



ŽILINSKÁ UNIVERZITA V ŽILINE

**Fakulta riadenia
a informatiky**

Semestrálna práca z predmetu
vývoj aplikácií pre mobilné zariadenia

TRIPLI

Vypracoval: Filip Kubica

Študijná skupina: 5ZYR33

Akademický rok: 2024/2025

V Žiline dňa 1.4.2025



Obsah

| | |
|--|----|
| Úvod | 2 |
| Prehľad podobných aplikácií | 2 |
| Wanderlog – Trip Planner App | 2 |
| Triplt – Travel Planner | 2 |
| Roadtrippers – Trip Planner | 3 |
| Tripadvisor: Plan & Book Trips | 4 |
| Záver prehľadu podobných aplikácií | 4 |
| Analýza navrhovanej aplikácie | 5 |
| Jednoduchá štruktúra aplikácie..... | 5 |
| Návrh architektúry aplikácie | 6 |
| Domovská obrazovka | 6 |
| Zobrazenie itinerára | 6 |
| Detail miesta..... | 6 |
| Prihlásenie | 6 |
| Registrácia | 6 |
| Profil | 7 |
| Spodné navigačné menu | 7 |
| Návrh vzhľadu obrazoviek..... | 8 |
| Implementácia aplikácie..... | 9 |
| Vytvorenie štruktúry súborov | 9 |
| Vytvorenie modelov | 9 |
| Vytvorenie komunikácie so serverom | 9 |
| Autentifikačný systém | 10 |
| Tvorba UI obrazoviek..... | 11 |
| Google API | 12 |
| Navigácia v aplikácii..... | 12 |
| Zoznam zdrojov | 13 |

Úvod

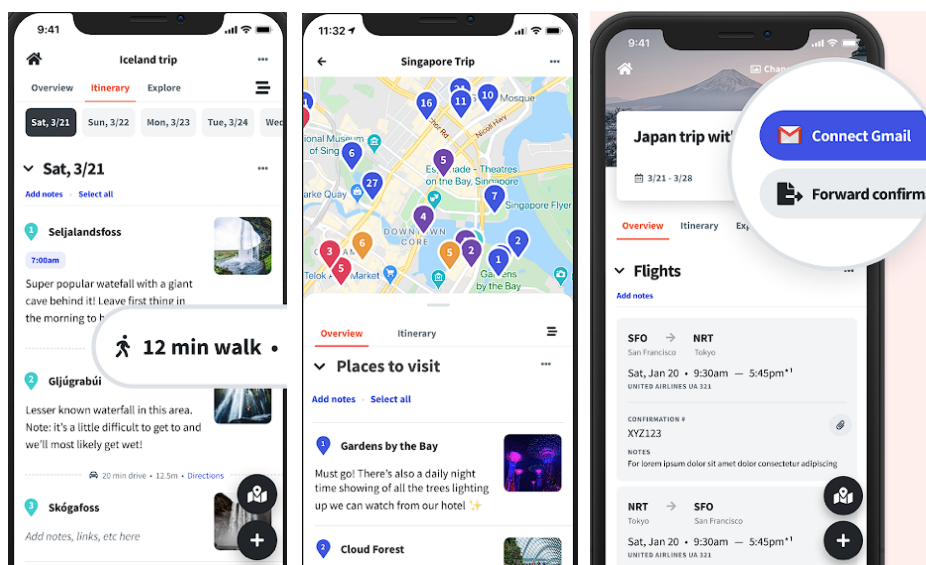
V dnešnej dobe, kedy máme tak ľahko prístupné cestovanie a nie jeden ľudský jedinec má túžbu spoznať kúsok sveta, aj keď i len v blízkom okolí, tak je dôležité si danú cestu dobre naplánovať. Na internete je množstvo aplikácií, ktoré nám môžu s týmto pomôcť. Či už Google Mapy, alebo iné z časti platené aplikácie s prémiovými funkciami. Tieto aplikácie vedia byť ale zároveň aj priam preplnené rôznymi funkcionalitami, že bežný používateľ, sa v nich stráca. Alebo ak sa chystáte na takzvané „budget“ cestovanie, tak niektoré platené aplikácie k tomu veľmi neprispievajú. Preto som sa rozhodol vyvinúť aplikáciu, ktorá bude určená pre cestovateľov, ktorí majú radi jednoduchý dizajn a ovládanie aplikácie a chcú na nej aj ušetriť. Tu sa začína cesta novej cestovateľskej aplikácie Tripli.

Prehľad podobných aplikácií

Wanderlog – Trip Planner App

Jednoduchá, bezplatná (okrem prémiových funkcií), cestovná aplikácia na plánovanie každého druhu výletu. Používateľ si môže vytvoriť vlastný itinerár, v ktorom si môže so spoluvýletníkmi organizovať celý program, ako napr. rezervácie letov, hotelov, áut a tak ďalej.

Aplikácia dokáže prepojiť emailový účet a tak importovať emailovú komunikáciu, ktorá sa týka napr. rezervácií. Umožňuje nastaviť aj rozpočty, sledovať výdavky a rozdeľovanie si účtov so skupinou. Funguje aj v offline (v platenej verzii) režime. V prémiovej verzii obsahuje algoritmus na optimalizáciu trasy, aby používateľ prešiel všetkými zadanými bodmi za čo najnižšiu spotrebu/čas/dĺžku trasy.

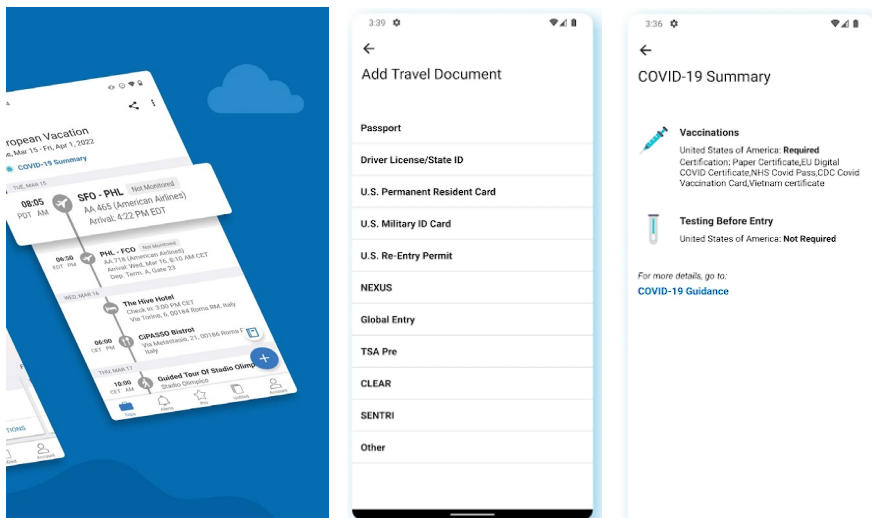


Obr. 1 – Aplikácia Wanderlog – Trip Planner App

Tripl – Travel Planner

Cestovateľská aplikácia. Umožňuje pridať existujúcu rezerváciu do plánu po zaslaní príslušného emailu na danú emailovú adresu. Tak isto umožňuje priložiť nový dokument do aplikácie, ako napr. vodičský preukaz pre daný štát, očkovací certifikát a rôzne iné dokumenty/certifikáty. Obsahuje jednotlivé informácie o očkovacích požiadavkách, alebo o covid opatreniach pre jednotlivé krajiny.

Oznamuje zmeny v odletových bránach na letiskách. Naviguje používateľov po jednotlivých letiskách pomocou interaktívnej mapy.



Obr. 2 – Aplikácia Triplt – Travel Planner

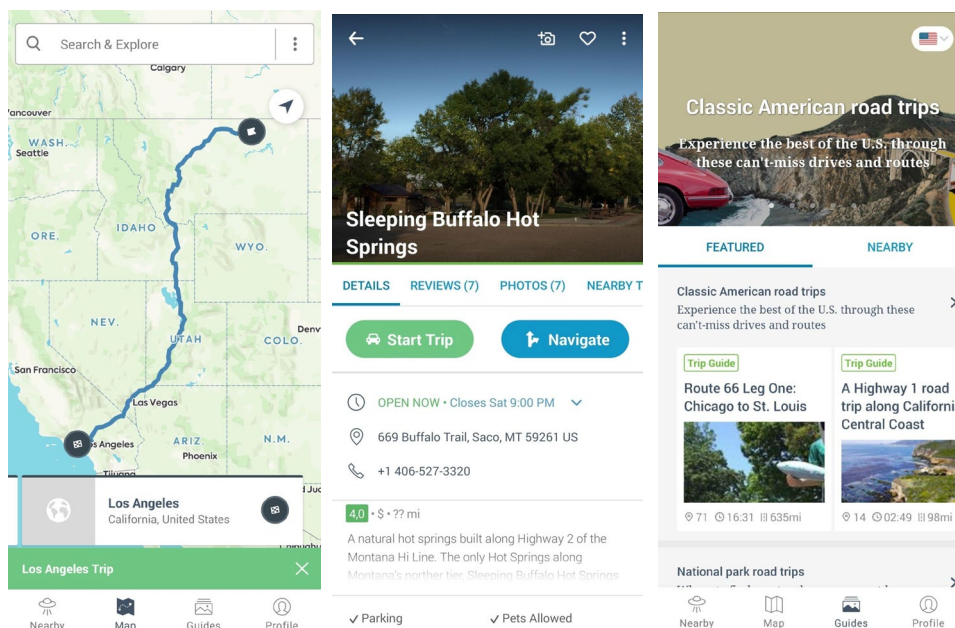
Roadtrippers – Trip Planner

Ideálna aplikácia na plánovanie si výletov po USA, Kanade, Austrálii a Novom Zélande. Ponúka široký výber miestnych reštaurácií, bizarných atrakcií pri cestách, malebných miest, národných parkov a hotelov. Má vopred pripravených turistických sprievodcov po najzaujímavejších a jedinečných trasách.

Používateľ si zvolí začínajúcu polohu a cieľovú destináciu svojej trasy. Aplikácia následne zobrazí odporúčanú trasu na mape. Následne si môže používateľ filtrovať jednotlivé kategórie, ktoré chce, aby sa mu zobrazili ako napr. ubytovanie, atrakcie, bary, reštaurácie, miesta na kempovanie, aktivity, čerpacie stanice, obchody a mnoho ďalšieho. Na základe týchto filtrov to používateľovi zobrazí najlepšie výsledky.

Aplikácia taktiež ponúka upozornenia na rozsiahle požiare, zápchy a iné obmedzenia, ktoré sa môžu počas cesty vyskytnúť.

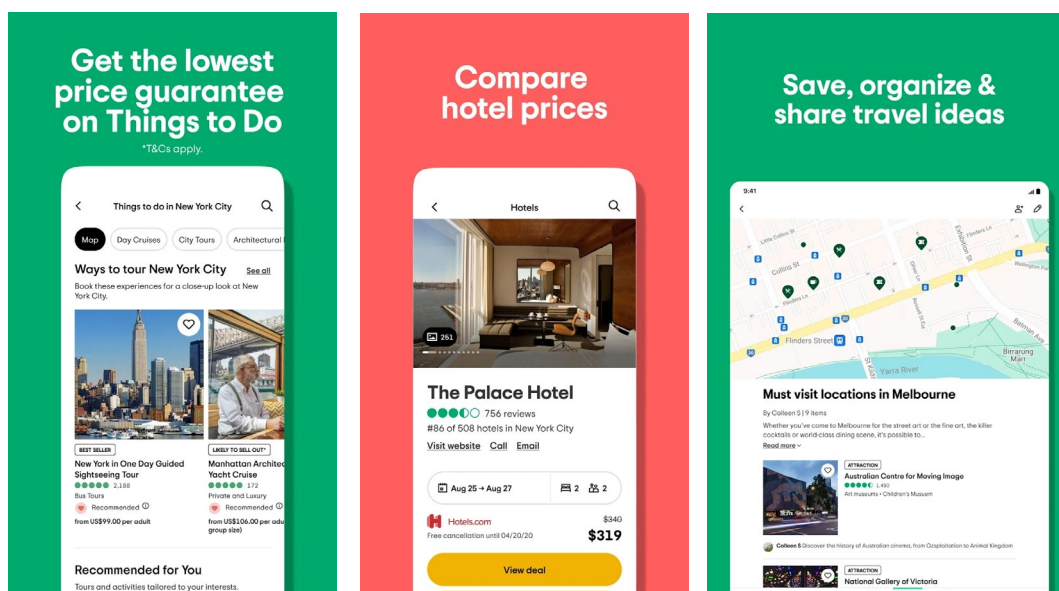
Pre tých, ktorí sa nevedia rozhodnúť, kam by chceli ísť, si môžu pozrieť už vopred spísané návrhy, teda itineráre na výlet od rôznych používateľov. Jednotlivé itineráre obsahujú hodnotenia, fotky, čo sa dá vidieť na výlete, kde sa ubytovať, kde jesť a mnoho ďalšieho.



Obr. 3 – Aplikácia Roadtrippers – Trip Planner

Tripadvisor: Plan & Book Trips

Jedna z najznámejších cestovateľských aplikácií. Vyhľadáva najlacnejšie ceny za jednotlivé atrakcie na viacerých zdrojoch. Porovnáva ceny jednotlivých ubytovaní a zobrazuje ich používateľovi, ktorý sa môže rozhodnúť, ktoré si vyberie. Zobrazuje najlepšie hodnotené ubytovanie/reštaurácie/aktivity v okolí. Používatelia môžu medzi sebou zdieľať jednotlivé itineráre. V aplikácii je veľká komunita ľudí, ktorí sú ochotní pomôcť a dať rôzne tipy a triky na cestovanie.



Obr. 4 – Tripadvisor: Plan & Book Trips

Záver prehľadu podobných aplikácií

Ak by som mal cestovať po USA, Kanade, Austrálii alebo Novom Zélande, tak by som si určite stiahol Roadtrippers – Trip planner, inak by som sa rozhodoval medzi zvyšnými, ostávajúcimi



aplikáciami, ktorých výber nie je ľahký. Každá má svoje plusy a mínusy. Nieкто, kto obľubuje plno funkcionálit, na úkor prehľadnosti aplikácie, by radšej siahol po Wanderlog – Trip Planner App. Na druhej strane, ak nieкто uprednostňuje minimalistickejší dizajn, na úkor menej funkcionálit, tak by mohol siahnuť po Tripadvisor: Plan & Book Trips alebo Triplt – Travel Planner. Medzi týmito dvomi aplikáciami je to už len o preferenciách používateľa, aké UI a UX sa mu viac páči.

Analýza navrhovanej aplikácie

Aplikácia má slúžiť na vytváranie a vyhľadanie ideálnych itinerárov pomocou filtra, ktorý si používateľ zvolí. Itineráre sú vytvorené používateľmi a nie cestovnými kanceláriami. Celá aplikácia je zadarmo, no pre prístup ku niektorým funkcionalitám je potrebné mať vytvorený účet a byť prihlásený.

| | |
|---------------------------|--|
| <i>názov projektu:</i> | Tripli |
| <i>typ projektu:</i> | mobilná aplikácia |
| <i>cieľové platformy:</i> | Android |
| <i>zameranie:</i> | cestovanie |
| <i>účel projektu:</i> | umožniť koncovým používateľom vyhľadávať itineráre od konkrétnych "cestovateľov", nie od cestoviek |
| <i>cieľová skupina:</i> | cestovatelia |

Jednoduchá štruktúra aplikácie

| | |
|--------------------------------|--|
| <i>zoznam itinerárov</i> | zoznam filtrovaných itinerárov s možnosťou ďalšieho filtrovania pomocou rozšíreného filtra |
| <i>detail itinerára</i> | zobrazenie itinerára s jednotlivými informáciami o ceste, ako nadpis, popis, mapa s lokáciami a jednotlivé lokácie |
| <i>profil</i> | možnosť odhlásenia sa, zoznam vytvorených itinerárov, jednoduché informácie o účte |
| <i>prihlásenie/registrácia</i> | prihlásenie alebo registrovanie nového používateľa do aplikácie |



Návrh architektúry aplikácie

Domovská obrazovka

Obrazovka, ktorá slúži ako hlavná stránka aplikácie. Bude tu zahrnuté vyberanie itinerárov zo zoznamu itinerárov alebo filtrovanie pomocou rozšíreného filtra, do ktorého sa používateľ dostane cez tlačidlo.

Zobrazenie itinerára

Obrazovka aplikácie, na ktorej je zobrazený itinerár. Používateľ si tu bude môcť prekliknúť medzi obrázkami výletu alebo mapou, na ktorej budú zobrazené body, ktoré by mal používateľ navštíviť v danom poradí. Stránka bude obsahovať:

- hodnotenie itinerára,
- popis výletu,
- dĺžka trvania výletu,
- cena výletu,
- meno autora,
- zobrazenie miest, ktoré by mal používateľ navštíviť
- preklik na detail miesta
- editovací režim, v ktorom používateľ bude môcť upraviť, alebo vytvoriť existujúci itinerár

Detail miesta

Detail miesta, je vlastne už úplne konkrétne miesto, ktoré neobsahuje žiadne veľké celky. V tomto bode sú samozrejme obrázky daného miesta, jeho pozícia na mape a popis. Daný bod bude doplnkové informácie, ako napríklad webstránka daného miesta ak ju má, čas, ktorý by mal používateľ tráviť na tomto mieste, alebo iné užitočné informácie. Používateľ sa môže pomocou tlačidla naspäť vrátiť na zobrazenie itinerára.

Prihlásenie

Obrazovka aplikácie, na ktorej sa používateľ bude môcť prihlásiť do svojho účtu vyplnením nasledovných údajov:

- e-mail
- heslo

Po vyplnení týchto polí, to používateľ potvrdí kliknutím na tlačidlo potvrdiť a v prípade, že boli údaje správne, tak bude mať používateľ prístup do svojho profilu a k svojim itinerárom. Ak používateľ nemá ešte vytvorený účet, tak z tejto obrazovky sa bude môcť prekliknúť na obrazovku pre registráciu.

Registrácia

Obrazovka aplikácie, na ktorej sa používateľ bude môcť zaregistrovať vyplnením nasledovných údajov:

- používateľské meno



- e-mail
- heslo
- potvrdenie hesla

Po vyplnení týchto polí, to používateľ potvrdí na tlačidlo potvrdiť a v prípade, že boli údaje vyplnené správne podľa požadovaných kritérií, bude používateľ automaticky prihlásený do aplikácie. Ak si používateľ spomenie, že už má vytvorený účet, tak z tejto obrazovky sa bude môcť prekliknúť na obrazovku pre prihlásenie.

Profil

Ak je používateľ prihlásený, tak sa mu zobrazí zoznam jeho vytvorených itinerárov, informácie o jeho profile, ale aj možnosť odhlásiť sa. Ak používateľ nie je prihlásený, tak ho to presmeruje na obrazovku pre prihlásenie sa.

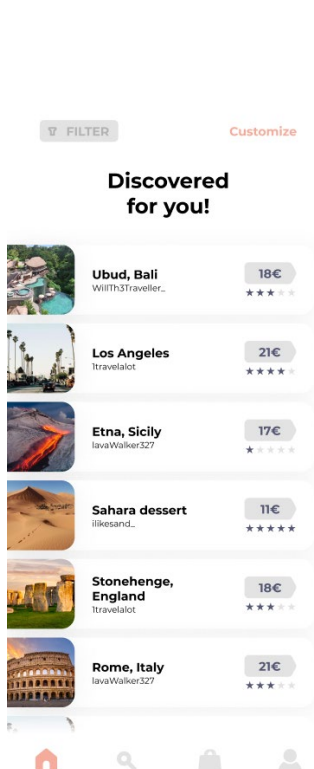
Spodné navigačné menu

Spodne navigačné menu slúži na navigáciu v aplikácii. Nachádza sa na každej obrazovke. V navigačnom menu sú 3 tlačidlá a to:

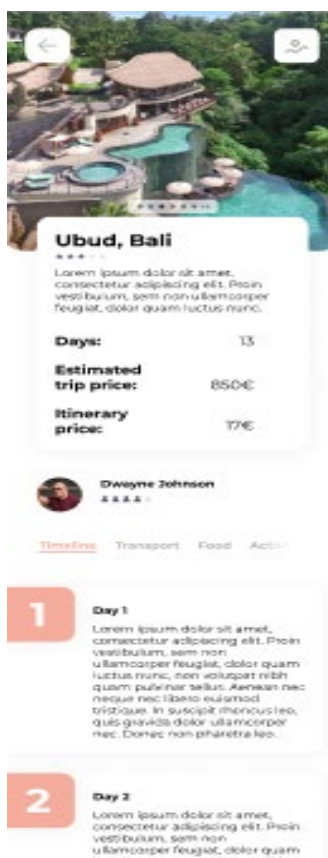
- tlačidlo, ktoré presmeruje na domovskú stránku,
- tlačidlo, ktoré presmeruje na vytvorenie itinerára,
- tlačidlo, ktoré presmeruje na používateľský profil



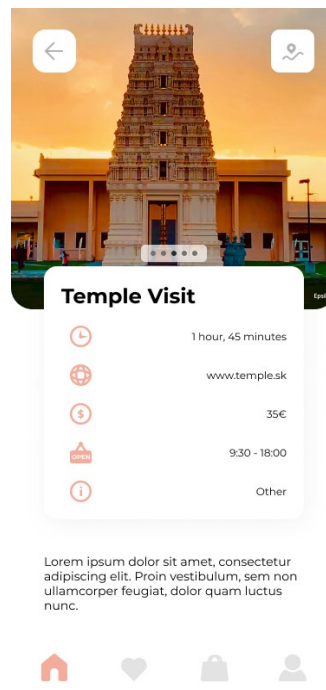
Návrh vzhľadu obrazoviek



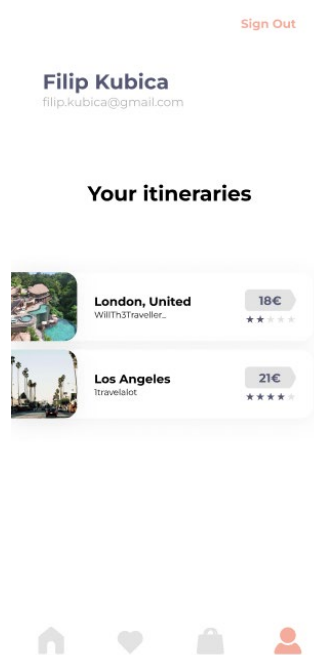
Obr. 5 – Domovská obrazovka



Obr. 6 – Zobrazenie itinerára



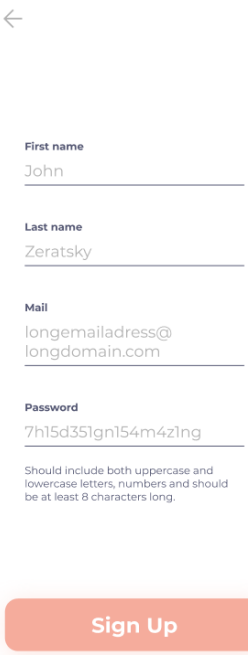
Obr. 7 – Konkrétne miesto



Obr. 8 – Profil



Obr. 9 – Prihlásenie



Obr. 10 - Registrácia

Implementacia aplikácie

Vytvorenie štruktúry súborov

V priečinku UI (používateľské rozhranie) si definujeme celú vizuálnu časť a prácu s ňou. Máme tu definované jednotlivé obrazovky a ich logiku na prácu s nimi.

Na začiatku implementácie aplikácie bolo dôležité navrhnuť základnú štruktúru projektu, v ktorej by bola orientácia intuitívna a prehľadná. Štruktúru súborov sme implementovali nasledovne. V koreni projektu sme vytvorili priečinky, resp. balíky pre utils, domain, dependency injection, data a ui (používateľské rozhranie).

V priečinku utils sa nachádzajú znovupoužiteľné metódy, ktoré sa využívajú vo viacerých odlišných súboroch.

V priečinku domain si uchováame stav o prihlásenom používateľovi, údaje o ňom a jeho bezpečnostný token používaný pri komunikácii so serverom.

V priečinku dependency injection (di) sa nachádzajú súbory, ktoré vieme používať odkiaľkoľvek v aplikácii bez potreby ich odovzdávania pomocou parametrov metód a tried.

V priečinku data sa nachádzajú ďalšie podpriečinky, ako model, network a repository. V priečinku model si uchováame dátové modely objektov, ktoré budeme používať v aplikácii. V priečinku network sú súbory, ktoré pracujú so sieťovou komunikáciou, ako definovanie interceptora alebo autentifikačného „strážcu“, tzv. guard. Ďalej sa tam nachádzajú ešte API cesty a metódy, ktoré definujú, čo sa robí pri volaní jednotlivých API ciest, aké dáta sa odosielajú a ako sa spracovávajú.

V priečinku ui (používateľské rozhranie) si definujeme celú vizuálnu časť a prácu s ňou. Máme tu definované jednotlivé obrazovky a ich logiku na prácu s nimi.

Vytvorenie modelov

Po vytvorení základnej štruktúry súborov sme vytvorili jednotlivé modely objektov, ktoré používame v aplikácii. Medzi ne patria napríklad objekty Place, Itinerary, LoginRequest, OptimalRouteResponsea podobne.

```
1 package com.example.tripli.data.model.routeOptimization
2
3 data class OptimalRouteResponse(
4     val route: List<RoutePlace>,
5     val distanceMeters: Int,
6     val durationSeconds: Int
7 )
```

Obr. 11 – Dátový model pre OptimalRouteResponse

Vytvorenie komunikácie so serverom

Aby sme mohli vyhľadávať, vytvárať a spravovať itineráre alebo sa registrovať či prihlásiť do aplikácie, potrebovali sme implementovať komunikáciu so serverom. Začali sme definovaním API ciest, na ktoré náš server „počúva“.

```
interface ItineraryApiService {  
    @GET("v1/tripli/itineraries")  
    suspend fun getItineraries(@QueryMap filters: Map<String, String>? = null): Response<JsonObject>  
  
    @GET("v1/tripli/itineraries/{id}")  
    suspend fun getItineraryById(@Path("id") id: Int): Response<JsonObject>  
  
    @POST("v1/tripli/itineraries/create")  
    suspend fun createItinerary(@Body itinerary: Itinerary): Response<JsonObject>  
  
    @PUT("v1/tripli/itineraries/{id}")  
    suspend fun updateItinerary(@Path("id") id: Int, @Body itinerary: Itinerary): Response<JsonObject>  
  
    @DELETE("v1/tripli/itineraries/{id}")  
    suspend fun deleteItinerary(@Path("id") id: Int): Response<JsonObject>  
  
    @POST("v1/tripli/routes/optimal-route")  
    suspend fun getOptimalRoute(@Body routeMatrixRequest: RouteMatrixRequest): Response<JsonObject>  
}
```

Obr. 12 – Interface pre ItineraryApiResponse

Následne sme implementovali jednotlivé metódy, ako `getItinerary`, `createItinerary`, `getOptimalRoute` a podobne. V týchto metódach je uložená logika, čo sa má diať po ich zavolaní, napríklad aké dáta sa majú odoslať na server, akú odpoveď očakávame a ako ju spracujeme.

Aby mohla prebiehať komunikácia so serverom, potrebovali sme na to vhodný nástroj. Použili sme Retrofit v kombinácii s `OkHttpClient`. Retrofit používame na volanie jednotlivých HTTP REST API požiadaviek a `OkHttpClient` na využitie interceptora, či už spomenutého „strážcu“ – guard. Z dôvodu zložitej logiky máme definované dve Retrofit inštancie a dve `OkHttpClient` inštancie. Jedna skupina je vytvorená bez interceptora a strážcu a druhá už s plnohodnotnou kontrolou autentifikácie.

Autentifikačný systém

Aby sa zabezpečila bezpečnosť dát používateľov, napríklad aby používateľom vytvorené itineráre vedel spravovať iba autor daného itinerára, bolo potrebné vytvoriť autentifikačný systém. V aplikácii sa používajú tokeny na overenie používateľa.

Po úspešnom prihlásení do aplikácie server vráti v odpovedi bezpečnostný token, ktorý sa používa na overenie používateľa. Tento token je potrebné odosielať na server v hlavičke HTTP požiadaviek vždy, keď sa volajú požiadavky na server, ktoré to vyžadujú. Medzi ne patrí napríklad vytvorenie itinerára, jeho úprava a vymazanie.

Vytvorili sme preto tzv. `AuthInterceptor`, ktorý má na starosti vkladať do každej požiadavky daný token, ktorý následne server overí. Tieto tokeny však majú určitú dobu platnosti na serveri. Po uplynutí stanovenej doby sa token stáva neplatným, čo znamená, že je potrebné vygenerovať nový. Toto by sme mohli docieľiť opätovným prihlásením používateľa, no takéto riešenie je nekomfortné. Preto je na serveri implementovaná funkcionálna kontrola, ktorá vie token obnoviť.

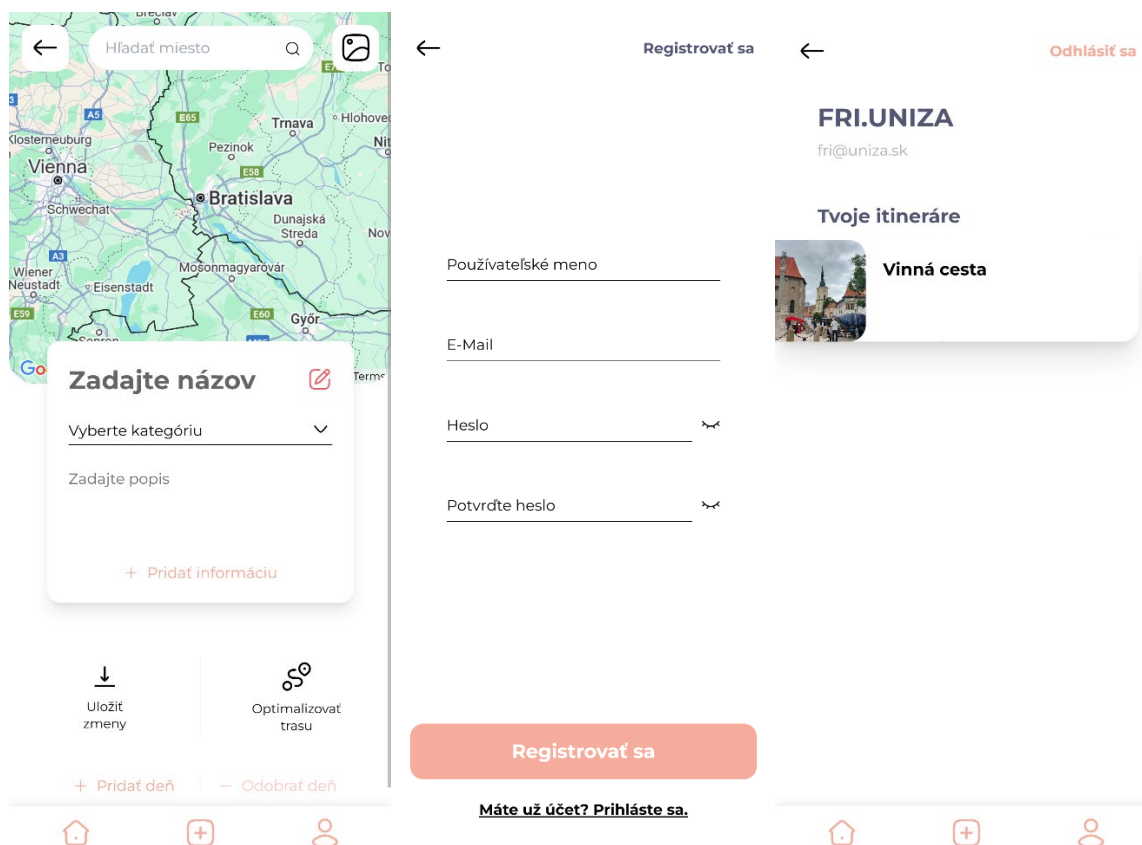
Vždy, keď platnosť tokenu vyprší, vráti sa v odpovedi zo servera kód 401 – Unauthorized. V triede `AuthGuard` sledujeme, či nám prišla odpoveď s takýmto kódom, a ak áno, zavoláme na server požiadavku na obnovenie tokenu. Ak požiadavka uspeje, aktualizujeme používateľovi token. V opačnom prípade používateľa odhlásime z aplikácie.

Tvorba UI obrazoviek

Pri tvorbe obrazoviek sa v Jetpack Compose často používa štruktúra MVVM (Model – View – ViewModel). V rámci každej obrazovky sme vytvorili štyri súbory: NazovScreen, NazovContent, NazovUiState, NazovViewModel.

- V NazovViewModel je celá logika funkcionalít obrazovky, napríklad čo sa stane po kliknutí na tlačidlo, aké dáta sa načítajú a kedy sa zobrazia.
- V NazovUiState je uchovaný aktuálny stav obrazovky, napríklad údaje zadané v poliach alebo načítaný itinerár.
- V NazovContent je definované, ako má vyzeráť obrazovka – kde má byť tlačidlo, akú má mať farbu a podobne.
- NazovScreen slúži ako premostenie NazovViewModel a NazovUiState s NazovContent. NazovUiState je tiež dátová trieda.

V tomto kroku sme vytvorili všetky potrebné obrazovky: Home, Itinerary, Place, Profile, Login, Register a Profile.



Obr. 13 – Ukážky implementácie obrazoviek

Google API

Aby sme do itinerára mohli vyhľadávať a vkladať miesta, potrebovali sme ich získať z externého zdroja. Rozhodli sme sa pre Google API, konkrétne Google Places API, Google Maps API, Google Autocomplete API a Google Geocoding API.

- Google Places API slúži na získavanie údajov o konkrétnom mieste na základe jeho jedinečného place ID (názov, poloha, adresa, fotografie, kategória atď.).
- Google Maps API slúži na zobrazenie mapy, na ktorej vieme vizuálne vyhľadávať miesta, vybrať ich a zobrazovať označenia.
- Google Autocomplete API slúži na vyhľadávanie miest pomocou vyhľadávacieho poľa. Na základe zadaného textu vracia predikcie toho, čo sa snažíme vyhľadať, a place ID vybraného miesta, ktoré následne použijeme v Places API.
- Google Geocoding API slúži na získanie place ID z geografických súradníc. Ak klikneme na mape na objekt bez označenia, dostaneme jeho súradnice a tie použijeme na získanie place ID.

Pre mapu aj vyhľadávacie pole s predikciami sme vytvorili samostatné globálne komponenty.

Navigácia v aplikácii

Aby si používateľ mohol zobrazíť všetky potrebné obrazovky, bolo potrebné vytvoriť navigačný systém so spodným navigačným menu.

Pre každú obrazovku sme vytvorili dátovú triedu Screen definujúcu cestu k nej. Následne sme vytvorili vizuál spodného navigačného menu a implementovali triedu AppNavGraph. Táto trieda slúži ako mapa, ktorá definuje, aké obrazovky v aplikácii existujú a ako sa medzi nimi naviguje v rôznych stavoch.

Napríklad ak používateľ nie je prihlásený a chce zobrazíť profil alebo vytvoriť itinerár, aplikácia ho namiesto toho presmeruje na prihlasovaciu obrazovku.

V rámci tejto navigácie vytvárame aj príslušné viewmodely pre jednotlivé obrazovky alebo ich zdieľame medzi obrazovkami podľa potreby.



Zoznam zdrojov

<https://wanderlog.com/>

<https://www.tripit.com/web>

<https://roadtrippers.com/>

<https://www.tripadvisor.com/>