

Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Seminarski rad iz predmeta Softverski inženjering

Tema:
Bankomat(ATM)

Student:
Mladen Cvetkovic 649/2019

Predmetni profesor:
Prof. Dr. Nenad Filipović

Kragujevac 2022.

1.1.1 Sadržaj:

1.1.1 Sadržaj:.....	1
1.1 Definisanje zadatka	1
1.2 Opis korišćenja aplikacije	1
2.1 Uvod.....	2
2.1.1 Tkinter	2
2.1.2 OS.....	2
2.1.3 Time.....	3
2.1.4 Random	3
2.1.5 Mysql.connector	3
2.1.6 PIL(Pillow)	3
2.2 Datoteka atm_projekat.py	3
2.3 Datoteka user.py.....	3
2.4 Datoteka stringHandler.py.....	4
3. UML Dijagrami.....	4
4 Uputstvo za korišćenje	9
5. Literatura	9

1.1 Definisanje zadatka

Osnovni zadatak ovog projekta jeste projektovanje sistema koji simulira rad bankomata(atm). Potrebno je jasno definisati korisnika, njegove permisije, i moguće aktivnosti. Pored korisnika, neophodno je napraviti precizan sistem po kojem funkcioniše sam bankomat. Samim tim imamo potrebu za adekvatnom, uređenom bazom podataka kako bi sve 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 korisnicima olakšali posao, i pružili prijatno iskustvo..

1.2 Opis korišćenja aplikacije

Kada otvori aplikaciju, korisnik mora da se loguje odnosno registruje ukoliko nema napravljen nalog. Zatim kada se uloguje ulazi u start meni gde ima 3 opcije I to je da pristupi programu, da izađe iz njega ili da ukoliko želi može promeniti nalog (ako ih ima više).

Kada uđe u glavni meni ima sledeće opcije a to su da: Podigne novac, uplati novac, dobije informacije o svom nalogu, da odštampa izveštaj, da proveriti stanje racuna i dugme za izlaz na startni meni. Svaka od prethodnih opcija ima svoj sopstveni meni unutar koga korisnik može da odradi već definisane operacije.

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
- OS
- Time
- Random
- Mysql.connector
- 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.

2.1.2 OS

Standardna Python biblioteka za korišćenje sistemskih funkcija. U ovom projektu, ova biblioteka se koristila kako bi se postavili putanju na folder gde se nalaze slike.

2.1.3 Time

Ova biblioteka se koristi kako bi smo odredili tacno vreme prilikom stampanja izvestaja kao i trenutno vreme koje stoji u bottom frejmu.

2.1.4 Random

Ovu biblioteku smo koristili kako bi smo pri registraciji dodelili korisniku neki random serijski broj.

2.1.5 Mysql.connector

Ovu biblioteku smo koristili kako bi smo nas projekat povezali sa mysql databazom na koju ce se cuvati svi podaci o korisnicima.

2.1.6 PIL(Pillow)

Ovu biblioteku smo koristili kako bi smo procesirali slike, podržava dosta formata, i pruža mogućnost da prikazemo sliku bez dodavanja u bazu podataka. Za izradu slika koriscen je program Adobe Express.

2.2 Datoteka atm_projekat.py

U ovoj datoteci se sadrzi skoro sve I to od ekrana za logovanje/registraciju pa sve do ekrana za podizanje, uplatu, stampanje izvestaja, informacija o korisniku, glavnog menija... Ona nam je glavni deo projekta I ostale datoteke su includovane u nju. U njoj su definisane klase za svaki Page tj za svaki ekran I razne funkcije koje se pozivaju prilikom koriscenja određenih buttona.

2.3 Datoteka user.py

U ovoj datoteci se sadrzi klasa korisnik koja nam definise samog korisnika. U njoj smo takodje definisali funkcije za withdraw, deposit, getInfo, getBalance, chechIfExists (za kreiranje databaze), addUserToDb (za dodavanje korisnika u databazu), PrintInfo (za stampanje izvestaja) itd...

2.4 Datoteka stringHandler.py

Usluzna datoteka koje se koristila kako bi se dekodovao bytestring u string. Ne treba da se koristi vise.

3. UML Dijagrami

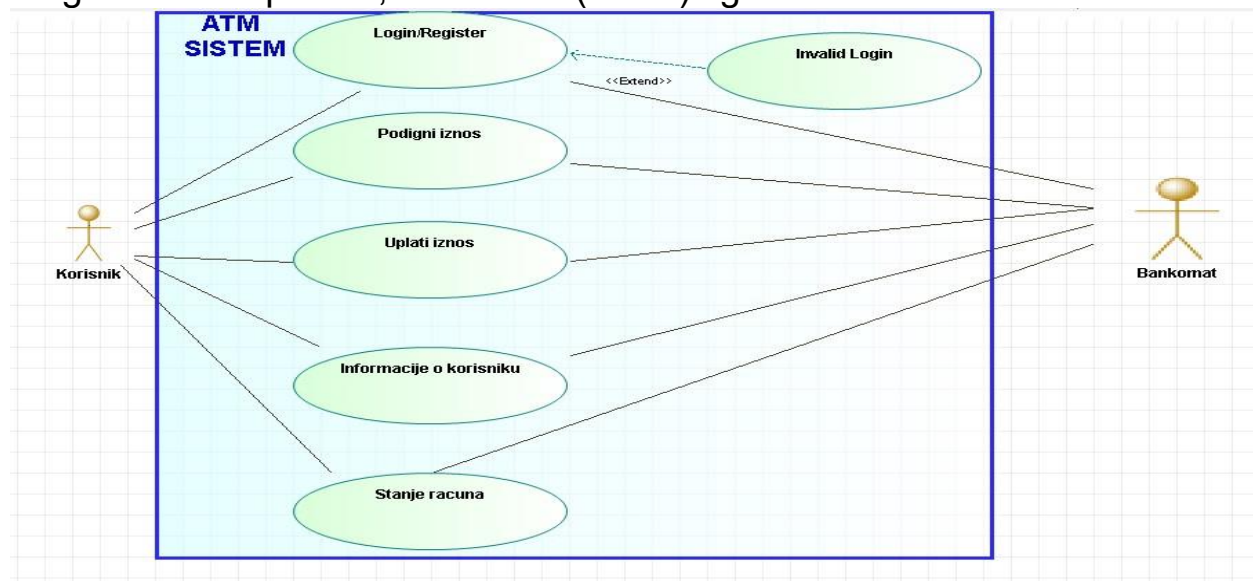
The Unified Modeling Language ili skraćeno UML je standardni grafički jezik za modelovanje objektno-orijentisanog softvera.

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.



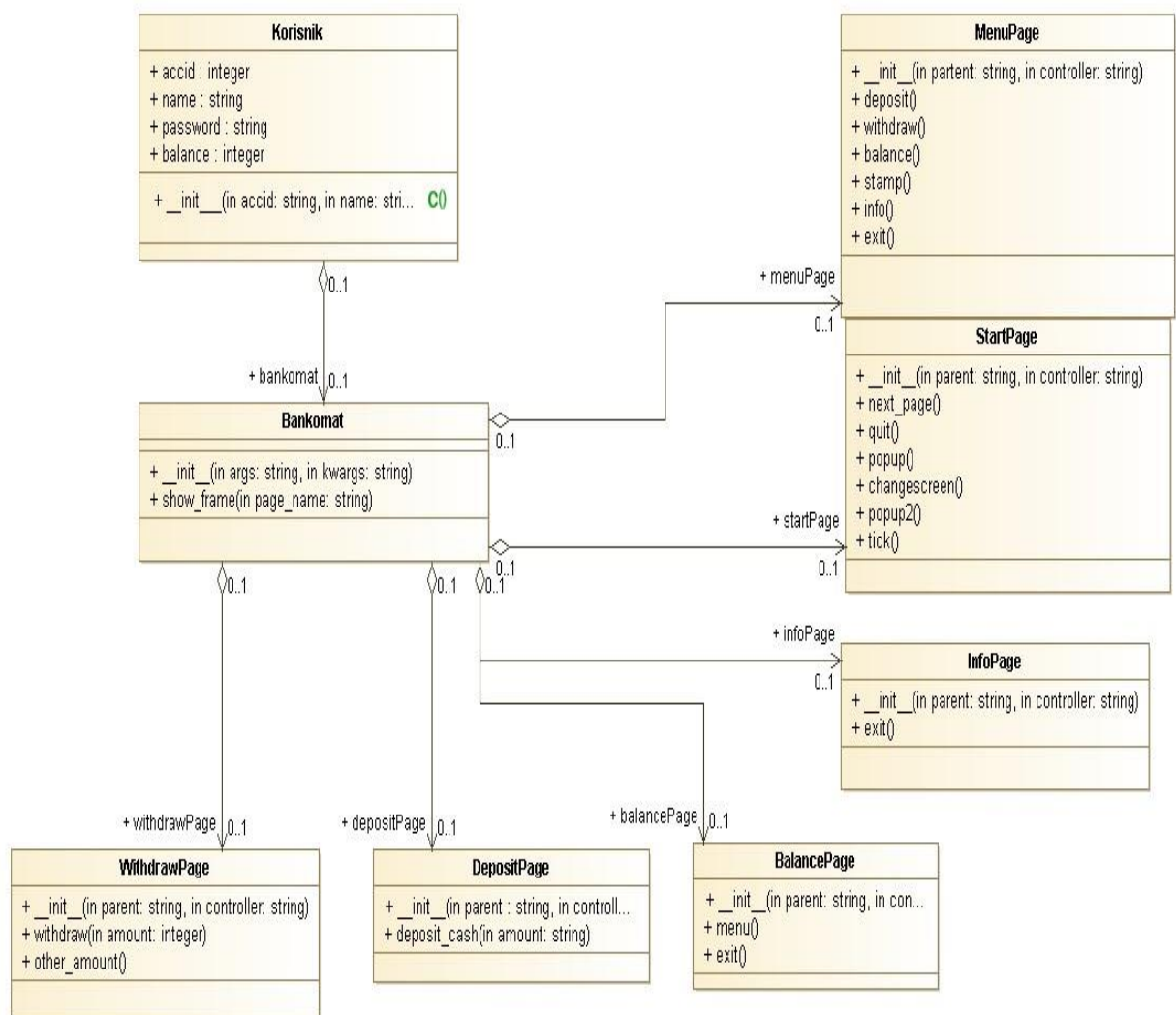
Slika 1, Use Case dijagram

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:

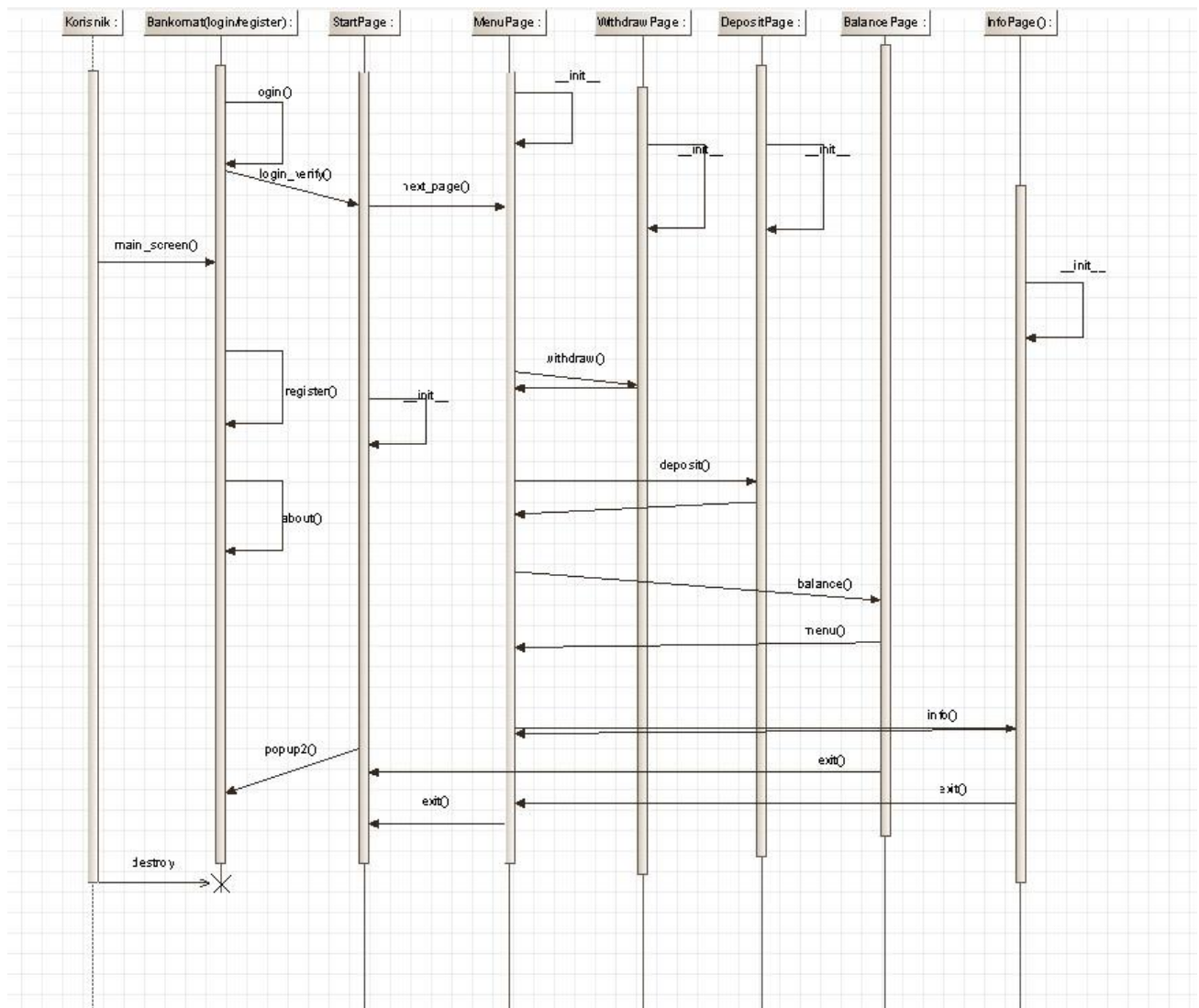
- stvari: klasa, interfejsi, tipovi, izuzeci, šabloni, saradnje, paketi
- relacije: zavisnosti, generalizacije, asocijacije, realizacije



Slika 2, Dijagram klasa

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.



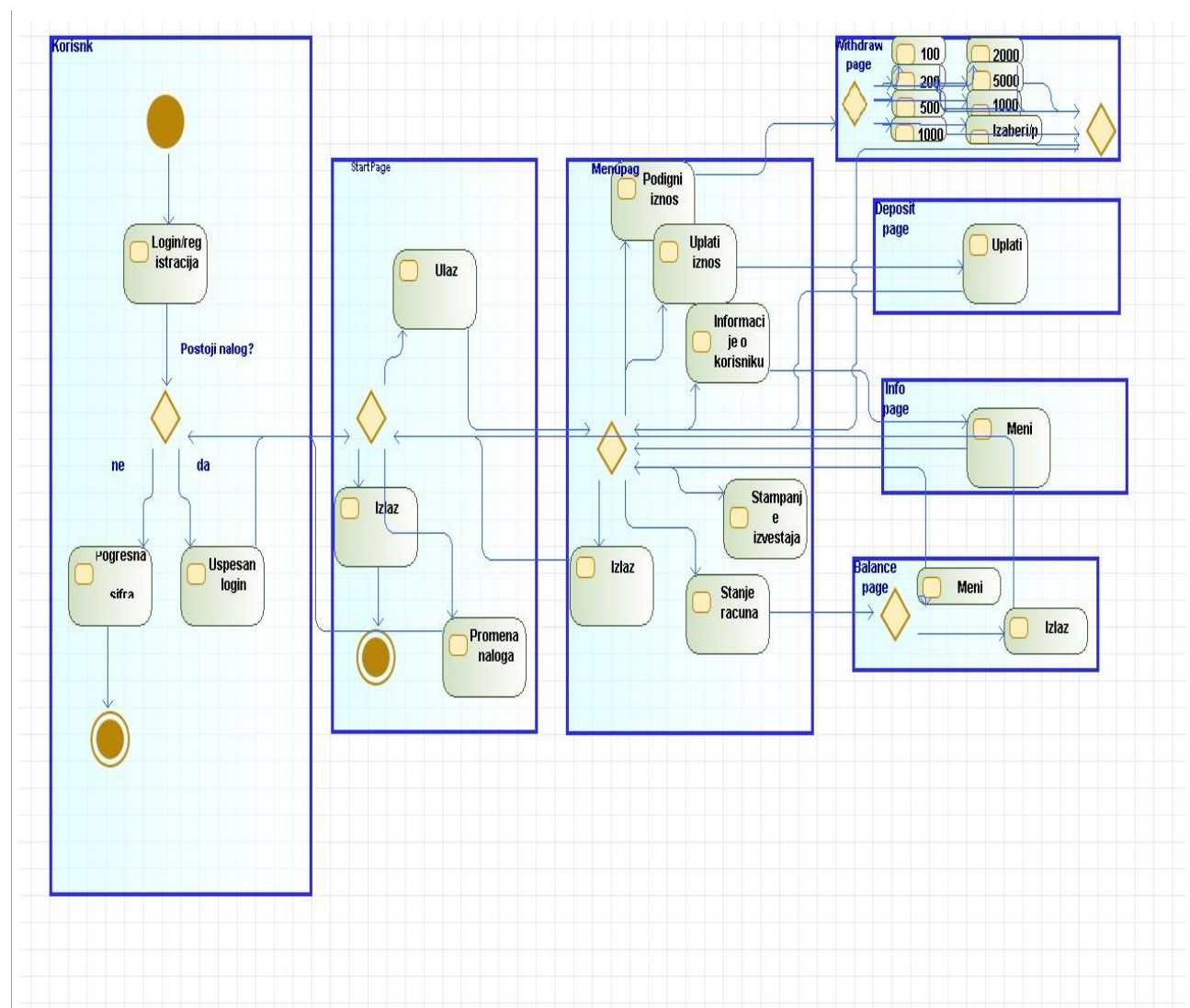
Slika 3, Dijagram sekvenci

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.

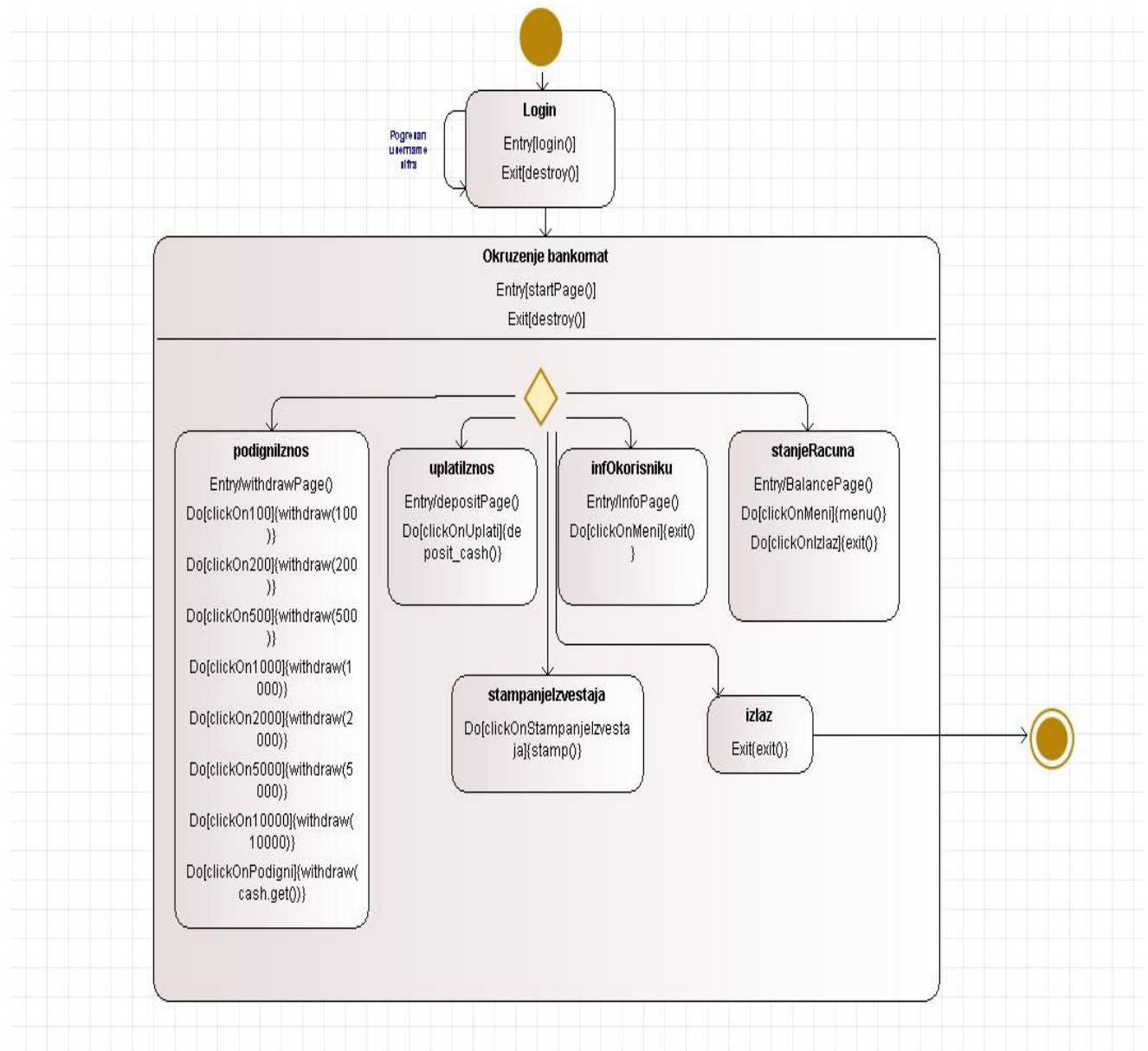
Aktivnost je specifikacija parametrizovanog ponašanja koje se izražava kroz tok izvršenja preko sekvenciranja i konkurisanja podaktivnosti. Dijagram aktivnosti je graf koji sadrži čvorove i grane.



Slika 4, Dijagram aktivnosti

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.



Slika 5, Dijagram stanja

4. Uputstvo za korišćenje

Uputstvo za korišćenje Kako bi se pokrenuo program, potrebno je pokrenuti `atm_projekat.py` datoteku. Biće pokrenut prozor za login/registraciju sa kojeg ce te dalje pristupiti glavnom meniju.

5. Literatura

<https://docs.python.org/3/library/os.html>

<https://docs.python.org/3/library/tkinter.html>

<https://pillow.readthedocs.io/en/stable/>

<https://dev.mysql.com/doc/connector-python/en/>

<http://moodle.fink.rs/>