long</li> </ul>	<b>TestResult</b> TestResult(int, int, String, Boolean) <ul style="list-style-type: none"> <li>date String</li> <li>id Int?</li> <li>points Int</li> <li>successful Boolean</li> <li>outOf Int</li> </ul>
<b>MainMenuFragment</b> MainMenuFragment() <ul style="list-style-type: none"> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View?</li> <li>onDestroyView() Unit</li> <li>onViewCreated(View, Bundle?) Unit</li> </ul>	<b>MainPageFragment</b> MainPageFragment() <ul style="list-style-type: none"> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View</li> <li>onDestroyView() Unit</li> <li>onViewCreated(View, Bundle?) Unit</li> </ul>	<b>TestResultFragment</b> TestResultFragment() <ul style="list-style-type: none"> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View</li> <li>onDestroyView() Unit</li> </ul>	<b>TestResultDao</b> suspend deleteResult(TestResult) Unit suspend insertResult(TestResult) Unit results LiveData<List<TestResult>>
<b>AboutFragment</b> AboutFragment() <ul style="list-style-type: none"> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View</li> <li>onDestroyView() Unit</li> </ul>	<b>MainPageResultsFragment</b> MainPageResultsFragment() <ul style="list-style-type: none"> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View</li> <li>onDestroyView() Unit</li> <li>onViewCreated(View, Bundle?) Unit</li> </ul>	<b>TestFragment</b> TestFragment() <ul style="list-style-type: none"> <li>onCreateView(LayoutInflater, ViewGroup?, Bundle?) View</li> <li>onDestroyView() Unit</li> <li>addPoints(int) Unit</li> <li>finish() Unit</li> <li>answered() Unit</li> <li>backButton() Unit</li> <li>answered Int</li> <li>scoreText String</li> <li>questions List&lt;Question&gt;</li> <li>points Int</li> <li>successful Boolean</li> </ul>	<b>Questions</b> Questions() <ul style="list-style-type: none"> <li>loadFrom(Context) Unit</li> <li>generateQuestion(int) List&lt;Question&gt;</li> <li>categories List&lt;Category&gt;</li> </ul>
<b>BasicDefinitionsActivity</b> BasicDefinitionsActivity()	<b>RecyclerViewAdapter</b> RecyclerViewAdapter(Context, LiveData<List<TestResult>>) <ul style="list-style-type: none"> <li>onBindViewHolder(ViewHolder, int) Unit</li> <li>onCreateViewHolder(ViewGroup, int) ViewHolder</li> <li>itemCount Int</li> </ul>		<b>TestResultDatabase</b> dao TestResultDao

## 4. UML Diagram



## 5. Popis Implementácie

WelcomeActivity prechádza buď na MainActivity alebo BasicDefinitionsActivity. MainActivity sa skladá zo začiatku z fragmentu, ktorý obsahuje RecyclerView, ktorý obsahuje zoznam všetkých predošlých nezmazaných výsledkov v LiveData uložených v TestResultDatabase (Room). Po stlačení „floating“ tlačidla napravo dole sa zobrazí Dialog, ktorý si vypýta druh testu. Následne sa vymení aktuálny fragment za TestFragment, ktorý sa skladá z TestQuestionFragmentu a TestResultFragmentu. TestQuestionFragment má Data bind na otázky a následne po poslednej otázke sa vymení TestQuestionFragment za TestResultFragment, ktorý obsahuje výsledok. Následne sa užívateľ môže vrátiť späť na MainPageFragment, ktorý obsahuje jeho nový výsledok, ktorý môže prípadne zmazať. Testy sú náhodne generované podľa oficiálnych testových pravidiel z otázok načítaných z JSONu pomocou GSONu.

## 6. Použité implementácie

- **Data Binding**
- **RecyclerView**
- **LiveData**
- **Navigation**
- **Room**
- **Knižnica GSON** – Spracovávanie JSONu, od Google
- ...



## 7. GSON

Implementácia čítania JSON-u

```
val inputStreamReader = InputStreamReader(context.assets.open( fileName: "data.json"))

val root = JsonParser().parse(inputStreamReader).asJsonObject

for (category in root.entrySet()) {
    val categoryTmp = category.value.asJsonObject
    val categoryObj = Category(categoryTmp.get("title").asString,
        categoryTmp.get("min").asInt,
        categoryTmp.get("max").asInt,
        categoryTmp.get("points").asInt)

    _categories.add(categoryObj)

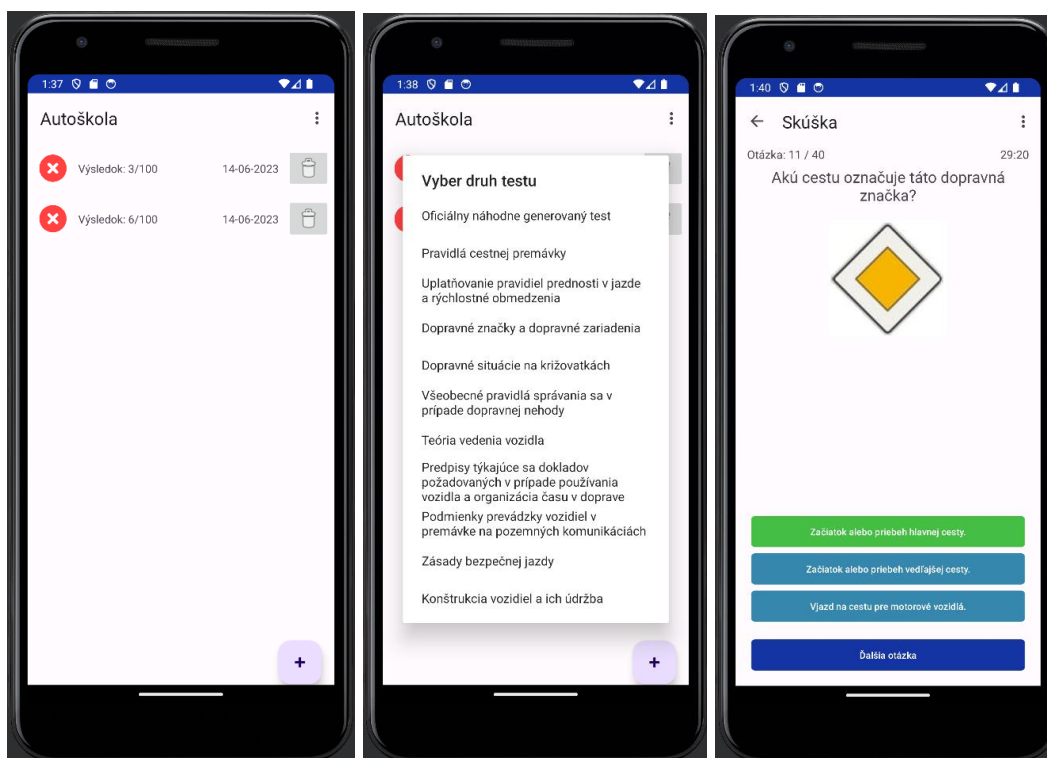
    for (question in categoryTmp.get("questions").asJsonObject.entrySet()) {
        val questionTmp = question.value.asJsonObject

        categoryObj.questions.add(
            Question(questionTmp.get("question").asString,
                questionTmp.get("right_answer").asInt,
                questionTmp.get("image").asString,
                questionTmp.get("answer1").asString,
                questionTmp.get("answer2").asString,
                questionTmp.get("answer3").asString,
                categoryObj)
        ))
    }
}

inputStreamReader.close()
```

## 8. Príručka

Po vstupe do aplikácie sa zobrazí hlavná stránka, na ktorej si môžeme zvoliť testy. Tu sa nachádza zoznam výsledkov predošlých testov. V pravom dolnom rohu sa nachádza tlačidlo so symbolom „+“, ktoré zobrazí ponuku na začatie nového testu. Tu máme na výber z niekoľko rôznych druhov testov. Najdôležitejší test je „Oficiálny náhodne generovaný test“, ktorý je identický test ako je na skúške z autoškoly, ostatné sú na precvičenie z určitej kategórie otázok. Po vybratí testu sa začína test, na ktorého vypracovanie má užívateľ 30 minút. Pri výbere odpovede sa užívateľovi okamžite zobrazí správna odpoveď a môže pokračovať na ďalšiu otázku. Po dokončení testu sa test vyhodnotí a uloží sa do histórie, následne sa užívateľ môže vrátiť na domovskú obrazovku. Hlavná obrazovka ešte v pravom hornom rohu obsahuje odkaz na informácie o aplikácii. Aplikácia taktiež obsahuje základné definície.



## **9. Záver**

Podarilo sa vytvoriť jednoduchú aplikáciu, ktorá umožňuje prípravu na skúšku v autoškole. Aplikácia sa ale ešte môže podstatne rozširovať, pridanie viac študijných materiálov, viac druhov testov a podobne.

## 10. Použité zdroje

- **Data.json a obrázky k testom** – Export z oficiálnych testov online skúšobných Minist. Vnútra
- **Základné definície** – Zákon č. 8/2009 Z. z.