SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Goran Čehulić, gcehulic@foi.hr**

**Marko Marciuš, mmarcius@foi.hr**

**Mateo Cindrić, matcindri@foi.hr**

WatchMe

PROJEKT IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA

TEHNIČKA DOKUMENTACIJA

Varaždin, 2017.SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Tim: AIR602

**GitHub:** <https://github.com/gcehulic/WatchMe>

Goran Čehulić, 45320/16-R

Marko Marciuš, 44453/15-R

Mateo Cindrić,45277/16-R

WatchMe

PROJEKT IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA

TEHNIČKADOKUMENTACIJA

Mentor:

Dr. sc. Ivan Švogor

Varaždin, veljača 2017.

Sadržaj

[1. Uvod 1](#_Toc474842451)

[1.1 Podaci o tvrtki partneru 1](#_Toc474842452)

[1.2 Korištene tehnologije 1](#_Toc474842453)

[1.3 Temeljne funkcionalnosti 2](#_Toc474842454)

[1.4 Nefunkcionalni zahtjevi aplikacije 4](#_Toc474842455)

[2. Arhitektura aplikacije 5](#_Toc474842456)

[2.1 Detaljne funkcionalnosti aplikacije 5](#_Toc474842457)

[2.2 Skica aplikacije 6](#_Toc474842458)

[2.3 Implementacijski zadaci 7](#_Toc474842459)

[2.4 ERA model baze podataka 8](#_Toc474842460)

[2.5 Dijagrami klasa 9](#_Toc474842461)

[2.5.1 Dijagram klasa database 9](#_Toc474842462)

[2.5.2 Dijagram klasa app modul 10](#_Toc474842463)

[2.5.3 Dijagram klasa notification modul 11](#_Toc474842464)

[2.6 Vanjske biblioteke 13](#_Toc474842465)

[3. Podaci za prijavu u aplikaciju 15](#_Toc474842466)

[4. Testiranje prihvatljivosti 16](#_Toc474842467)

# 1. Uvod

Aplikacija WatchMe je namijenjena praćenju popularnih TV serija. Za svoj rad aplikacija koristi Trakt.tv API. Pomoću njega dohvaćaju se podaci o serijama koji se koriste u ovoj aplikaciji.

## 1.1 Podaci o tvrtki partneru

Degordian je zagrebačka agencija koja se bavi gotovo svime vezanim uz digitalni marketing. Tvrtka ima nešto više od sto zaposlenika, a osim zagrebačkog ima urede i u  Beču, Beogradu, Mostaru, Londonu i Ljubljani. Naš mentor iz te tvrtke je Tomislav Smrečki i zaposlen je na poziciji Android Developer-a.

## 1.2 Korištene tehnologije

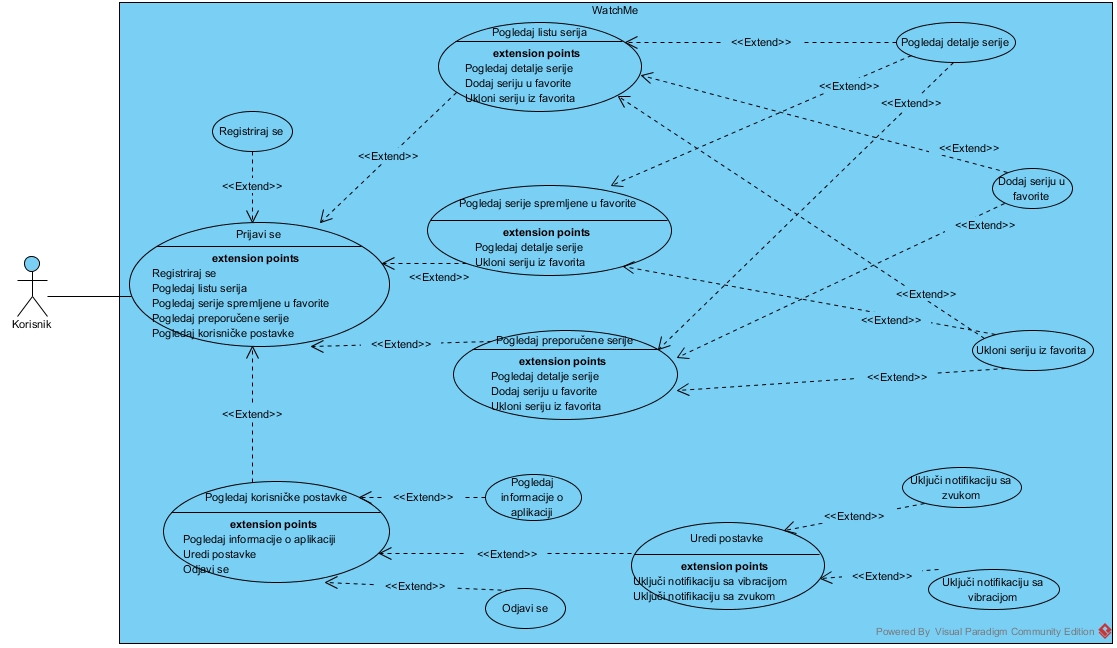
* Tehnologije:
  + Java
  + Android
* Alati:
  + Android Studio
  + Genymotion
  + VisualParadigm
  + Microsoft Office
  + GitHub (<https://github.com/gcehulic/WatchMe>)

## 1.3 Temeljne funkcionalnosti

Temeljne funkcionalnosti aplikacije WatchMe su sljedeće:

* Registracija korisnika
* Prijava korisnika u aplikaciju s korisničkim imenom i lozinkom
* Korisnik može pregledavati najpopularnije serije, te klikom na pojedinu seriju se prikažu detalji o njoj. Za dohvat podataka o serijama koristit ćemo API
* Korisnik može spremiti seriju kao favorit
* Korisnik će dobivati notifikacije prilikom emitiranja serije koje su dodane u favorite
* Korisnik može odabrati želi li zvuk ili vibraciju kod dolaska obavijesti
* Temeljem podataka o serijama pohranjenim u favorite (žanr) aplikacija bi preporučila korisniku nove serije koje bi mu se mogle svidjeti.

Dijagramom slučajeva korištenja (eng. Use Case Diagram) smo prikazali slučajeve korištenja koji su definirani nakon što su utvrđeni zahtjevi aplikacije (Slika 1). Ovaj dijagram prikazuje opis osnovnih funkcionalnih cjelina i vanjskih učesnika (korisnik aplikacije) te interakciju korisnika sa sustavom.



Slika 1 Dijagram slučajeva korištenja

## 1.4 Nefunkcionalni zahtjevi aplikacije

Nefunkcionalni zahtjevi odnose se na zahtjeve koji su potrebni kako bi bilo omogućeno neometano korištenje aplikacije. Sama aplikacija za svoj rad zahtjeva stalnu povezanost na internet. Razlog tome je potreba za stalnim dohvaćanjem podataka o serijama. Samu aplikaciju moguće je koristiti na pametnom telefonu ili tabletu. Bitno je naglasiti da je na uređaju potrebno imati instaliran android operacijski sustav minimalne verzije 4.4 (KitKat) te da je minimalna SDK verzija 19. Autentifikacija korisnika izvršava se upisom korisničkog imena i lozinke pa nije potrebno imati senzore poput čitača otiska prstiju. Brzina rada aplikacije ovisi o kvaliteti interneta kojeg korisnik koristi. Podaci o serijama koje je korisnik dodao u favorite spremaju se u lokalnu bazu podataka iz koje je čitanje postojećih i dodavanje novih zapisa brže. Iduće je da uređaj treba imati mogućnost pristupa YouTube servisu (za pregledavanje kratkog filma o seriji – detalji serije).

# 2. Arhitektura aplikacije

Aplikacija će biti modularna, što znači da ćemo kreirati u potpunosti neovisne dijelove koda koji s ostatkom aplikacije komuniciraju preko definiranih sučelja. U aplikaciji ćemo imati nekoliko funkcionalnih modula. Prvi je *app* koji je glavni modul aplikacije, potom modul koji će biti kreiran u posljednjem sprintu za slanje obavijesti o emitiranju serije. Tako je uz *app* modul prisutan i modul *notification*. Primjenom modularnosti omogućit ćemo ponovnu iskoristivost napisanog programskog koda te mogućnost nadogradnje aplikacije, odnosno programskog koda.

## 

## 2.1 Detaljne funkcionalnosti aplikacije

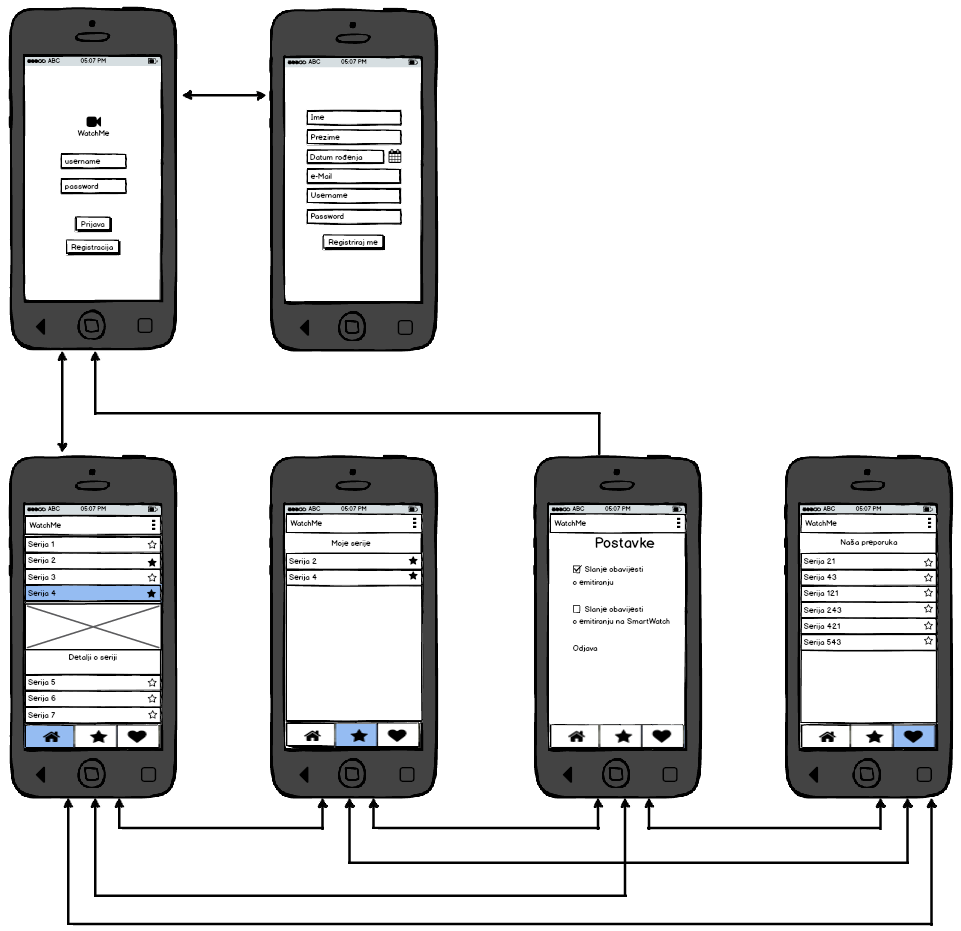
Detaljne funkcionalnosti aplikacije WatchMe:

* Da bi korisnik mogao koristiti aplikaciju, mora se prethodno registrirati. Registracija korisnika obuhvaća informacije kao što su ime, prezime, email, korisničko ime te lozinka.
* Prijava korisnika s korisničkim imenom i lozinkom. Provjera ispravnosti unesenih podataka, tj. da li korisničko ime i lozinka postoje u bazi registriranih korisnika.
* Nakon uspješne prijave, korisniku se prikazuje početni zaslon gdje se nalazi popis serija. Klikom na pojedinu seriju se prikazuju detalji o toj seriji. Za dohvat podataka o serijama se koristi Trakt.tv API koji je dostupan na poveznici:

<http://docs.trakt.apiary.io/#reference/shows/get-trending-shows> .

* Pored svake serije se nalazi simbol zvjezdice, a klikom na zvjezdicu se ta serija sprema pod favorite. Podaci o toj seriji spremaju se u bazu. Ponovnim klikom na zvjezdicu, serija se uklanja s popisa Favorita.
* Sama funkcionalnost slanja obavjesti korisniku implementirana je pomoću AlarmManager sustava. Aplikacija uzima iz lokalne baze podataka vremena emitiranja pojedinih serija, koje se onda konvertira u europsko vrijeme. Korisnik u postavkama ima mogućnost odabira načina obavijesti (zvuk ili vibracija).
* Posljednja funkcionalnost je preporuka serija. Pomoću algoritma temeljem podataka o serijama pohranjenim u favorite (žanr), aplikacija bi preporučila korisniku nove serije koje bi mu se mogle svidjeti.

## 2.2 Skica aplikacije



Slika 2 Skica aplikacije

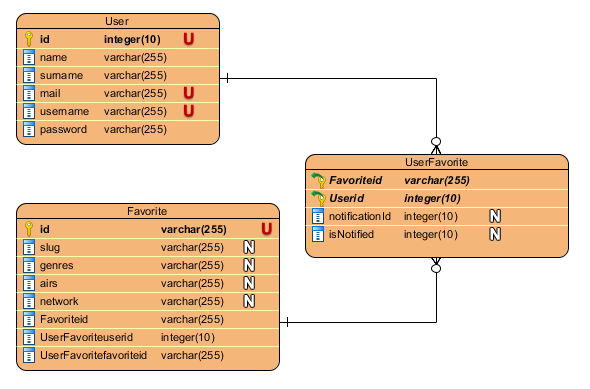
Slika 2 prikazuje skicu aplikacije. Predviđeno je da će nakon ulogiranja korisnika biti tri taba na dnu ekrana (u obliku tri kartice) te će se tako korisnik moći kretati kroz zaslone. Biti će i zaslon postavki koji je namjenjen za korisničke postavke o načinu slanja obavijesti.

## 2.3 Implementacijski zadaci

|  |  |
| --- | --- |
| **Zadatak** | **Opis** |
| **Sprint 1 (24.10. – 11.11.)** | |
| Izrada korisničkog sučelja za prijavu | Izrada korisničkog sučelja i klase za upravljanje korisničkim sučeljem za prijavu. |
| Izrada korisničkog sučelja za registraciju | Izrada korisničkog sučelja i klase za upravljanje korisničkim sučeljem za registraciju. |
| „Autorizacija“ korisnika | Provjera da li postoji aktivni korisnik u bazi prema upisanom korisničkom imenu i lozinki. |
| Spremanje podataka registriranog korisnika | Spremanje podataka o novom korisniku u bazu (ime, prezime, email, korisničko ime, lozinka). |
| Izrada glavnog izbornika | Kreiranje osnovne navigacijske strukture aplikacije. |
| Izrada modula baze podataka | Baza podataka za potrebe registracije i prijave korisnika. |
| **Sprint 2 (14.11. – 10.12.)** | |
| Dohvat serija s API i njihov prikaz na početnom zaslonu (home) | Prikazati najpopularnije serije na početnom zaslonu |
| Prikaz detalja o seriji | Klikom na pojedinu seriju se prikazuju detaljne informacije o toj seriji (naziv, žanr,...). |
| **Sprint 3 (12.12. – 31.12.)** | |
| Označavanje serija kao favorit | Staviti oznaku zvjezdice pored svake serije te označene serije pohraniti u favorite |
| Pohrana podataka o seriji u bazu | Pohraniti podatke o seriji (naziv, žanr,...) u bazu podataka |
| Uklanjanje serije iz favorita | Ponovnim pritiskom na simbol zvjezdice pored serije se ta serija uklanja iz popisa favorita i briše se iz baze |
| **Sprint 4 (02.01. – 21.01.)** | |
| Kreirati modul za funkcionalnost primanja obavijesti o emitiranju serija | Potrebno je kreirati modul koji implementira funkcionalnost primanja obavijesti o emitiranju serija. |
| Algoritam za preporuke serija | Pomoću nekog algoritma temeljem podataka o serijama pohranjenim u favorite (žanr), aplikacija bi preporučila korisniku nove serije koje bi mu se mogle svidjeti |
| Prikaz preporučenih serija na zaslonu | Prikazati listu preporučenih serija na zaslonu „Naša preporuka“ |

Tablica 1. Implementacijski zadaci

## 2.4 ERA model baze podataka



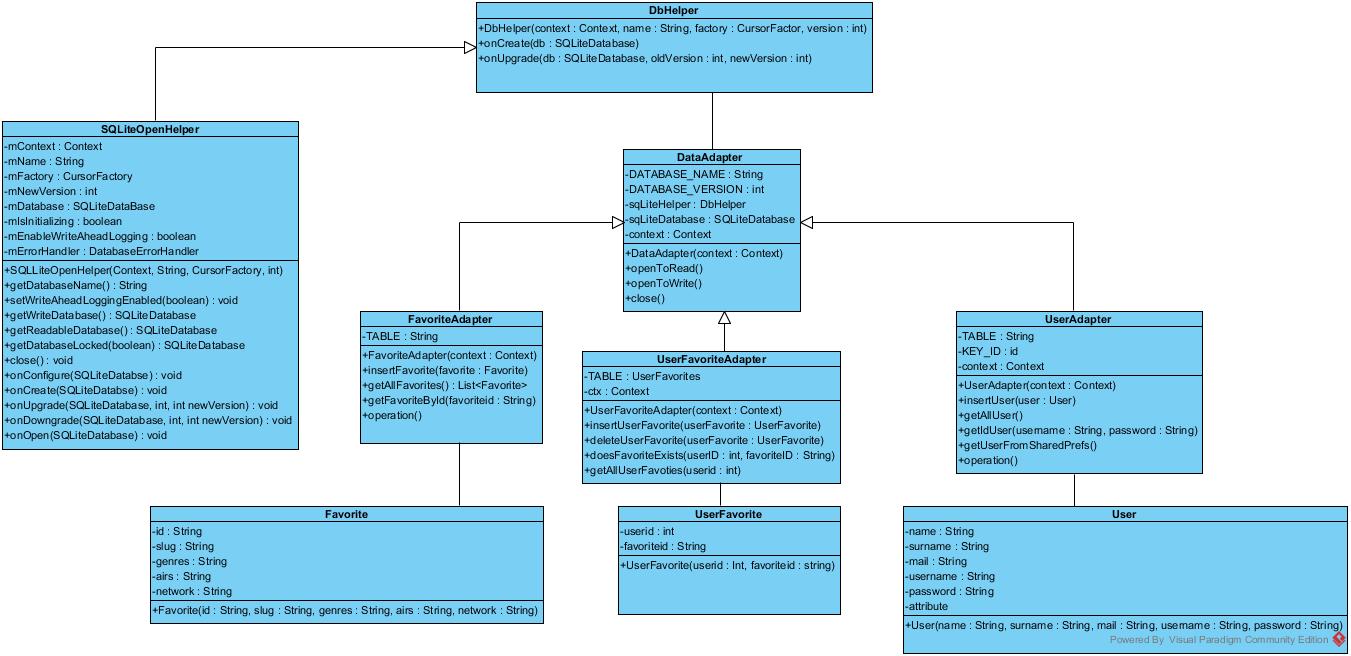
Slika 3ERA model baze podataka

Baza podataka sastoji se od 3 tablice (Slika 3). U tablici User nalaze se podaci o korisnicima. U tablicu Favorite sprema se podaci o seriji koji će se koristiti za implementaciju funkcionalnosti preporuke serija i obavijesti o emitiranju serija. Tablica UserFavorite koristi se kako bi omogućili da pojedini korisnik može spremiti više serija i da više korisnika može spremiti istu seriju.

## 2.5 Dijagrami klasa

Unutar ovog dijela možete vidjeti dijagrame klasa za svaki od modula aplikacije. Definirana su dva modula, *app* modul te *notification* modul. Radi preglednosti je dijagram klasa *database* izdvojen u posebni prikaz.

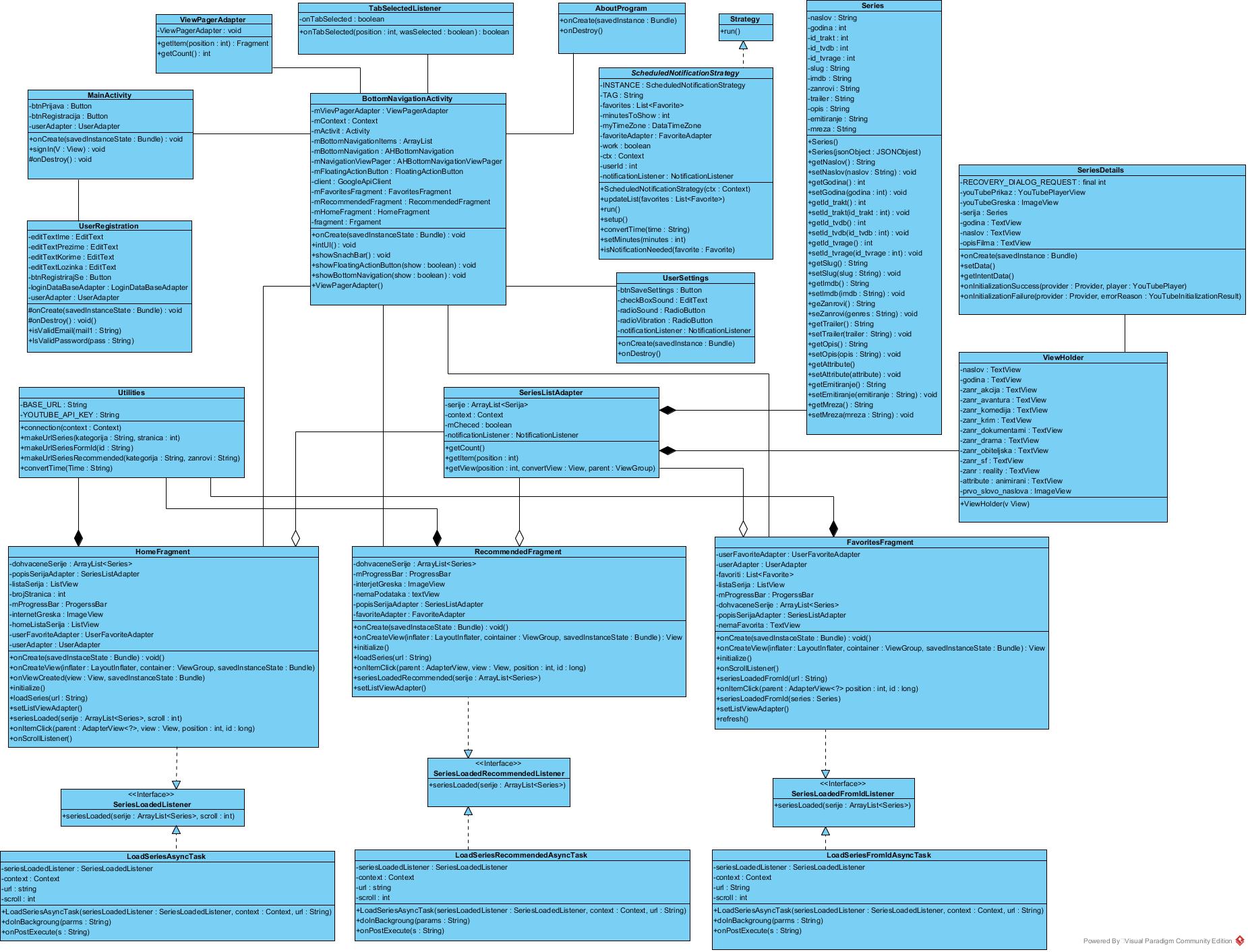
### 2.5.1 Dijagram klasa database



Slika 4 Dijagram klasa: sqlLite

Na slici 4 nalazi se 9 klasa koje se koriste za nativni pristup bazi podataka. Klasa SQLiteOpenHelper sadrži metode za rad sa SQLite bazom podataka. Tu klasu nasljeđuje klasa DbHelper koja sadrži dvije metode onCreate i onUpgrade. Klase Favorite, UserFavorite i User su entitetne klase te nasljeđuju klasu DataAdapter. Svaka entitenta klasa ima jedinstvenu adaptersku klasu. Primjer takve klase je entitentna klasa Favorite koja ima nasljeđenu adaptersku klasu FavoriteAdapter. Klasa User se koristi za spremanje podataka kod registracije korisnika te se iz nje čitaju podaci prilikom prijave. Klasa Favorite koristi za spremanje odabranih serija koje se čitaju i prikazuju u klasi FavoriteFragment.

### 2.5.2 Dijagram klasa app modul

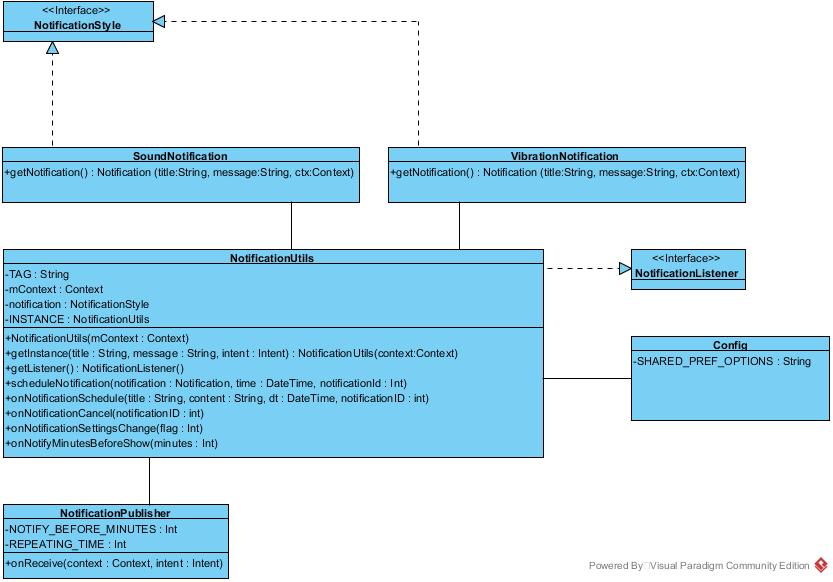


Slika 5 Dijagram klasa app modul

Na slici 5 se može vidjeti dijagram klasa app modula. Kod app modula možemo izdvojiti 23 klasa. Klasa UserRegistrationsadrži atribute koji se odnose na podatke iz tablice „User“. Ona nam služi za registraciju korisnika u aplikaciju. Za prijavu u aplikaciju koristimo klasu MainActivity. Ukoliko smo uspješno ulogirani u aplikaciju aplikacija ona nas prebaci na klasu HomeFragment. U klasi HomeFragment aplikacija prikazuje nove serije koje dohvaća s interneta. S klasom HomeFragment povezane su klase SeriesListAdapter, Utilities i SeriesLoadListener. U klasi SeriesDetails nalaze se podaci o serijama te su povezane klase ViewHolder. Klasa Utilities je povezana s klasama HomeFragment, FavoritesFragment, RecommendedFragment. Služi za povezivanje s API-om s kojeg se dohvaćaju serije. Klasa BottomNavigationActivity služi za dizajn te su preko nje povezane klase HomeFragment, FavoritesFragment, RecommendedFragment, MainActivity dok klase ViewPagerAdapter i TabSelectedListener se nalaze unutar same klase BottomNavigationActivity.

Klasa ViewPagerAdapter služi za definiranje pozicije fragmenata dok klasa TabSelectedListener definira metodu koja javlja klasi BottomNavigationActivitykada da promjeni fragment. U modulu se nalaze još klase UserSettings i AboutProgram. UserSettings klasa povezana je s klasom BottomNavigationActivity. Klasa FavoritesFragment prikazuje serije koje je korisnik odabrao te pomoću njih algoritam daje podatke za klasu RecommendedFragment. Klasa RecommendedFragment prikazuje preporučene serije korisniku.

### 2.5.3 Dijagram klasa notification modul



Slika 6 Dijagram klasa notification modul

Na slici 6 se može vidjeti dijagram klasa notification modula koji ima 7 klasa. U klasi Config definirani su koji podaci se uzimaju iz SharedPreferences. Klasa NotificationPublisher povezna je s klasom NotificationUtils. Klasa NotificationUtils povezana je s klasama SoundNotification i VibrationNotification. Klasa NotificationStyle služi za kreiranje notifikacija. Te je su s klasom NotificationStyle povezane klase SoundNotification i VibrationNotification. SoundNotification gradi notifikacije koje će imati implementiran zvuk, a klasa VibrationNotification će graditi notifikaciju koja ima implementiranu vibraciju.

Klasa NotificationPublisher ima ugrađen broadcastReciver koji sluša alarm menager. Također u njoj se nalazi notificationManager koji prikazuje notifikacije. Klasa notificationListener je interface koji povezuje modul notifikacije s app modulom.

## 2.6 Vanjske biblioteke

Sljedeća tablica (Tablica 2) prikazuje plan korištenja vanjskih biblioteka.

|  |  |
| --- | --- |
| **Naziv vanjske biblioteke** | **Svrha** |
| **com.aurelhubert:ahbottomnavigation:1.3.3** | Biblioteka je korištena za kreiranje dizajna navigacije u obliku kartica koje se nalaze na dnu ekrana. Odlučili smo se za ovaj oblik navigacije budući je ovakav oblik idealan do 4 navigacijske kartice, a u aplikaciji su primjenjene tri kartice. Više od četiri kartice bi stvaralo gužvu i ne bi imalo smisla primjeniti ovakav oblik navigacije. |
| **libs/YouTubeAndroidPlayerApi.jar** | Ovime je omogućeno da se uključi funkcija reprodukcije videozapisa, a kontrole koje su dodatno uključene su opcije pokretanja i pauziranja videa, te odlaska na određenu točku videa koji se prikazuje. |
| **com.amulyakhare:com.amulyakhare.textdrawable :1.0.1** | Ova biblioteka omogućuje stvaranje slike koja obuhvaća prvo slovo teksta, konkretno prikazuje sliku s prvim slovom naziva serije |
| **joda-time:joda-time:2.9.4** | Koristi se u modulu notification. Joda-time je biblioteka za datum i vrijeme. U aplikaciji se koristi u klasi ScheduledNotificationStrategy.java kako bi se uzela vremenska zona za Europu/Zagreb te radi izračuna datuma i vremena kada će se poslati notifikacija o emitiranju serije. |

Tablica 2. Vanjske biblioteke

# 3. Podaci za prijavu u aplikaciju

Nakon instaliranja aplikacije na mobilni uređaj, potrebno se registrirati kako bi se kasnije moglo sa korisničkim imenom i lozinkom ulogirati u aplikaciju ili možete koristiti korisničko ime i lozinku automatski kreiranog korisnika (Default User). Za registraciju morate navesti ime, prezime, email (VAŽNO: email mora biti oblika npr. [nešto@nešto.nešto](mailto:nešto@nešto.nešto)), korisničko ime te lozinku (VAŽNO: lozinka mora imati najmanje 7 znakova). Uvjet za uspješnu prijavu je da su sva polja popunjena te da upisano korisničko ime ne postoji u bazi. Baza je lokalna i vezana je samo uz konkretni uređaj.

Podaci za prijavu (Default User):

* Username: **user**
* Password: **password**

# 4. Testiranje prihvatljivosti

Testiranje prihvatljivosti aplikacije proveli smo temeljem kriterija tražene funkcionalnosti. Svaku od napravljenih funkcionalnosti moguće je ocjeniti sa prihvatljivom odnosno neprihvatljivom. Svaku od funkcionalnosti napisan je komentar kojim je obrazložena ocjena prihvatljivosti. Na samom dnu testa navedeno je tko je i kada proveo testiranje.

|  |  |  |  |
| --- | --- | --- | --- |
| **Testiranje prihvatljivosti** | | | |
| **Redni broj** | **Kriterij** | **Prihvatljivo / neprihvatljivo** | **Komentar** |
| **1.** | Registracija korisnika | **Prihvatljivo** | Omogućeno je registriranje više korisnika. Podaci za registraciju imaju dobro definirana ograničenja (format maila, broj znakova za lozinku) |
| **2.** | Prijava korisnika | **Prihvatljivo** | Kod svakog pokretanja aplikacije potrebna je prijava korisnika. Postavljene su provjere unosa podataka. |
| **3.** | Prikaz serija | **Prihvatljivo** | Na početnom fragmentu prikazuju se serije prema populatnosti. Navedeni su osnovni podaci o seriji i omogućeno dodavanje u favorite, kao i brisanje iz istih. |
| **4.** | Prikaz detalja o serijama | **Prihvatljivo** | Prikazani su svi potrebni detalji o seriji i uključen je prikaz trailera. Potrebno dodatno poraditi na dizajnu samog fragmenta. Prikaz detalja o serijama omogućen je sa sva tri fragmenta na kojima se serije pokazuju. |
| **5.** | Prikaz Favorita | **Prihvatljivo** | Postoji prikaz serija favorita. Oni su pohranjeni u bazu podataka. Moguće je uklanjanje favorita. Nakon uklanjanja potrebno je prebaciti fragment kako bi se osvježio. |
| **6.** | Prikaz preporuka serija | **Prihvatljivo** | Postoji prikaz preporučenih serija. Prikaz se vrši temeljem žanorova serija pohranjenih u favorite. Serije koje su spremljene u favorite ponavljaju se na popisu preporuka. Popis preporuka ograničen je na 20 serija. |
| **7.** | Notifikacije | **Prihvatljivo** | Implementirane su notifikacije. U izborniku postavke korisnik može odabrati koliko minuta prije početka serije želi dobiti obavijest o njezinom emitiranju. Dolazak notifikacija nije moguć ako je aplikacija skroz ugašena (ne radi u pozadini). |
| **8.** | Način dolaska notifikacija | **Prihvatljivo** | Korisnik u postavkama odabire način obavjesti o novoj notifikaciji. Ima na izbor zvuk ili vibraciju. |
| **Datum testiranja: 31.01.2017.** | | | |
| **Testiranje proveo: Goran Čehulić** | | | |