



ŽILINSKÁ UNIVERZITA V ŽILINE

**Fakulta riadenia
a informatiky**

Fakulta riadenia a informatiky

Vývoj aplikácií pre mobilné zariadenia
Semestrálna práca

Jakub Smieško

Št. skupina: 5ZYR22

Šk rok: 2023/2024

Obsah

1. Prehľad podobných aplikácií	1
1.1 Recipes Haven.....	2
1.2 Tasty	3
2. Analýza navrhovanej Aplikácie	4
3. Architektúry aplikácie	5
4. Vzhľad aplikácie.....	6
5. UML diagram	7
6. Popis implementácie.....	8
7. GITHUB	10

1. Prehľad podobných aplikácií

V Apple Store a Obchode Play môžeme nájsť veľa rôznych aplikácií na varenie. Takéto aplikácie sú rôznorodé v tom, že sa môžu špecializovať na všetky druhy receptov od hlavných jedál po dezerty alebo na konkrétne vybrané druhy receptov (napríklad talianska kuchyňa, koláče).

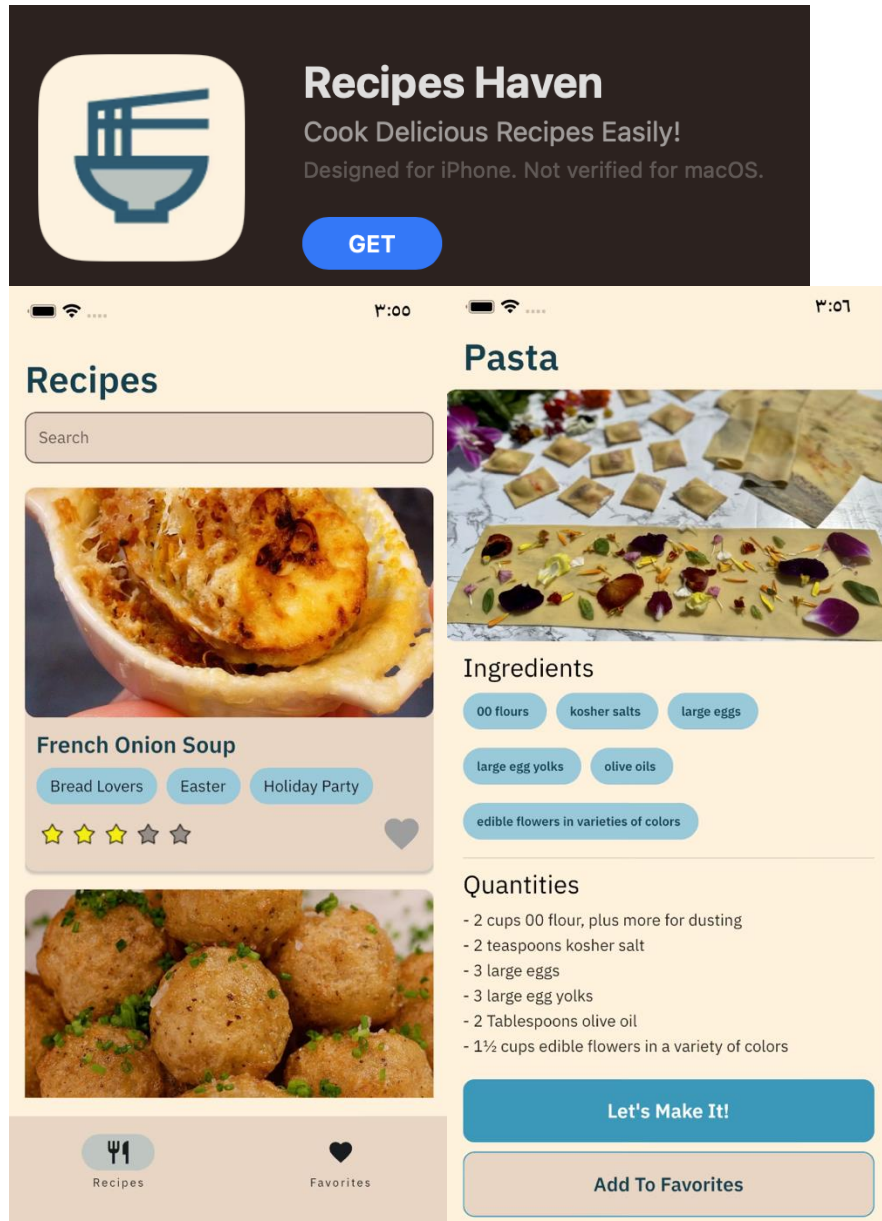
Hlavným účelom je ukázať užívateľovi potrebné ingrediencie, nástroje, postupy a prípadne aj videá na prípravu daného zvoleného receptu. V takýchto aplikáciách si užívateľ môže zvyčajne daný recept pridať medzi svoje obľúbené/neobľúbené, aby ho v budúcnosti nemusel znovu hľadať medzi všetkými ostatnými receptami, prípadne ho ohodnotiť hviezdikami od 1 do 5 podľa toho, ako sa mu daný recept páčil.

Takéto aplikácie sú určené pre domácich kuchárov alebo pre ľudí, ktorí skúšajú niečo nové.

Následne budeme analyzovať dve aplikácie: Recipes Haven a Tasty.

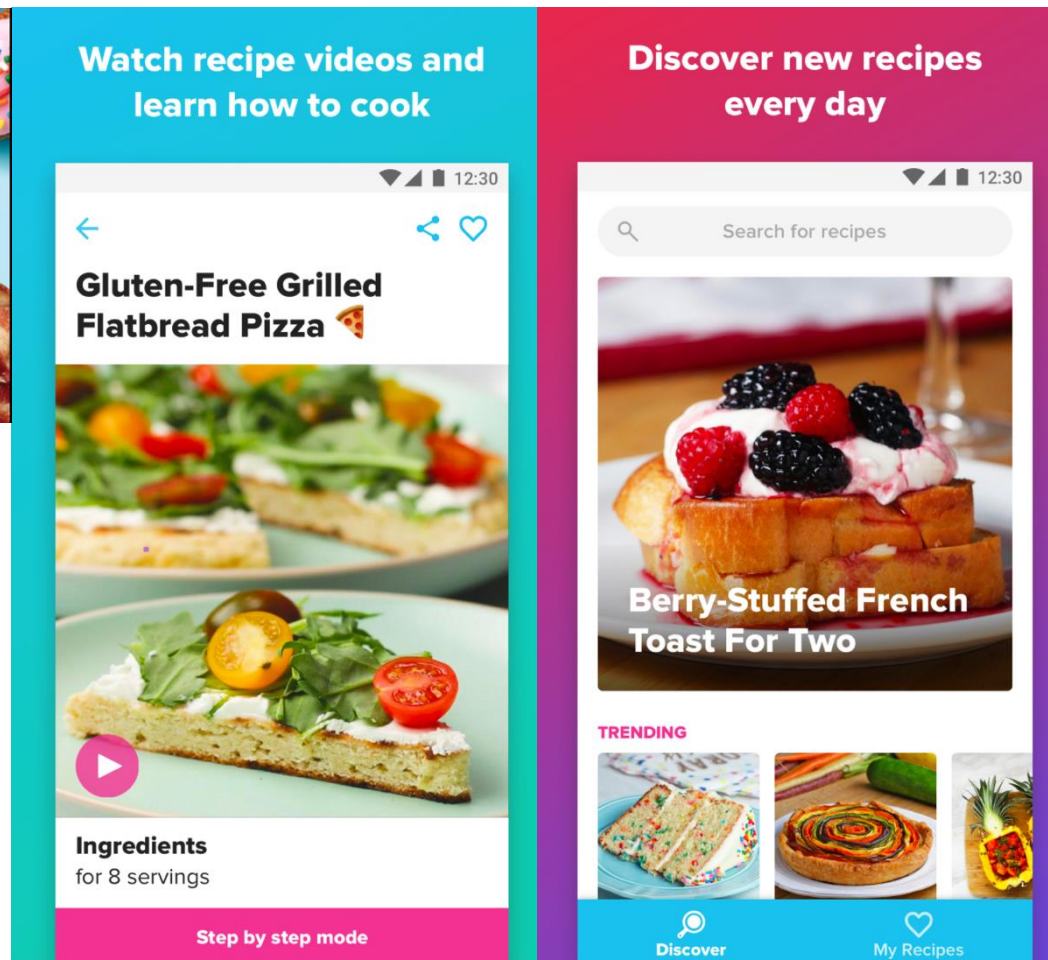
1.1 Recipes Haven

Recipes Haven je jednoduchá aplikácia na recepty, čo znamená, že aplikácia obsahuje okolo 30 receptov s ich postupmi na prípravu. V aplikácii si užívateľ môže ohodnotiť jednotlivé recepty hviezdikami od 1 do 5 a pridať si recept medzi svoje obľúbené.



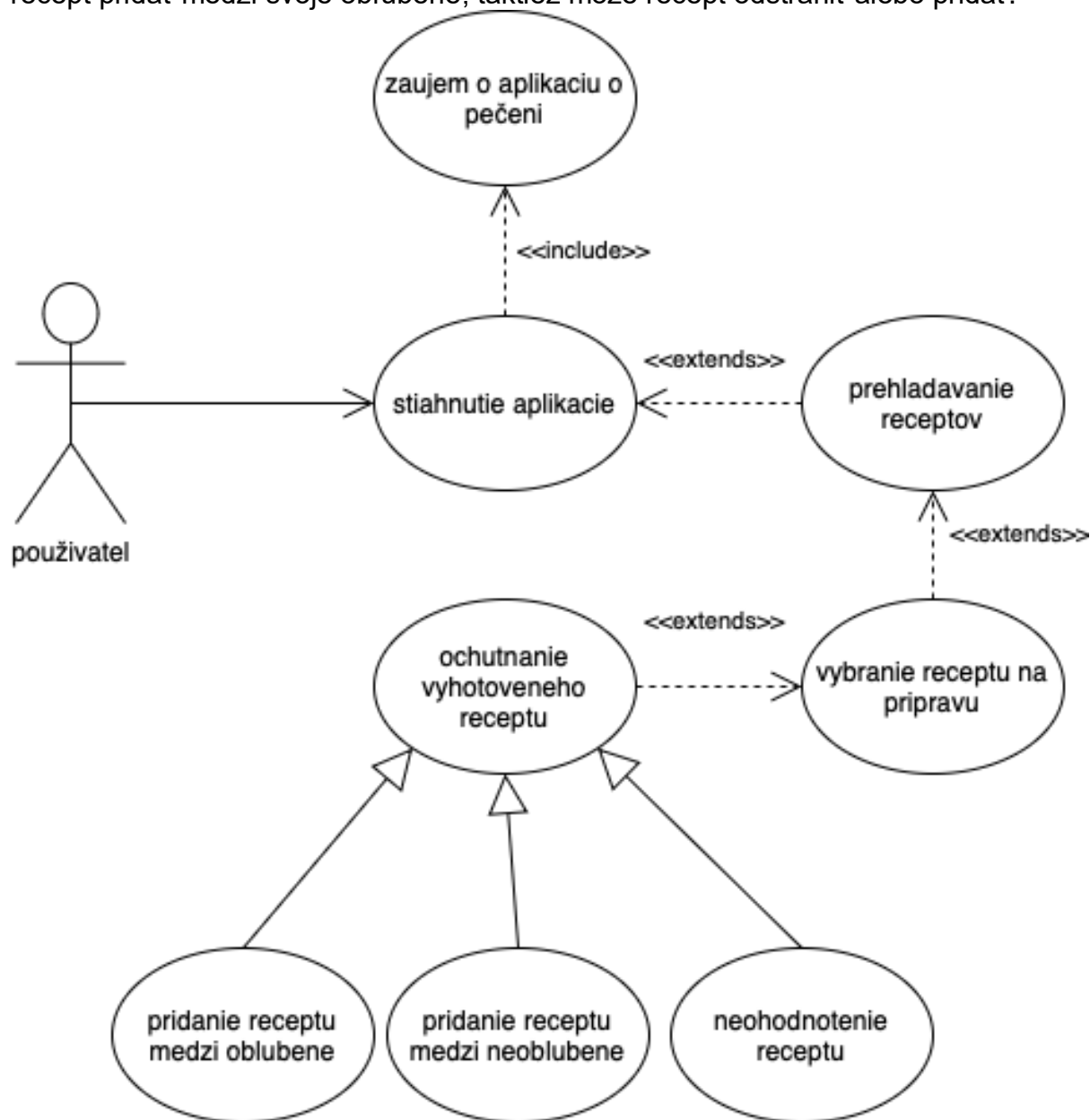
1.2 Tasty

Tasty je jednou z najpopulárnejších aplikácií, ktorá okrem svojich receptov a funkcií v predošlej aplikácii ponúka aj podrobné videá na prípravu jedál. Používateľ má na výber z mnohých receptov rôznych druhov, ktoré môže prechádzať aj v zvolených kategóriách. Tasty taktiež poskytuje funkciu prehľadávania najobľúbenejších jedál daného dňa. Okrem toho Tasty ponúka možnosť nakúpiť potrebné ingrediencie prostredníctvom aplikácie Walmart Grocery.



2. Analýza navrhovanej Aplikácie

Android aplikácia – Lets C00k - umožňuje používateľovi prechádzať receptami jedál ako aj potrebné ingrediencie a postup ich prípravy. Používateľ následne bude môcť recept pridať medzi svoje obľúbené, taktiež môže recept odstrániť alebo pridať.



3. Architektúry aplikácie

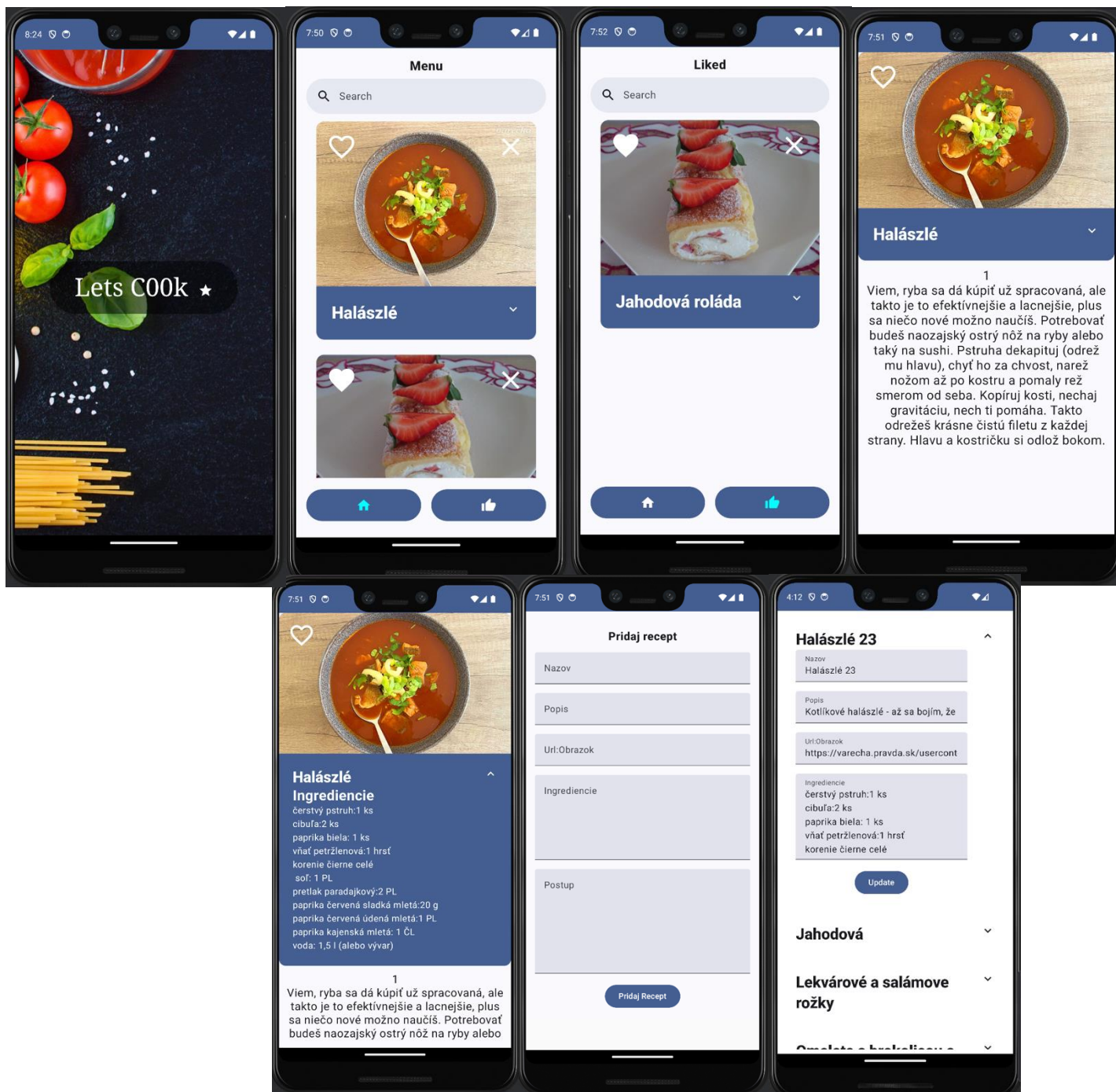
Aplikácia je postavená na architektonickom MVVM(Model-View ViewModel) vzore. MVVM je architektonický vzor, ktorý rozdeľuje aplikáciu do troch hlavných častí:

1. **Model:**
 - Zodpovedný za dáta a logiku aplikácie.
2. **View:**
 - Zobrazuje dáta užívateľovi.
3. **ViewModel:**
 - Spája Model a View, spracováva užívateľský vstup a aktualizuje View.

MVVM tiež využíva mechanizmy ako viazanie dát, príkazy a oznámenia na udržanie komponentov spojených a synchronizáciu dát v aplikácii.

4. Vzhľad aplikácie

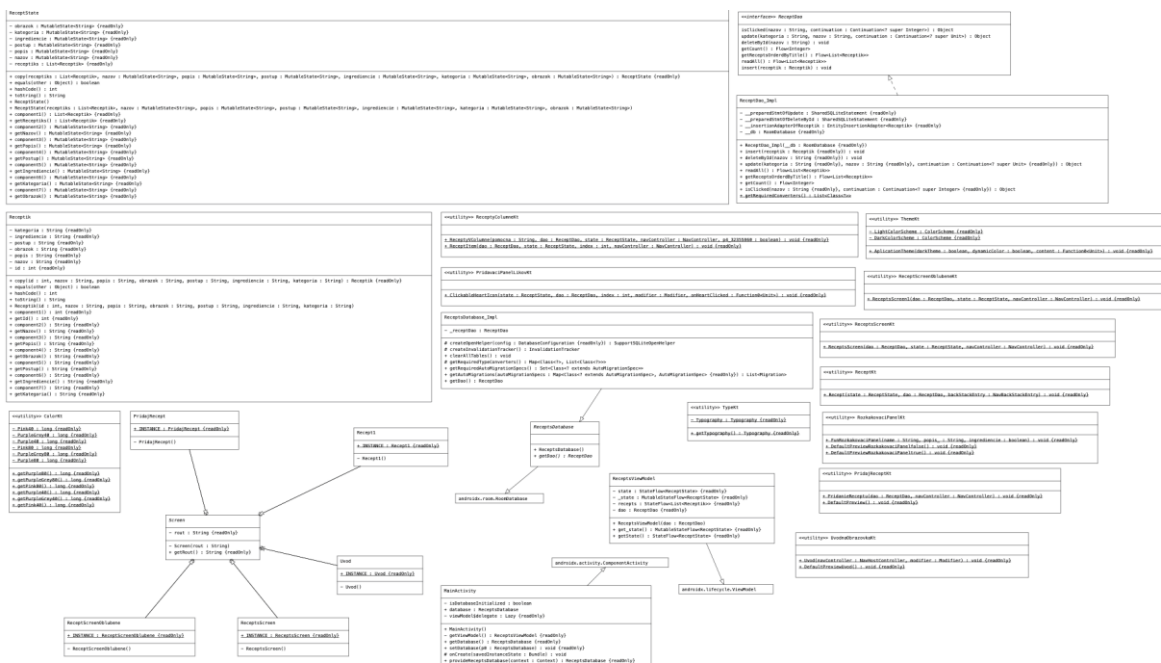
Po spustení aplikácie sa používateľ ocitne na uvítacej obrazovke, kde sa nachádza tlačidlo. Po jeho kliknutí sa používateľ dostane do hlavného menu, kde je možné kliknutím na text Menu pridať recept, taktiež tam sú zobrazované všetky recepty v scrollovacom stĺpci. Používateľ môže interagovať nasledovne: kliknutím na ikonu srdiečka (podľa stavu srdiečka prázdne/plné sa recept buď pridá medzi obľúbené, alebo odstráni), kliknutím na ikonu krížika, ktorého funkcia je vymazanie receptu (po kliknutí vyskočí okno, ktoré sa pýta používateľa, či chce naozaj odstrániť daný recept) a kliknutím na rozkazovací panel, kde sa zobrazí krátky popis k receptu. Na vrchu obrazovky sa nachádza vyhľadávací panel, vďaka ktorému si používateľ môže vyhľadávať konkrétne recepty podľa ich názvu. Na spodku obrazovky sa nachádzajú dve tlačidlá, s pomocou ktorých je možné preklikať sa medzi hlavným menu a menu obľúbených receptov, po kliknutí na text Liked nás presmeruje na okno editovania. Po kliknutí na konkrétny recept sa používateľ dostane na obrazovku zobrazujúcu daný recept, kde je vidieť obrázok receptu, swipovací panel, kde sú zobrazované očíslované postupy receptu a rozkazovací panel, po ktorom rozkliknutí sa zobrazia ingrediencie.



5. UML diagram

Nasledujúci UML diagram je automaticky vygenerovaný pomocou Android štúdia pluginu UML Generator.

Pre lepšiu viditeľnosť pridávam do repozitára .jpg uml diagramu



Krátky popis zopár tried z uml Diagramu:

ReceptsDatabase - definuje ROOM databázu.

Receptik - Entitata ROOM databázy(Recept)

ReceptState - je dátová trieda, ktorá predstavuje stav obrazovky s receptami.

ReceptsViewModel - Manažovanie UI-dáta a logiku pre obrazovky, aktualizuje po tom ako sú dáta zmene, Interakcia s databázou, manažuje list receptov z databázy.

ReceptDao - tento interface definuje operácie pre prácu s databázou receptov.

Screen - definuje rôzne obrazovky (screeny) v aplikácii pomocou ich ciest (routes). Používa sa na navigáciu medzi rôznymi obrazovkami. Každá vnorená objektová trieda reprezentuje jednu obrazovku s unikátnou cestou.

6. Popis implementácie

Obrazovky:

V aplikácii sa nachádza 5 obrazoviek:

-Úvodná obrazovka(UvodnaObrazovka) - obsahuje tlačidlo, ktoré po kliknutí nás preniesie do Menu.

-Menu(ReceptsScreen) - Na obrazovke sa nachádza názov obrazovky(Menu), searchbar, skrolovaci column, v ktorom sú jednotlivé recepty, ktoré môžeme následne po kliknutí príslušnej ikonky liknut alebo odstrániť a 2 tlačidla(navigácia medzi obľúbenými a menu).

-Menu obľúbené (ReceptScreenOblubene) - Iste gui a funkcie ako na menu až na to, že v skrolovacom columnne sa nachádzajú len obľúbené recepty.

-Recept(Recept) - Na obrazovke sa nachádza obrázok receptu s príslušnými funkčnými ikonkami ako sú na receptoch v menu/menu obľúbené, ingrediencie + postup receptu.

-Pridanie Receptu(PridanieReceptu)- Tu je možné pridať recept do databázy(prístup k tejto obrazovke je po kliknutí názvu na obrazovke Menu.

-Editovanie(Editovanie)- tu je možné editovať ľubovoľný zvolený recept.

Navigácia:

Pre navigáciu medzi rôznymi obrazovkami v aplikácii som implementoval navigáciu pomocou Jetpack Compose Navigation Component. Navigator je inicializovaný v MainActivity pomocou rememberNavController(), ktorý poskytuje centrálnu kontrolu pre navigáciu v celej aplikácii. Jednotlivé obrazovky sú definované ako composable funkcie v rámci NavHost, ktorý je súčasťou základného zloženia aktivity.

ROOM Databáza:

Na prácu s receptami používam ROOM databázu, ktorá sa po prvej otvorenej aplikácie(iba raz) neinicializuje zo json súbora, následne je možné s databázou pracovať v štýle odstránenia alebo pridania receptov !(na načítanie databázy som čiastočne použil YouTube tutoriál, trebalo príslušný kód prerobiť na moju databázu/entity)!

Lifecycle:

V triede ReceptsViewModel využívam životný cyklus pre synchronizáciu operácií s dátami s cyklom života ViewModel. Týmto spôsobom zabezpečujem správnu správu dát a ich aktualizáciu po zmene, ako aj interakciu s databázou pre manažovanie zoznamu receptov.

Libraries:

Jetpack Libraries:

- androidx.core:core-ktx:1.9.0
- androidx.lifecycle:lifecycle-runtime-ktx:2.6.2
- androidx.activity:activity-compose:1.8.0
- androidx.compose.ui:ui
- androidx.compose.ui:ui-graphics
- androidx.compose.ui:ui-tooling-preview
- androidx.compose.material3:material3-android:1.2.1
- androidx.paging:paging-common-android:3.3.0-beta01
- androidx.datastore:datastore-core-android:1.1.0
- androidx.room:room-ktx:2.6.0
- androidx.navigation:navigation-compose:2.7.5
- androidx.compose.material:material-icons-extended:1.5.4
- androidx.lifecycle:lifecycle-viewmodel-compose:2.4.0-alpha05

Kotlin Libraries:

- org.jetbrains.kotlinx:kotlinx-coroutines-core:1.5.2

Dependency Injection:

- com.google.dagger:hilt-android:2.40.5
- androidx.hilt:hilt-lifecycle-viewmodel:1.0.0-alpha03
- androidx.hilt:hilt-navigation-compose:1.0.0
- javax.inject:javax.inject:1

Image Loading:

- com.squareup.picasso:picasso:2.71828
- io.coil-kt:coil-compose:2.4.0

Jpgs from urls:

Obrázky, ktoré sú využité v aplikácií sú načítane z internetu pomocou url obrázka.

Dark Mode:

Aplikácia sa prispôbuje darkmodu na mobilnom zariadení

7. GITHUB

Práca s githubom:

Počas vypracovania svojej semestrálnej práce som sa stretol s niekoľkými problémami. Napríklad sa mi nedarilo implementovať KAPT do Gradle, čo je potrebné pre Room databázu. Vyriešil som to tým, že som stiahol projekt z internetu, ktorý už obsahoval túto implementáciu, a potom som do tohto stiahnutého projektu prekopíroval svoj vlastný kód. Nepoužil som však nič zo stiahnutého projektu, iba som využil jeho konfiguráciu. Okrem toho som sa stretol aj s problémom pri pushovaní do druhého repozitára na Gite. Kvôli týmto problémom som tak nakoniec skončil s tromi repozitármi: 1. Pôvodný, na začiatku vytvorený, 2. Nový, ktorý som stiahol z internetu, a 3. Repozitár, ktorým som vyriešil problémy s pushovaním.

Linky na githubom:

1.
<https://github.com/007Guineapig/AndroidSemkaAliqatorNaMasle.git>
2.
<https://github.com/007Guineapig/AndroidSemkaNaMasle1.git>
3.
<https://github.com/007Guineapig/dalsi.git>