**Fakulta riadenia a informatiky, Žilinská univerzita v Žiline**



**Vývoj aplikácií pre mobilné zariadenia**

semestrálna práca

 2020/2021

Ján Kováč

# Aplikácia pre servis výpočtovej techniky

## Spracovanie prehľadu dostupných aplikácií podobného zamerania.

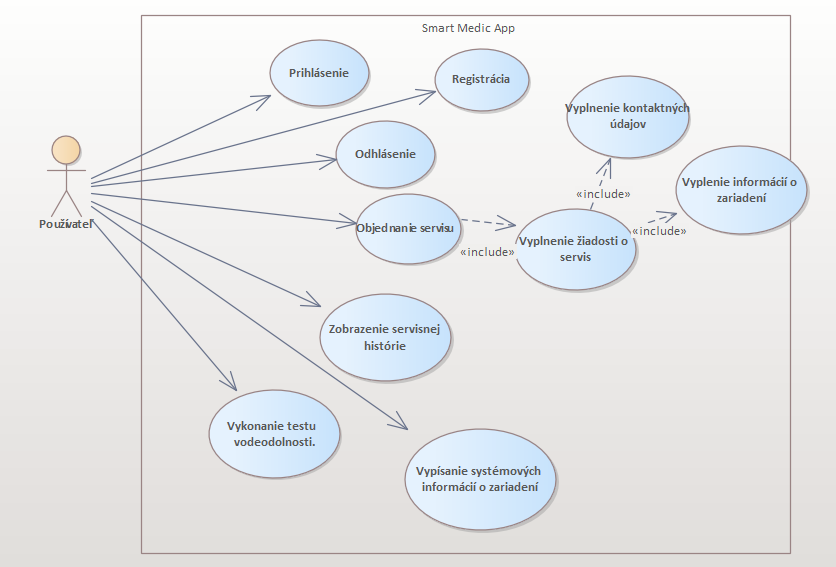
Bohužial pri hľadaní podobných aplikacíi sa mi nepodarilo nájsť žiadnu aplikáciu ktorá by bola podobná aplikácií na ktorej budem pracovať. Pri hľadaní som sa zameral na slovenské firmy ktoré sa venujú servisu výpočtovej techniky avšak ani jedna z nich pre svojich zákazníkov neponúka možnosť mobilnej aplikácie.

Pri hľadaní som sa zameral na firmy ako „Fix Servis“ , „Pc Express“, „Mobil Online“ avšak všetky menované firmy ponúkajú len webové rozhranie pre svojich zákazníkov.

# Po neúspechu som sa zameral aj na zahraničné firmy avšak dospel som k rovnakému záveru ako pri Slovenských firmách.

# Analýza navrhovanej aplikácie

## Use-case Diagram

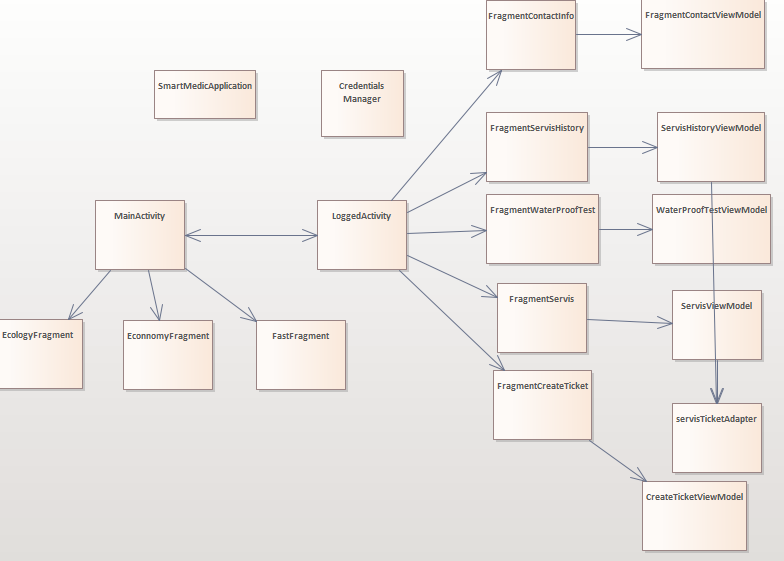


## Popis analýzy

Pri analýze aplikácie som musel brať ohľad na celkovú funkčnosť aplikácie tak aby fungujúcemu servisnému stredisku zjednodušila organizovanie jednotlivých objednávok a zákaziek. Rovnako by aplikácia mala zákazníkovi servisného strediska uľahčiť prehľad o aktuálne servisovaných zariadeniach a zlepšiť jeho informovanosť o procesoch pri spracovávaní jeho servisnej požiadavky.

Aplikácia bude zároveň prihlásenému používateľovi ponúkať možnosť vykonania testu odolnosti voči vode na jeho zariadení..

## UML diagram tried



## Popis Implementácie

Aplikácia obsahuje hlavnú aktivitu, aktivitu main. Po zapnutí aplikácie sa spustí táto aktivita a umožní používateľovy interakciu s 3 zobrazenými obrázkami, po kliknutí na tieto obrázky sa k nim zobrazia bližšie informácie. Ďaľej je používateľovy umožnené pokračovať do aplikácie pomocou tlačítka „Prihlásiť“. Po kliknutí na toto tlačítko sa používateľovy otvorí interface kde je mu umožnené sa prihlásiť pomocou účtu Google. Prihlásenie pomocou googlu prináša mnohé benefity ako napríklad rýchlosť prihlásenia bez nutnosti registrácie a zdĺhavého vytvárania účtu. Rovnako výhodné je poskytovať túto možnosť aj pre samotnú aplikáciu keďže vďaka google účtu vieme o používateľovy získať informácie ktoré môžeme neskôr použiť.

Celý proces prihlasovania je implementovaný pomocou frameworku „Auth0“. O uchovávanie Informácií o používateľov sa v aplikácií stará trieda CredentialsManager.

Po prihlásení sa užívateľovy zobrazí uvítacia hláška a pomocou menu može užívateľ pokračovať na ďaľšie funkcionality.

Povýbere položky servis sa mu zobrazí Fragment s recycler view ktorý je možné odfiltrovať pomocou tlačidiel „Mobil“ „PC“ „Console“ po stlačení tlačidla sa zobrazí vyfiltrovaný zoznam, výzor tlačidiel je implementovaný pomocou selektorov „Color“. V prípade že sa užívateľ rozhodne vytvoriť požiadavku na servis môže tak učiniť pomocou tlačidla „+“ po stlačení sa mu otvorí formulár kde je potrebné zadať informácie o zariadení, tieto informácie sa validujú a je potrebné ich zadať správne.

Ak sú všetky informácie zadané správne odblokuje sa tlačidlo „Pokračovať“ a užívateľ môže pokračovať na zadanie informácií ktoré budú slúžiť ako kontakt pre komunikáciu s majiteľom zariadenia.

Po stlačení tlačidla sa otvorí nový formulár v ktorom je možné nájsť predvyplnené údaje, v prípade že užívateľ potrebuje, môže tieto údaje zmeniť. Táto funkcia je implementovaná práve vďaka Auth0 a prihláseniu cez Google účet.

Zobrazený formulár je rovnako ako predošlý formulár potrebné vyplniť správne, kontroluje sa dĺžka slov, formát emailu a formát telefónneho čísla.

Prechod medzi formulármi je realiovaný pomocou SafeArgs argumentov medzi fragmentami. Po vyplnení prvého formuláru a stlačení tlačidla sa údaje o zariadení dočasne uložia do Room databázy. Pri prechode na daľší formulár sa do argumentov vloží Id týchto dočasných informácií. Pri dokončení ďalšieho formulára sa tieto informácie využijú na vytvorenie nového servisTicketu spolu s informáciami získanými v druhom formuláry a následne sa tento ticket vloží do Room databázy a užívateľ je presmerovaný naspať na servisFragment s využitím PopUptoInclusive.

List v položke je typu MutableLive Data a recycler view sa pri každej zmene znova aktualizuje.

Používateľ má ďalej možnosť v menu vybrať položku servisná história a zobraziť tak zariadenia ktorých servis už bol ukončený. Tu rovnako ako pri servise je využitý recyclerView a MutableLiveData. Pri výbere možnosti test vode odolnosti sa užívateľovi zobrazí fragment s vyrozumením že test je len orientačný a je na jeho vlastnú zodpovednosť.

Ak používateľ prijme riziko môže pokračovať ďalej na reálny test, tu všetko znova spravuje viewModel vrátane interakcíí s používateľom. Na logiku testovania sa používa sensor typu „PRESSURE“ a sensorManager. Na testovanie sa používateľ musí riadiť pokynmi aplikácie, v prípade že test prebehne v poriadku bude užívateľ upozornený na výsledok testu, ak je test neúspešný rovnko bude užívateľ notifikovaný o výsledku testu. Pre správne fungovanie a dostávanie informácí je potrebné zaregistrovať aktivitu ako listenera pre daný senzor, v prípade že by sa to tak nespravilo ModelNebude dostávať informácie a údaje zo senzora. Tu treba podotknút že listenera treba odregistrovať v životnom cykle onStop() inak by aplikácia zaťažovala systém na pozadí zbytočne.

V prípade že užívateľ vyberie možnosť odhlásiť bude odhlásený pomocou „Auth0“ a presmerovaný na úvodnú login stránku.

V aplikácii sa používajú 2 oddelené navigation graph aby som dosiahol požadované správanie aplikácie.

# Záver

Môžem skonštatovať že táto semestrálna bola pre mňa veľkým prínosom, pochopil som Vďaka nej základy fungovania Androidu a mobilných aplikácií ako takých. Na začiatku bol pre mňa veľkou neznámou jazyk Kotlin avšak vďaka tomuto kurzu už ovládam jeho základy. Pre mňa najzaujímavejšou témou bol jednoznačne Recycler view a Activity Lifecycles. Celý kurz hodnotím pozitívne a určite ho budem odporúčať.