Univerzitet u Kragujevcu

Fakultet inženjerskih nauka

Seminarski rad iz predmeta
Softverski inženjering

Tema:
Bioskop

PROFESOR: **STUDENT:**

Prof. Dr. Nenad Filipović

Đorđe Karišić 657/2019

_____

Kragujevac, . . 2022.

Sadržaj

1. Postavka zadatka i detaljan opis aplikacije 4
1.1. Definisanje zadatka 4
1.2 Opis korišćenja aplikacije 4
2. Opis delova programa 5
2.1. Uvod 5
2.1.1. Tkinter 5
2.1.2. Pandas 5
2.1.3. Numpy 5
2.1.4. Twilio 6
2.1.5. OS 6
2.1.6. RE 6
2.1.7. DateTime 6
2.1.8. IMDbPy 6
2.1.9. URLlib 6
2.1.10. WebBrowser 6
2.1.11. IO 6
2.1.12. Pillow PIL 6
2.2. Datoteka **interface.py** 7
2.3. Datoteka **korisnickiInterfejs.py** 7
2.4. Datoteka **film.py** 7
2.5. Datoteka **bioskopska_sala.py** 7
2.6. Datoteka **korisnik.py** 7
2.7. Datoteka **movieInfoMenu.py** 7
2.8. Datoteka **rezervacijaMenu.py** 8
2.9. Datoteka **termin.py** 8
2.10. Datoteka **zaposleni.py** 8
2.11. Datoteka **osoba.py** 8

2.12. Datoteka **admin.py** 8
2.13. Datoteka **adminMenu.py** 8
3. UML Dijagrami 9
3.1. Dijagram slučajeva korišćenja 9
3.2. Dijagram klasa 10
3.3. Dijagram sekvenci 11
3.4. Dijagram aktivnosti 12
3.5. Dijagram stanja 13
4. Moguća poboljšanja aplikacije 14
5. Uputstvo za korišćenje 14
6. Literatura 15

1. Postavka zadatka i detaljan opis aplikacije

1.1. Definisanje zadatka

Osnovni zadatak ovog projekta jeste projektovanje sistema koji simulira rad bioskopa. Potrebno je jasno definisati tipove korisnika, njihove permisije, i moguće aktivnosti. Pored korisnika, neophodno je napraviti preciznu hijerarhiju elemenata bioskopa- filmove, njihove termine, i sale u kojima se oni prikazuju, ovaj zahtev sa sobom nosi potrebu za adekvatnom, uređenom bazom podataka kako bi sve CRUD operacije nad elementima iznad bile delotvorne i tačne. Na kraju, potrebno je obezbediti uredan, estetski zadovoljavajuć interfejs koji je takođe jednostavan za korišćenje, kako bi zaposlenima olakšali posao, i korisnicima pružili prijatno iskustvo.

1.2 Opis korišćenja aplikacije

Bioskop bi trebalo da obezbedi i zadovoljava standarde tri tipa korisnika:

- Zaposleni
- Običan korisnik
- Administrator

Zaposleni će predstavljati korisnika koji ovu aplikaciju koristi kako bi manipulisao filmovima, terminima, i pregledao sve informacije koje su relevantne za obezbeđivanje rada jednog bioskopa. Jedan zaposleni ima potpunu kontrolu nad svim filmovima, i terminima u kojima se oni mogu projektovati.

Običan korisnik je korisnik koji može rezervisati određeni broj mesta za određeni film u određenom terminu, i u određenoj sali. Ovaj tip korisnika ima potpunu kontrolu nad svojim rezervacijama, što uključuje brisanje, dodavanje i pregledanje.

Tip administrator je tip čiji se interfejs, i pristup aplikaciji najviše razlikuju. Ovaj tip ima moć brisanja i dodavanja isključivo zaposlenih korisnika, i dodavanja nove sale. Iz razloga što prilikom izgradnje jednog bioskopa se unapred zna koliko sala će (minimalno) postojati u okviru njega, ovom korisniku nije potrebna opcija za uništavanje sale, već samo za dodavanje nove.

Tipovi zaposleni i običan korisnik mogu podesiti svoj nalog, promeniti šifru i korisničko ime, dok tip administrator koristi isključivo jednu kombinaciju za pristup aplikaciji, koja je definisana van baze podataka, kako bi u slučaju pada baze podataka i gubitka informacija administrator mogao da pristupi aplikaciji.

2. Opis delova programa

2.1. Uvod

Za izradu ovog projekta biće korišćeno razvojno okruženje Visual Studio Code, i programski jezik Python sa sledećim dodatnim bibliotekama:

- Tkinter
- Pandas
- Numpy
- Twilio.REST
- OS
- RE
- Datetime
- IMDbPy
- URLlib
- WebBrowser
- IO
- PIL (Pillow)

2.1.1. Tkinter

Ova biblioteka omogućava izradu i stilizovanje interfejsa i predstavlja industrijski standard što se tiče projektovanja GUI-ja. U ovom projektu, ova biblioteka pruža lako manevrisanje između prozora, jednostavno i uredno pakovanje Widgeta na roditeljski prozor, i olakšano kombinovanje elemenata. Mana ove biblioteke jeste sporo očitavanje Top Window-a, što se može primetiti na prozoru Dodatne Informacije u okviru klijentskog interfejsa.

2.1.2. Pandas

Biblioteka Pandas nam pruža mogućnost potpune i precizne kontrole baza podataka. Uz pomoć funkcija ove biblioteke se lako spajaju, menjaju, dodaju i prikazuju tabele baze ovog projektnog zadatka.

2.1.3. Numpy

Retko korišćena biblioteka u ovom projektu, koristi se u kombinaciji sa bibliotekom Pandas zarad lakše simulacije upita.

2.1.4. Twilio

Uz pomoć ove biblioteke obaveštavamo korisnika o potvrđenim rezervacijama mesta u bioskopu. U ovom projektu je korišćena Trial verzija ove biblioteke, koja dozvoljava slanje poruke samo na unapred verifikovanim brojevima telefona.

2.1.5. OS

Standardna Python biblioteka za korišćenje sistemskih funckija. U ovom projektu, ova biblioteka čuva u sistemskom okruženju potrebne podatke za prijavu i autentifikaciju na Twilio REST API.

2.1.6. RE

Ovaj modul pruža mogućnost korišćenja regularnih izraza, koje se u ovom projektu koriste za proveru validnosti broja telefona.

2.1.7. DateTime

Kako bi se obezbedilo kreiranje jedinstvenih kodova rezervacije, ova biblioteka igra veliku ulogu u ovom projektu.

2.1.8. IMDbPy

IMDbPy koristimo kako bi olakšali zaposlenom unos parametara jednog filma i omogućili prikaz dodatnih informacija korisniku za neki film. IMDbPy nalazi i upravlja podacima sa IMDb baze podataka.

2.1.9. URLlib

Paket koji sadrži više modula koji rade sa URL-jevima. U ovom projektu najčešće korišćen modul je **urllib.request**.

2.1.10. WebBrowser

Ovaj modul obezbeđuje interfejs za prikaz web dokumenata korisnicima. Koristi se uz URLlib za generisanje trejlera nekog filma iz baze podataka.

2.1.11. IO

Modul za obradu raznih I/O elemenata. U ovom slučaju, pomoću ovog modula pretvaramo sliku da bude u *raw* formatu, tj. u obične bajtove koji će biti u memoriji, kako bi kasnije mogli da je očitamo u programu bez dodavanja u bazu podataka.

2.1.12. Pillow PIL

U kombinaciji sa prethodnim modulom se koristi za procesiranje slika, podržava dosta formata, i pruža mogućnost da prikažemo sliku bez dodavanja u bazu podataka.

2.2. Datoteka **interface.py**

Ova datoteka sadrži ekrane za logovanje, registraciju i interfejs zaposlenog lica. Ova datoteka se razlikuje od ostalih, iz razloga što ne zavisi od ostalih klasa, napisana je u drugačijem stilu.

2.3. Datoteka **korisnickiInterfejs.py**

Ova datoteka sadži klasu koja predstavlja interfejs običnog korisnika, kao i sve funkcije koje on može obavljati. Jedan korisnik može rezervisati mesto za neki film, pregledati dostupne filmove, I pregledati svoje rezervacije.

2.4. Datoteka **film.py**

U ovoj datoteci se nalazi klasa Film sa svim svojim funkcijama za brisanje, dodavanje, izmenu i vraćanje podataka filmova. Klasa u ovoj datoteci zahteva samo unos imena, a biblioteke i algoritam će uraditi sve ostalo kako bi izvukle potrebne informacije o filmu.

2.5. Datoteka **bioskopska_sala.py**

Ova datoteka je, nalik prethodnoj, zaslužna za sve operacije nad i predstavljanje bioskopske sale. Sadrži klasu BioskopskaSala. Jedna bioskopska sala sadrži svoj ID i broj mesta koji se nalazi u njoj.

2.6. Datoteka **korisnik.py**

Za predstavu korisnika, njegovih atributa i osobina, koristi se ova datoteka. Eksport i import, izmena osobina, brisanje i dodavanje korisnika su sadržani u klasi Korisnik, koja se nalazi u ovoj datoteci. Opisuje jednog korisnika sa svim neophodnim atributima i metodama.

2.7. Datoteka **movieInfoMenu.py**

Ova datoteka sadži klasu MovieInfo. Uz pomoć ove klase, možemo dodati novi Top Level Window i prikazati više informacija o nekom filmu, sliku, plot, glavnog glumca, režisera...

2.8. Datoteka **rezervacijaMenu.py**

Ova datoteka sadrži klasu Rezervacija. Uz pomoć klase Rezervacija možemo prikazati korisniku podatke za izabrani film, njegove termine i ID sale u kojima se prikazuje, i mogućnost rezervacije filma nakon izbora termina. Jednostavan sistem za rezervaciju nekog termina od strane korisnika.

2.9. Datoteka **termin.py**

Jednostavna datoteka koja sadrži klasu Termin sa svim njenim atributima i poljima. Upis i čitanje u ili iz tabele termin.txt je takođe definisano u ovoj datoteci.

2.10. Datoteka **zaposleni.py**

Datoteka zaposleni.py sadrži klasu Zaposleni, sve neophodne funkcije za dodavanje, brisanje, izmenu nekog objekta ove klase, kao i par funkcija za operaciju nad filmovima. Jedan zaposleni može dodavati, brisati filmove i termine, i menjati svoj profil.

2.11. Datoteka **osoba.py**

Datoteka osoba.py nam služi kako bismo napravili generalizaciju klasa Zaposleni i Korisnik. Sadrži pojedine zajedničke metode koje, generalizovane, omogućavaju zaposlenom ili osobi da pristupe aplikaciji, kao osobe. Koristi se prilikom login-a i registracije neke osobe.

2.12. Datoteka **admin.py**

Ova datoteka služi za predstavljanje specijalnog korisnika Admin. Admin nema login podatke sačuvane u bazama podataka, iz razloga što, u slučaju pada baze, može doći do nemogućeg pristupa aplikaciji. Admin se sistemski pamti kao kombinacija username-a admin i šifre administrator.

2.13. Datoteka **adminMenu.py**

Datoteka adminMenu.py ima zadatak da obezbedi sve operacije koje su potrebne jednom administratoru. Opcija za uklanjanje sala je izbačena, iz razloga što sala bioskopa ne može da nestane, i ako sala nije u funkciji, dovoljno je da zaposleni ukloni sve njene termine. Administrator jedino može da dodaje zaposlene korisnike, i to je jedini način kako se oni mogu dodati.

3. UML Dijagrami

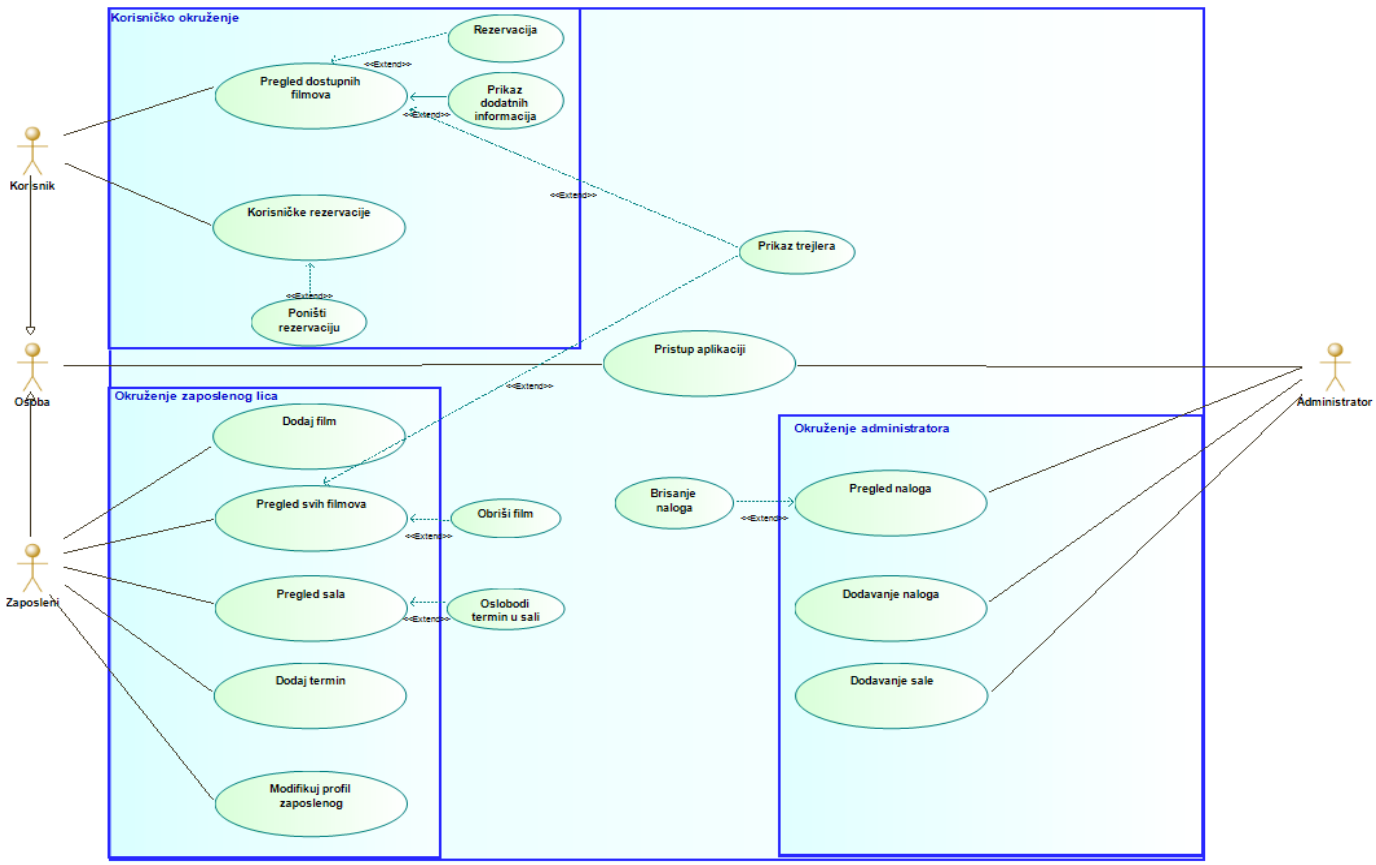
The Unified Modeling Language ili skraćeno UML je standardni grafički jezik za modelovanje objektno-orijentisanog softvera.

Pomoću ovog jezika biće prikazani sledeći dijagrami:

- Dijagram slučajeva korišćenja
- Dijagram klasa
- Dijagram sekvenci
- Dijagram aktivnosti
- Dijagram stanja

3.1. Dijagram slučajeva korišćenja

Dijagram slučajeva korišćenja je prikaz interakcije korisnika sa sistemom koji pokazuje odnos između korisnika i različitih slučajeva korišćenja u kojima je korisnik uključen. Slučajevi korišćenja predstavljeni su krugovima ili elipsama, a korisnici (akteri) figurama čoveka.



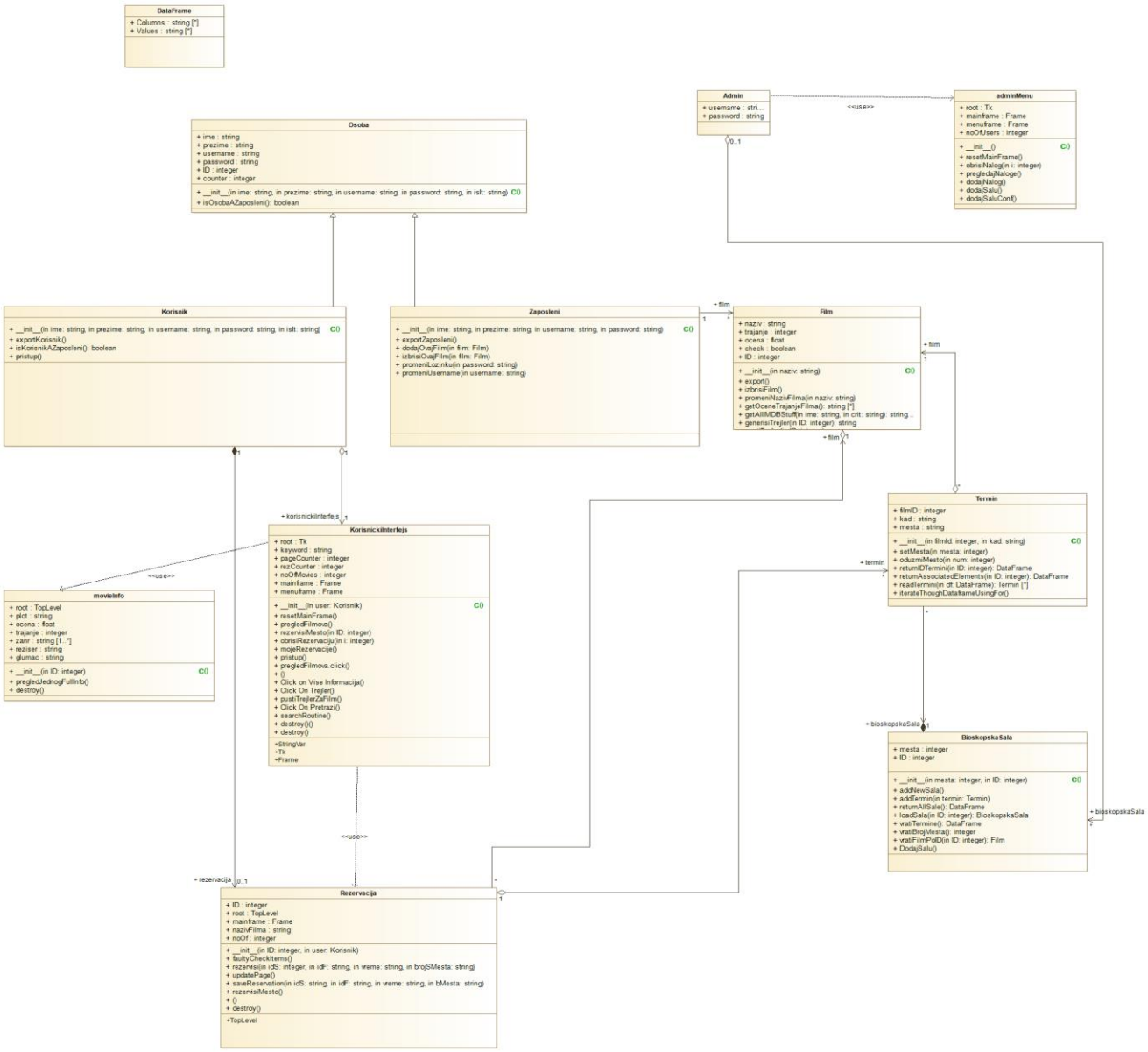
Ilustracija 1, Use Case dijagram aplikacije

3.2. Dijagram klasa

Dijagram klasa je vrsta strukturnog dijagrama u softverskom inženjeringu, koji opisuje strukturu sastava objašnjavajući klase unutar sastava, njihove atribute i odnose.

Elementi dijagrama klasa su:

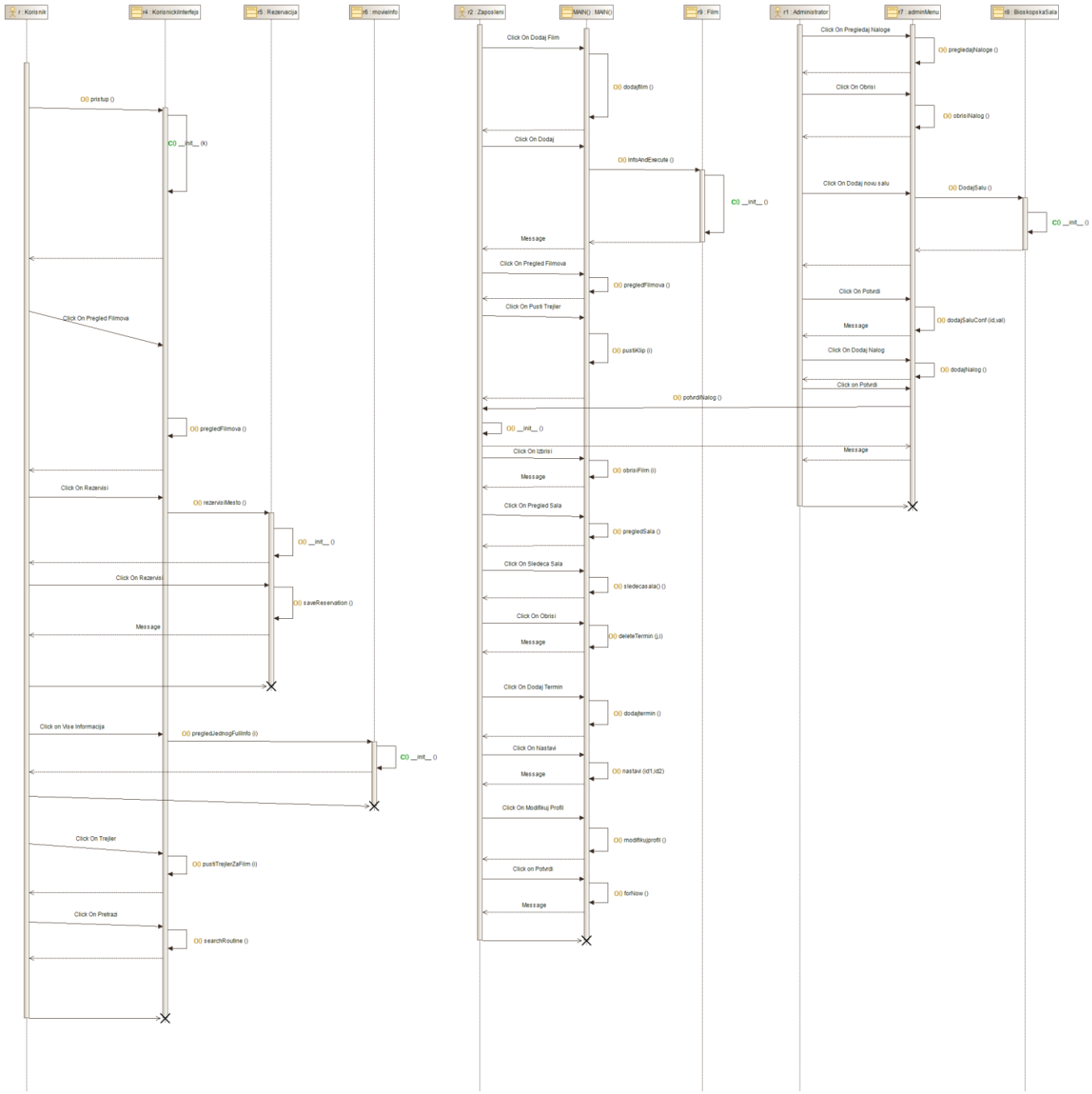
- stvari: klasa, interfejsi, tipovi, izuzeci, šabloni, saradnje, paketi
- relacije: zavisnosti, generalizacije, asocijacije, realizacije



Ilustracija 2, Dijagram klasa aplikacije

3.3. Dijagram sekvenci

Dijagram sekvenci prikazuje komunikaciju između skupa objekata, koja se ostvaruje porukama koje objekti međusobno razmenjuju u cilju ostvarivanja očekivanog ponašanja. Dijagram sekvenci može da sadrži aktere, objekte i poruke.

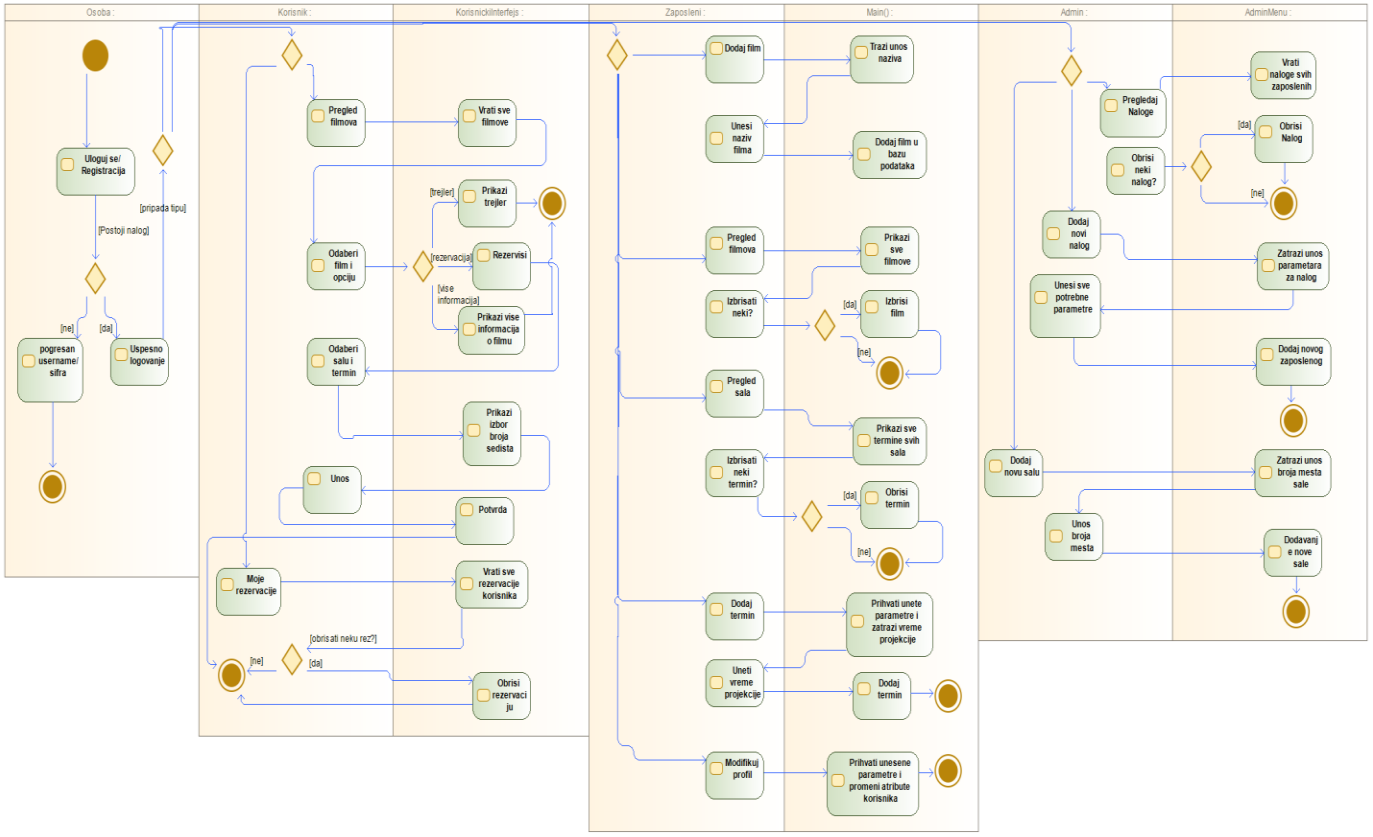


Ilustracija 3, Dijagram sekvenci aplikacije

3.4. Dijagram aktivnosti

Dijagrami aktivnosti su namenjeni modeliranju dinamičkih aspekata (ponašanja) sistema. Dijagram aktivnosti prikazuje:
- tok aktivnosti koju izvršavaju objekti;
- eventualno i tok objekata između koraka aktivnosti.

Aktivnost je specifikacija parametrizovanog ponašanja koje se izražava kroz tok izvršenja preko sekvenciranja i konkurisanja podaktivnosti. Dijagram aktivnosti je graf koji sadrži čvorove i grane.



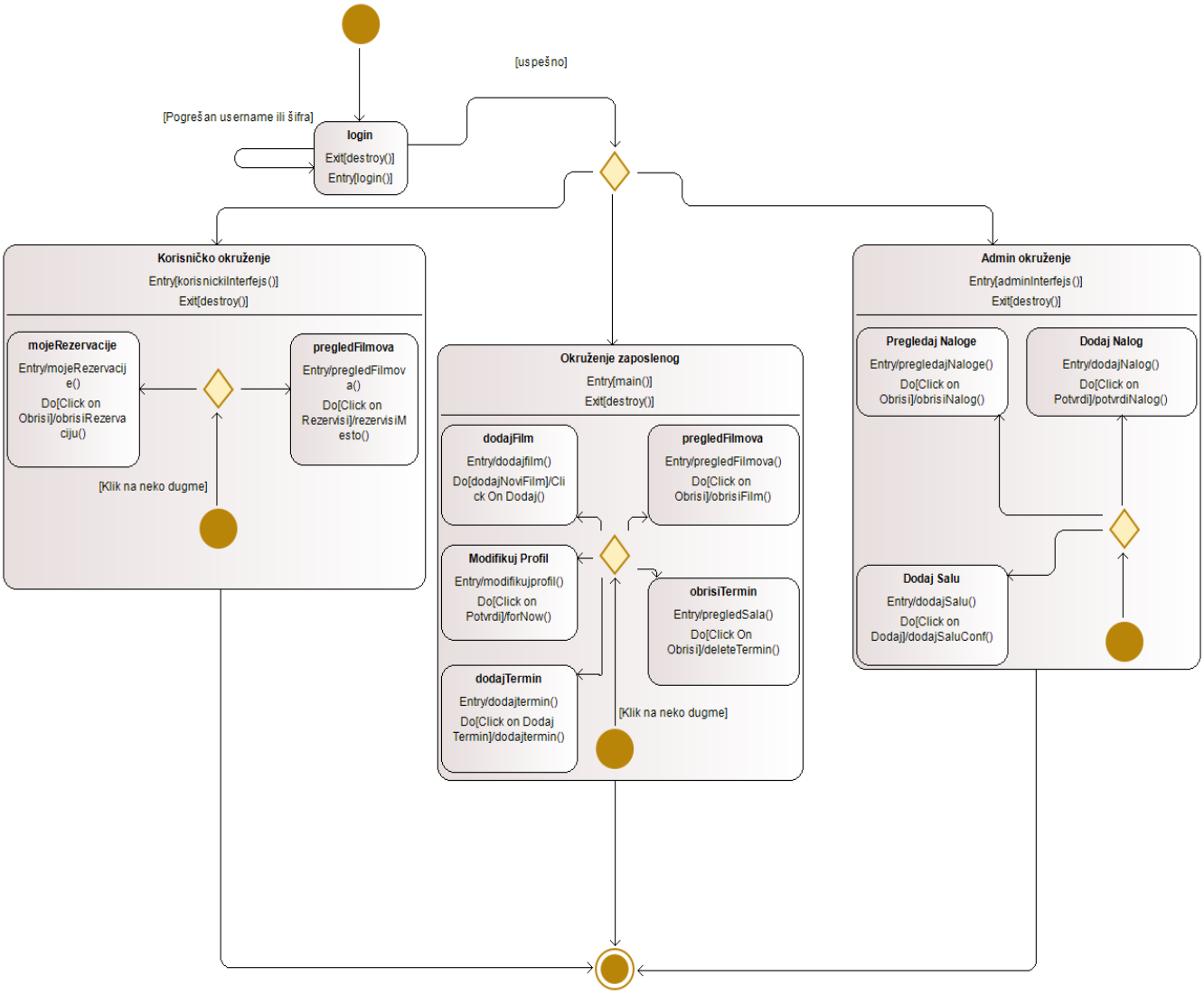
Ilustracija 4, Dijagram aktivnosti aplikacije

U ovom dijagramu su korišćene plivačke staze.

Plivačke staze (swimlanes) specificiraju odgovornosti za delove celokupne aktivnosti. Staza reprezentuje neki subjekat odgovoran za sprovođenje akcije.

3.5. Dijagram stanja

Dijagram stanja se koristi za opisivanje ponašanja sistema.
On može da opiše moguća stanja objekta kako se događaji pojavljuju.
Svaki dijagram obično predstavlja objekte jedne klase i prati različita stanja tih objekata kroz sistem.
Dijagram stanja se može upotrebiti da grafički predstavi automate konačnih stanja.
Stanje se označava pravougaonikom sa zaobljenim ivicama.



Ilustracija 5, Dijagram stanja aplikacije

4. Moguća poboljšanja aplikacije

U datoteci movieInfoMenu.py, moguće je koristiti multithreading kako bi se izbeglo duže čekanje prozora koji bi trebalo prikazati. Glavna nit može postati opterećena očitavanjem TopLevel prozora i pozivom IMDbPy funkcija, pogotovo očitavanjem slike.

U prethodno navedenoj datoteci, može se sačuvati prethodno pronađena slika u bazi podataka, kako bi se smanjilo vreme čekanja prozora, umesto da se svaki put generiše, isto se može uraditi za link trejlera za film.

Iz razloga što ova aplikacije koristi lokalnu bazu podataka i određen broj funkcija koje zahtevaju pristup internetu, moguće je modifikovati aplikaciju tako da ima offline verziju koja bi imala samo osnovne funkcije, i ograničiti korisnika koji nema pristup internetu da koristi iskljućivo tu verziju.

Moguće je dodati Exception-e za slučaj kada korisnik nije povezan na internet, kako bi bio obavešten da ne može koristiti neke od dodatnih funkcija koje zahtevaju internet. Primer bi bio poziv Tk.MessageBox koji bi prenosio poruku obaveštenja da je nemoguće koristiti odabranu funkciju.

Zbog postojanja više GUI datoteka i klasa koje imaju slične atribute, moguće je napraviti nadklasu, kao za Osoba-Korisnik(Osoba) i Osoba-Zaposleni(Osoba) koja bi obuhvatila sve atribute i funkcije koje ove klase dele.

Moguće je napraviti pojedina estetska poboljšanja, npr. napraviti dark mode opciju, dodati animacije,widgete...
Prvobitna verzija programa je koristila Media Player kako bi pustila trejler, ukoliko je moguće obezbediti multithreading, ovaj pristup bi bio optimalan.

Bolji pristup generisanju trejlera filma je korišćenje biblioteke BeautifulSoup, koja služi za web scraping.

5. Uputstvo za korišćenje

Kako bi se pokrenuo program, potrebno je pokrenuti interface.py datoteku. Prvenstveno, biće pokrenut prozor za registraciju, sa kojeg je moguće preći na login prozor.

6. Literatura

https://pypi.org/project/IMDbPY/ 04.05.2022. (15:43)

https://www.tutorialspoint.com/python/tk_toplevel.htm 04.05.2022. (16:25)

https://www.geeksforgeeks.org/python-parse-a-website-with-regex-and-urllib/

10.05.2022. (18:40)

https://www.twilio.com/ 10.06.2022. (11:30)

MOODLE PORTAL, Softverski inženjering http://moodle.fink.rs/ 22.06.2022. (13:10)