ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Semestrálna práca z predmetu VAMZ

MANGA APLIKÁCIA

Vypracoval: Michal Zúbek
Študijná skupina: 5ZYI25
Cvičiaci: doc. Ing. Patrik Hrkút, PhD.
Termín cvičenia: streda bloky 1-2

v Žiline dňa 7.4.2024

Obsah

1. Popis a motivácia

Cieľom tejto semestrálnej práce je vyriešiť problém, ktorý trápi ľudstvo už desaťročie a to jednoduchý prístup k japonským komiksom (naďalej manga). Na trhu existuje niekoľko aplikácií, ktoré sa tento problém pokúšali vyriešiť avšak nedosiahli toho úplne. Moja aplikácia sa bude snažiť o kombináciu a vylepšenie týchto aplikácií pre vytvorenie jednotného zdroja na mangu. Jeden z hlavných dôvodov pre výber tejto práce je koniec vývoja aplikácie Tachiyomi, jedného z lídrov v tomto obore. Avšak týmto sa na trhu vytvorila diera a z telefónu mi zmizli megabajty, ktoré sa pokúsim naplniť a nakoniec zlepšiť mojou aplikáciou.

2. Prieskum trhu

Dve aplikácie, na ktoré sa budeme pozerať budú Mangaplus a Tachiyomi:

2.1 Mangaplus

Mangaplus je jediná manga aplikácia, ktoré sa nachádza aj na google playstore z dôvodu, že je vlastnená vydavateľstvom Shueisha Inc., čo je najväčšia vydavateľská spoločnosť v Japonsku.

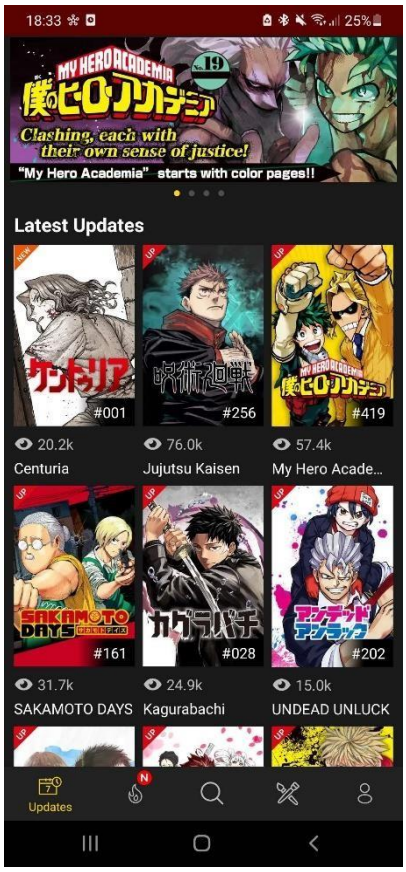
Aj napriek tomu, že mangaplus má za sebou takúto mimoriadnu spoločnosť, ich aplikácia zanecháva veľa priestoru na vylepšenie. Mangaplus disponuje týmito funkciami:

- vyhľadávanie podľa názvu
- limitované vyhľadávanie podľa žánru
- triedenie podľa jazyka
- trendujúce a populárne mangy
- obľúbené
- naposledy aktualizované

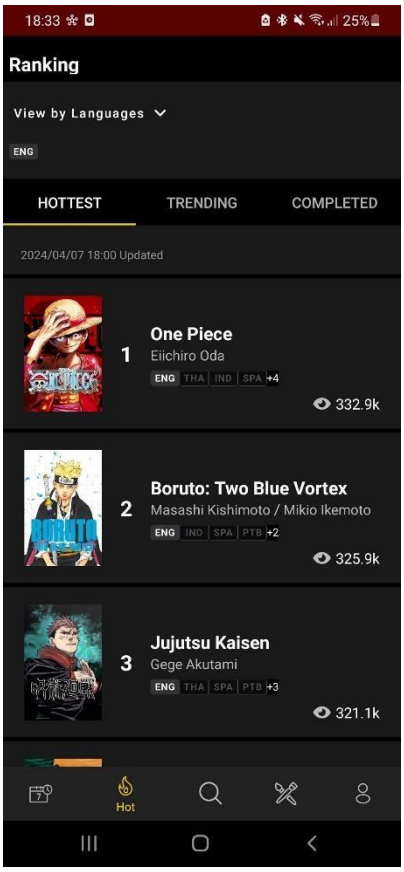
Free verzia aplikácie dovoľuje iba čítanie najnovšej kapitoly a pre prístup k predošlým si používateľ musí zaplatiť mesačný poplatok.

V nižšie uvedených obrázkoch je možné vidieť ako Mangaplus vyzerá.

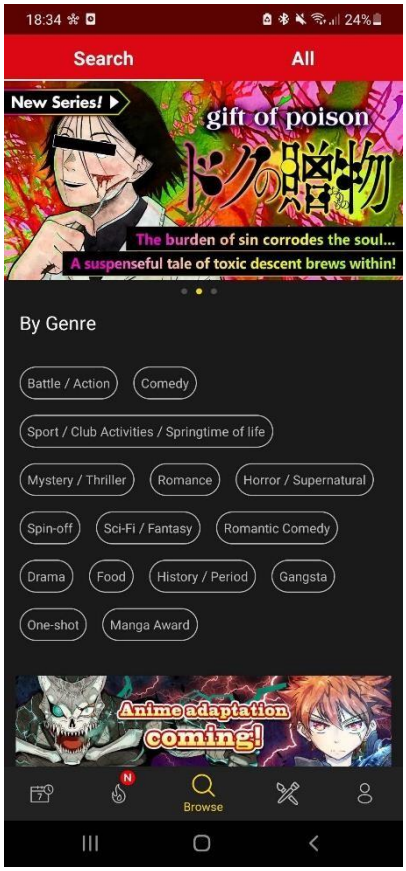
Mangaplus teda obsahuje postačujúci základ pre takúto aplikáciu ale pre náročnejších používateľov má ešte čo dodať.



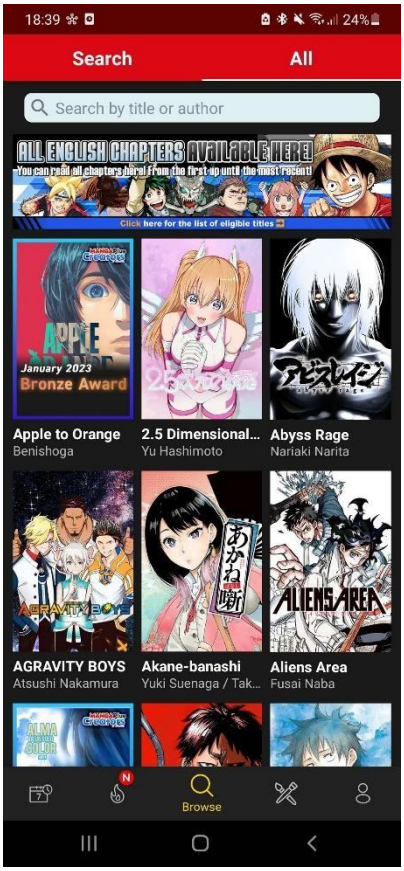
1.1 Naposledy aktualizované



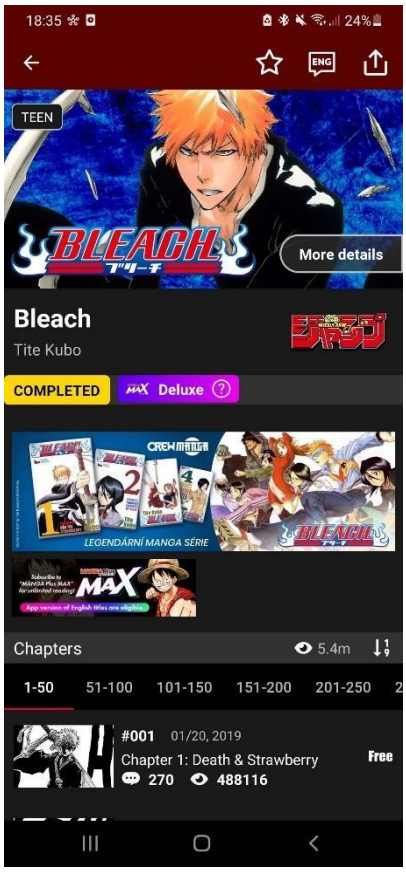
1.2 Trendy



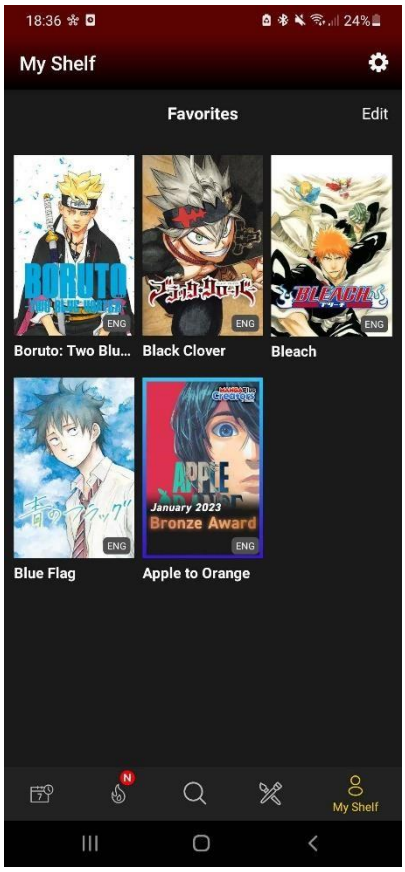
1.3 Vyhľadávanie podľa žánru



1.4 Vyhľadávanie podľa názvu

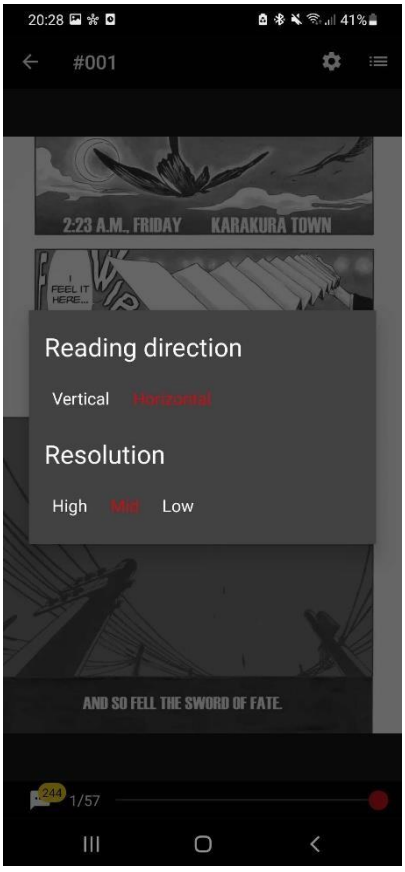


1.5 Prehľad kapitol



1.6 Obľúbené

1.7 Prehliadač strán mangy



1.8 Nastavenia prehliadača

2.2 Tachiyomi

Tachiyomi, teraz už nepodporovaná, je jedna z najpopulárnejších aplikácií pre čitateľov mangy pretože sa vyznačuje nielen vlastnosťami jej predchodcov ako Mangaplus ale taktiež pridáva nové funkcie ako napríklad:

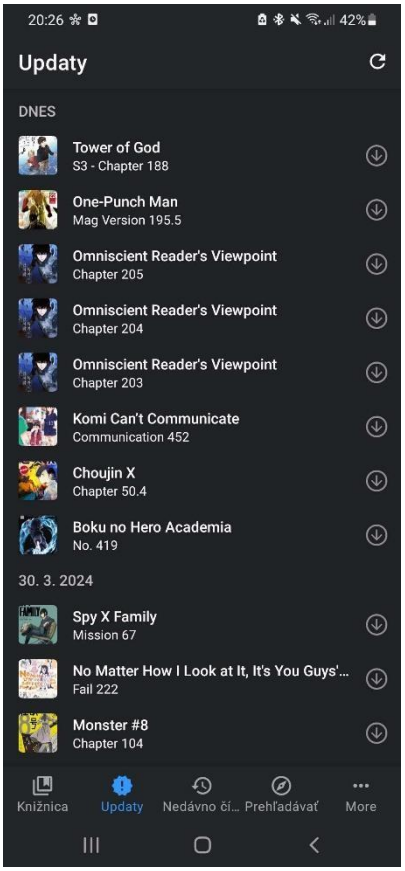
- veľký počet zdrojov z kadiaľ mangu čerpá
- možnosť si stiahnuť kapitoly na offline čítanie
- rozšírené nastavenia čo sa týka čítania
- lepšie vyhľadávanie
- nedávno čítané

Tachiyomi avšak zaostáva pri doporučení obsahu a ľahkosti navigácií cez používateľské rozhranie, ktoré občas môže byť mätúce.

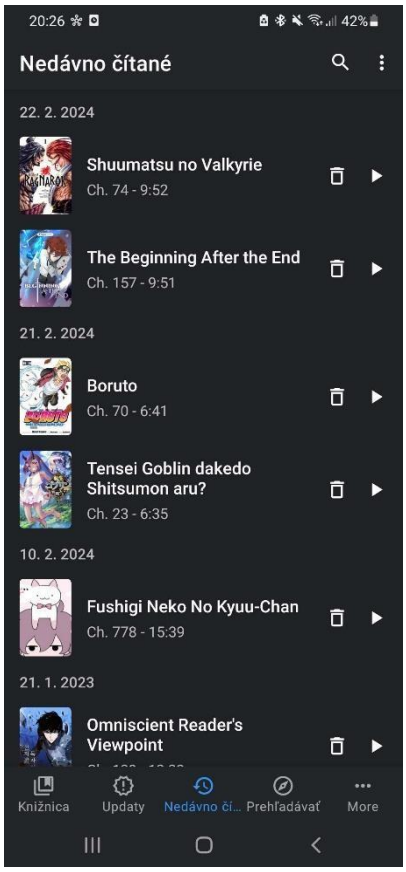
V nižšie uvedených obrázkoch je možné vidieť ako Tachiyomi vyzerá.

Tachiyomi je teda naozaj dobrá aplikácia ale taktiež má svoje nedostatky.

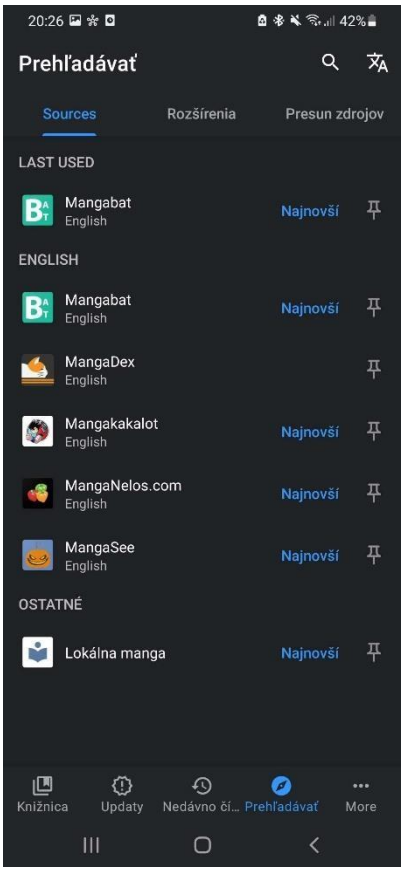
2.1 Knižnica/Obľúbené



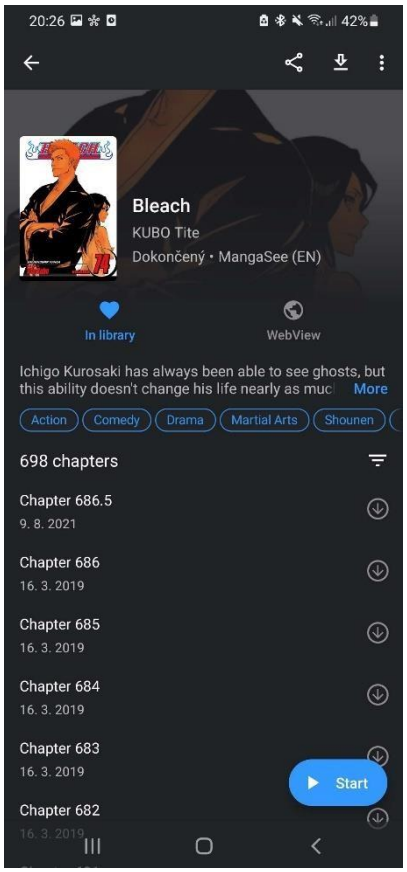
2.2 Nedávno aktualizované



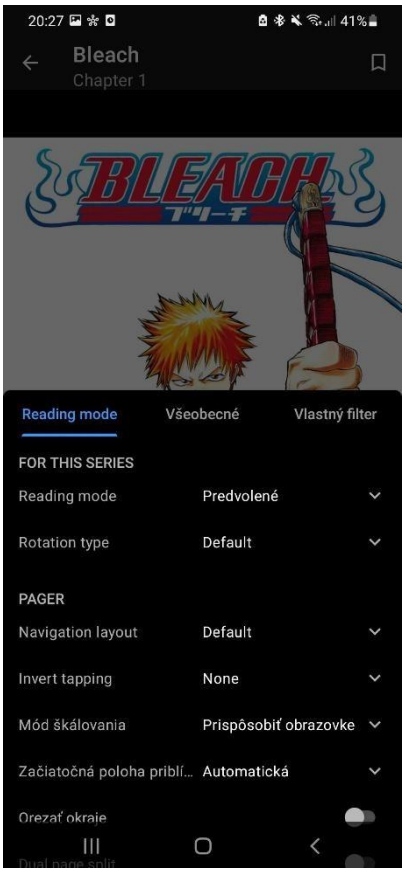
2.3 Nedávno čítané



2.4 Zdroje mangy

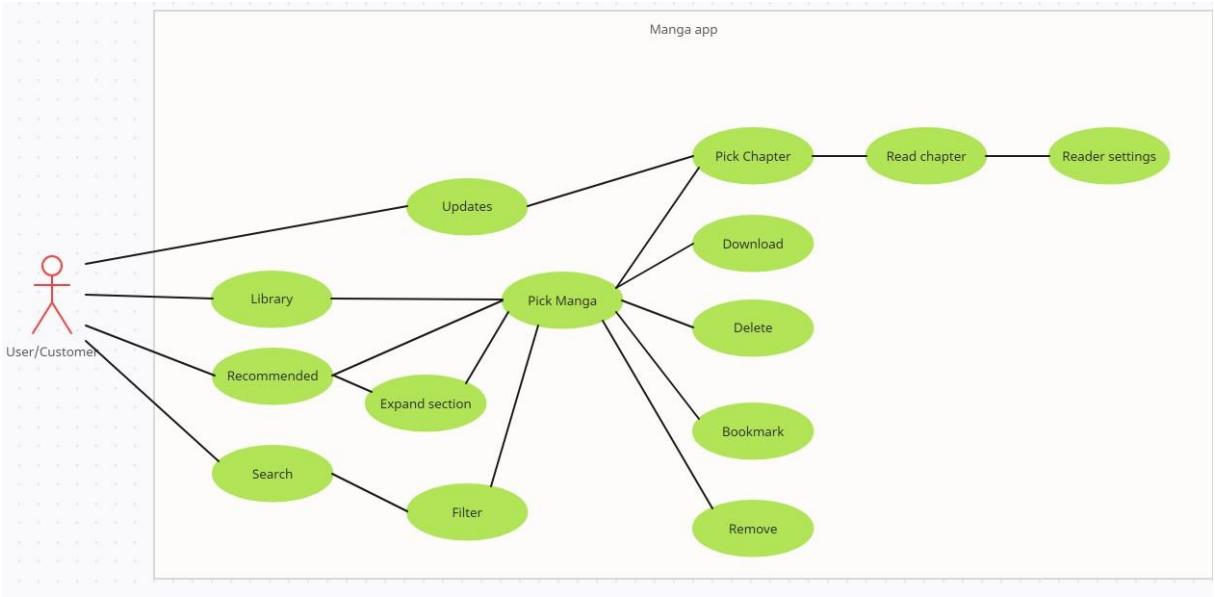


2.5 Prehliadač kapitol



2.6 Nastavenia prehliadača

3. Analýza navrhovanej aplikácie



3. Use case model

Vyššie uvedený obrázok je Use case model používateľa mojej aplikácie. Používateľ po spustení aplikácie ma niekoľko možností podľa toho na čo má práve chuť. Môže si skontrolovať novo aktualizované mangy, pozrieť svoju knižnicu obľúbených, nechať si niečo odporučiť alebo sám si vyhľadať čo vyžaduje. Vždy však nakoniec sa dostane k výbere určitej mangy kde si ju môže pridať/odstrániť do/z knižnice, stiahnuť si kapitoly alebo začať čítať, kde sa nakoniec ešte môže pohrať s nastaveniami prehliadača.

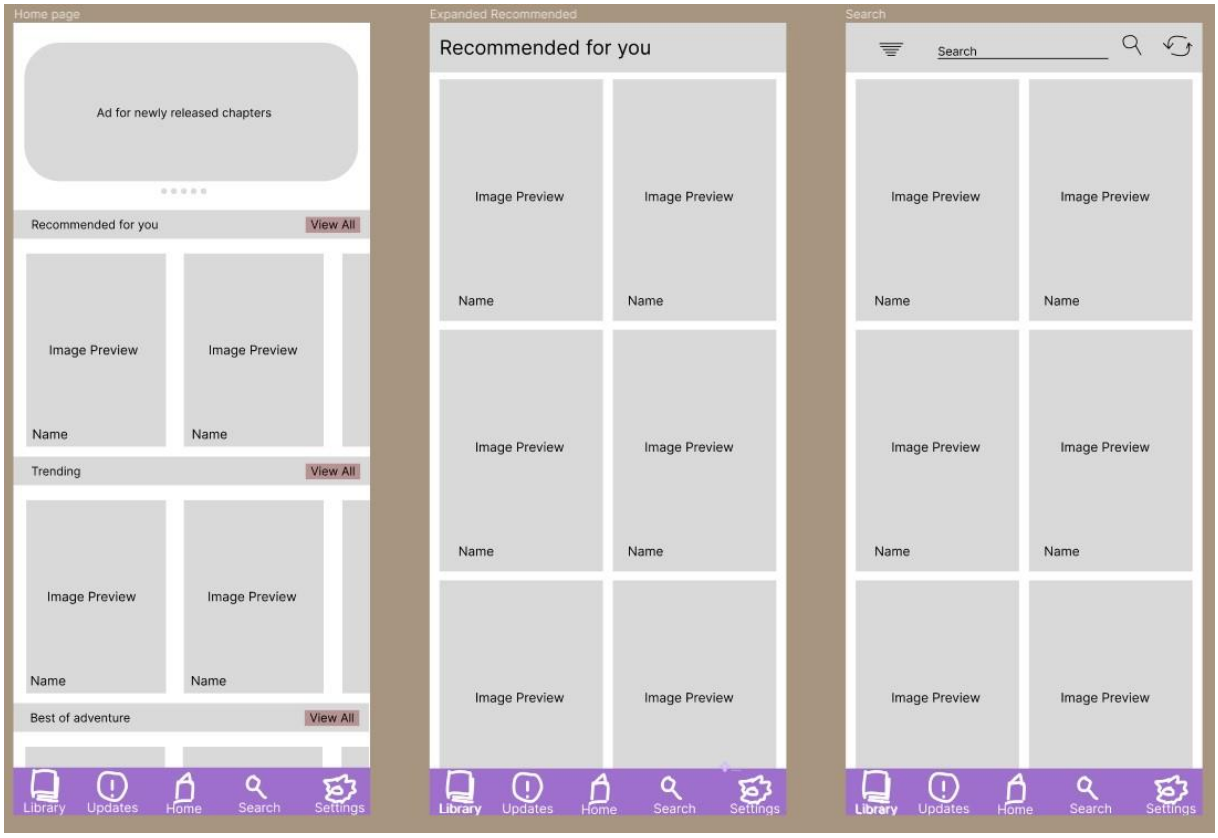
4. Návrh architektúry aplikácie

Moja aplikácia bude teda kombináciou a vylepšením týchto dvoch aplikácií. Poďme sa teda pozrieť na jej funkčnosť a vlastnosti:

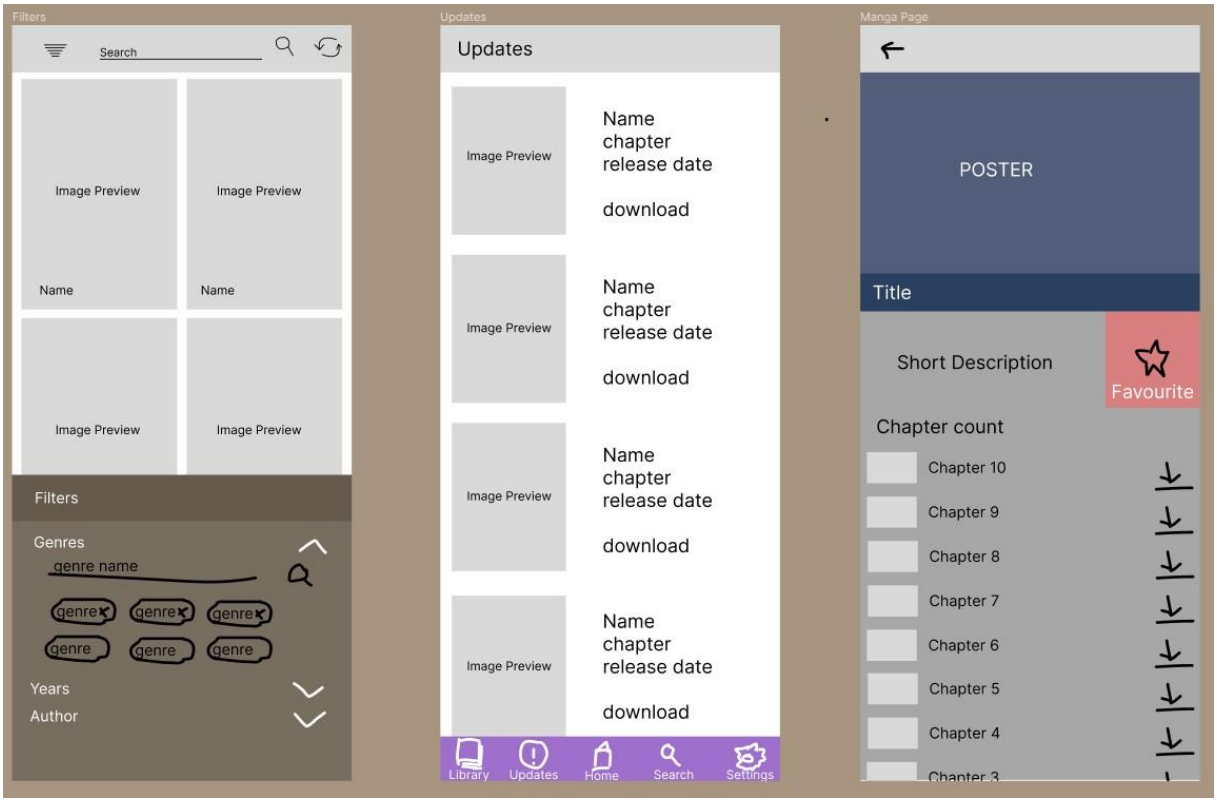
- Predná strana pre odporučenia a trendy
- Vyhľadávanie podľa názvu
- Vyhľadávanie podľa žánru a iných vlastností
- Rôzne zdroje z ktorých je možné čerpať mangu
- Nedávno aktualizované kapitoly
- Knižnica a priradenie do knižnice
- Naposledy čítané
- Oznámenia o pridaní novej kapitoly
- Sťahovanie jednotlivých kapitol
- Zobrazovanie prečítaných a neprečítaných
- Základné nastavenia
- Nastavenia Prehliadača ako reading mode a image scaling

- Podpora pre webtoons/Manwha

5. Wireframe model



4.1 Home page *4.2 Recommended expanded* *4.3 Search*



4.4 Filters *4.5 Updates* *4.6 Manga page*

6. Popis implementácie

Pre tvorbu aplikácie boli využité.

AndroidX komponenty:

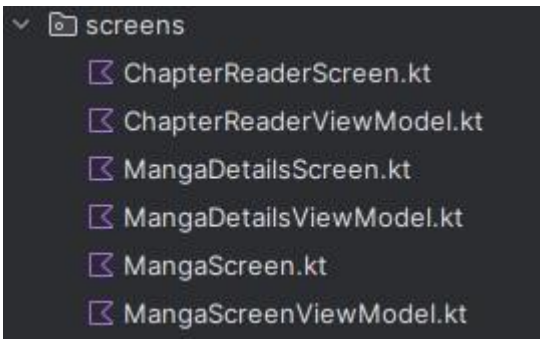
- LifeCycles – manažovanie stavov UI

```
var mangaUiState: MangaUiState by mutableStateOf(MangaUiState.Loading)
    private set

Michal Zúbek
var mangaSearchState: MangaSearchState by mutableStateOf(MangaSearchState( isSearching: false,  title: "",  offset: 0,  total: 0))
    private set
```

- Navigation – pohyb medzi obrazovkami

```
fun MangaApp(
    navController: NavHostController = rememberNavController()
) {
    val mangaScreenViewModel: MangaScreenViewModel =
        viewModel(factory = MangaScreenViewModel.Factory)
    val mangaDetailsViewModel: MangaDetailsViewModel =
        viewModel(factory = MangaDetailsViewModel.Factory)
    val chapterReaderViewModel: ChapterReaderViewModel =
        viewModel(factory = ChapterReaderViewModel.Factory)

    NavHost(
        navController = navController,
        startDestination = MangaAppScreens.HomeScreen.name,
        modifier = Modifier
    ) { this: NavGraphBuilder
        composable(route = MangaAppScreens.HomeScreen.name) { this: AnimatedContentScope  it: NavBackStackEntry
            MangaScreen(
                viewModel = mangaScreenViewModel,
                modifier = Modifier,
                onMangaClick = { manga ->
                    mangaDetailsViewModel.loadMangaDetails(manga)
                    navController.navigate(MangaAppScreens.MangaDetailsScreen.name)
                },
            )
        }
```

- ViewModel – rozdelenie logiky od UI, použitie ViewModel Factory



```
class MangaDetailsViewModel(
    private val mangaDexRepo: MangaDexRepo,
) : ViewModel() {

    // Holds the current state of the MangaDetails UI.
    Michal Zúbek
    var mangaDetailUiState: MangaDetailUiState by mutableStateOf(MangaDetailUiState.Loading)
        private set

    Michal Zúbek
    init {
        mangaDetailUiState = MangaDetailUiState.Loading
    }
```

Externé knižnice:

- Retrofit2 – komunikácia s externým api
- Kotlinx serilazation – serializácia json súborov do objektového typu

```
class DefaultAppContainer : AppContainer {
    private val baseUrl =
        "https://api.mangadex.org"

    private val json = Json { this: JsonBuilder
        ignoreUnknownKeys = true
        coerceInputValues = true
    }

    private val retrofit = Retrofit.Builder() Retrofit.Builder
        .addConverterFactory(json.asConverterFactory("application/json".toMediaType()))
        .baseUrl(baseUrl)
        .build()

    private val retrofitService: MangaDexApiService by lazy {
        retrofit.create(MangaDexApiService::class.java)
    }
    override val mangaDexRepo: MangaDexRepo by lazy {
        NetworkMangaDexRepo(retrofitService)
    }
}
```

- Coil – pre asynchrónné načítavanie obrázkov z internetu

```
val coverLink = "https://uploads.mangadex.org/covers/${manga.id}/${coverName}.256.jpg"

AsyncImage(
    model = ImageRequest.Builder(context = LocalContext.current)
        .data(coverLink)
        .crossfade( enable: true)
        .build(),
    error = if(secondaryColor == Color.Black) {
        painterResource(R.drawable.white_error_outline)
    } else {
        painterResource(R.drawable.error_outline)
    },
    placeholder = if(secondaryColor == Color.Black) {
        painterResource(R.drawable.white_cloud_queue)
    } else {
        painterResource(R.drawable.cloud_queue)
    },
    contentScale = ContentScale.Crop,
    contentDescription = "cover",
    modifier = Modifier.fillMaxHeight()
)
```

Iné komponenty:

- Service – získavanie dát z api

Sieťová komunikácia:

- Komunikácia s api.mangadex.org pre získavanie údajov o mangách a ich následné čítanie

```
interface MangaDexApiService {
    Michal Zúbek
    @GET("manga")
    suspend fun getManga(
        @Query("title") title: String?,
        @Query("includedTags[]") includedTags: List<String>?,
        @Query("excludedTags[]") excludedTags: List<String>?,
        @QueryMap order: Map<String, String>?,
        @Query("includes[]") includes: List<String> = listOf("author", "artist", "cover_art"),
        @Query("contentRating[]") contentRating: List<String> = listOf("safe", "suggestive"),
        @Query("limit") limit: Int,
        @Query("offset") offset: Int
    ): MangaResponse
```

7. Záver

Myslím si, že implementácia moje aplikácie prebehla celkom úspešne. Jedna z hlavných vecí, ktoré si z tohto projektu ponesiem sú štrukturálne poznatky ako sa má správne aplikácia navrhovať či už pomocou viewmodelu alebo manažovanie stavov v compose a state hoisting.

Aplikáciu ešte plánujem vylepšiť už len z toho dôvodu, že ju plánujem aj sám používať. Niektoré chýbajúce funkcionality, ktoré neskôr pridám sú pridávanie do obľúbených a sťahovanie jednotlivých kapitol pre offline čítanie.

8. Zdroje

https://stackoverflow.com

https://www.youtube.com/@PhilippLackner

https://www.youtube.com/@DeveloperChunk

https://www.reddit.com/r/androiddev/

https://discord.com/invite/D2cNrqX (Android dev discord)