

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku  
Principi softverskog inženjerstva (13S113PSI)



## *Projektni zadatak*



**STRATEGY  
PLAYGROUND**

Studenti:

Nikola Đokić, 2018/0128

Luka Lazović, 2018/0545

Danilo Mandušić, 2019/0607

Ana Vraneš, 2017/0562

# Sadržaj

1. Uvod.....	3
1) Rezime.....	3
2) Namena dokumenta i ciljne grupe.....	3
2. Opis problema.....	3
1) O spekulativnoj trgovini i njenoj automatizaciji .....	3
2) Backtesting simulacija.....	3
3) Opis programskog interfejsa za rad sa tržištem.....	4
4) Opis korisničkog interfejsa za rad sa tržištem.....	4
3. Kategorije korisnika.....	5
1) Gost.....	5
2) Registrovani korisnik.....	5
3) Administrator .....	5
4. Opis proizvoda .....	5
1) Arhitektura sistema.....	5
2) Pregled karakteristika .....	6
5. Funkcionalni zahtevi.....	7
6. Pretpostavke i ograničenja .....	9
7. Nefunkcionalni zahtevi.....	9
1) Sistemski zahtevi.....	9
2) Ostali zahtevi.....	9
8. Zahtevi za korisničkom dokumentacijom .....	9
9. Plan i prioriteti .....	9
10. Istorija izmena.....	10

# 1. Uvod

## 1) Rezime

Projektni zadatak Strategy Playground je predstavlja zadatak Tima bez imena za praktičan deo nastave na predmetu principi softverskog inženjerstva. Kao završen zadatak, on će predstavljati veb platformu namenjenu ljudima koji se bave spekulativnom trgovinom na valutnom tržištu i njima će služiti kao mesto gde će, koristeći elementarno znanje programskog jezika python, moći da eksperimentišu sa raznim strategijama automatske trgovine.

## 2) Namena dokumenta i ciljne grupe

Ovaj dokument je namenjen da opiše ideje i suštinu projektnog zadatka, kategorije korisnika, jednostavan opis aplikativnog programskog interfejsa prema korisničkim programima, funkcionalne i druge zahteve i namenjen je članovima razvojnog tima zaduženog za izradu istog.

# 2. Opis problema

## 1) O spekulativnoj trgovini i njenoj automatizaciji

Spekulativna trgovina je jedan od najprimamljivijih načina zarade novca modernih vremena. Sam koncept mogućnosti pojedinca da predviđanjem nečega kao što je kretanje cene na nekom tržištu zajedno sa postojanjem likvidnih sredstava čije su cene po prirodi nestabilne i haotične kao što su decentralizovane kriptovalute čine ovu aktivnost jako blisku igrama na sreću (kockanju), što je i njena glavna osobina u očima velike većine populacije.

Međutim, ovde se radi o dosta složenijem sistemu koji nije čisto oslonjen na verovatnoću slučajnih ishoda kao što su klasične kockarske aktivnosti. Postoji more uočenih pravilnosti koje ljudi i grupe ljudi eksploatišu da bi zaradili. Neka ponašanja su dovoljno pravilna da se strategije trgovine koje eksploatišu ta pravila mogu dovoljno formalizovati da se otvaruje mogućnost da računar radi po definisanoj strategiji i uspeva da zarađuje. Slično čoveku, i program može da prati promene na tržištu i da reaguje na njih ako se dovoljno precizno definiše kako da reaguje.

## 2) Backtesting simulacija

Kada se osmisli strategija trgovine potrebno je testirati, kako bi bila pouzdana i kako bi korisnici stekli poverenje da bi uložili novac u nju. Prvi i jedini način testiranja je istorijsko testiranje tj. backtesting. Ono se zasniva na simuliranju tržišta u nekom unapred definisanom vremenskom periodu da bi se videlo kako program reaguje na promene cene. U zavisnosti od vremenskog perioda na kome se testiranje vrši, ova simulacija može da traje izuzetno dugo. Frekvencija pojedinačnih promena nije 100% konzistentna i može se popeti na red veličine više promena u sekundi. Platforma koja se specifikira u ovom dokumentu je zato i zasnovana da predstavlja interfejs preko koga registrovani korisnik može da testira testira svoje strategije bez opterećenja svog ličnog računara (u nastavku dokumenta će ovaj deo sistema biti zvan igralište) uz sva propratna uputstva i dokumentaciju za korišćene ovog sistema kao i mogućnost nadmetanja sa drugim korisnicima po uspehu osmišljenih strategija za pozicije na rang listama.

### 3) Opis programskog interfejsa za rad sa tržištem

Da bi korisnički programski kod mogao da interaguje sa simuliranim tržištem, on mora da se nalazi na serveru i da bude povezan preko nekog aplikativnog programskog interfejsa. Zbog jednostavnosti programiranja i jednostavnosti povezivanja radnog okvira sa korisničkim kodom, izabrani programski jezik za korisnički kod je Python. Jednostavnost povezivanja se ogleda u tome što Python kod nije potrebno prevoditi i korisnički kod se može povezati tako što se uveze kao modul za vreme izvršavanja.

Tržište se u ovom kontekstu može posmatrati kao generator događaja koji predstavljaju promene cene. Zbog toga je zamišljeno da se uspostavi programska interakcija u okviru koje se primenjuje projektni uzorak posmatrač. Subjekat je ovde samo tržište, tj. apstrakcija istog, a korisnička strategija je posmatrač. Posmatrač implementira proceduru gde reaguje na promene cene na tržištu, a radni okvir simulira promene na osnovu istorijskih podataka ili podataka koje korisnik zadaje i obaveštava korisničku strategiju o istim, uz svest o protoku vremena u simulaciji (svaki događaj uz sebe ima vreme kada se desio). Protok vremena se ovde uključuje pošto omogućavaju da program može da radi stvari kao što je trgovanje britanskom funtom samo u periodu kad rade banke u Velikoj Britaniji i slično. Strategija može dodatno da interaguje sa tržištem da bi radila stvari kao što su otvaranje i zatvaranje pozicija. Za pomoćne potrebe, u interfejs su uključene procedure reagovanja na događaje početka i kraja simulacije.

### 4) Opis korisničkog interfejsa za rad sa tržištem

Kao što je već spomenuto, korisnik treba da dostavi svoj kod na server kako bi on mogao biti simuliran. Osnovni način je da jednostavno razvije programski kod na svom računaru i pošalje datoteku sa kodom na server, što je jedan način koji sistem nudi.

Na višem nivou, korisniku je u interesu da strategije razbije na jednostavnije module koji su nekako organizovani pa sistem ovde nudi korisnicima da prave strategije koje bi odgovarale projektima u integrisanim razvojnim okruženjima. U okviru svake strategije, postoji hijerarhija direktorijuma i datoteka sa kodom. Korisnik može da koristi ugrađeni code editor na sajtu da menja postojeće datoteke direktno na serveru, kao i da ih briše, pravi prazne i grupiše u direktorijume. Prilikom simulacije strategije, radni okvir poštuje ovu hijerarhijsku strukturu datoteka i dozvoljava da moduli uvoze i zovu jedni druge. Prilikom pokretanja simulacije, korisnik zadaje modul koji će radni okvir koristiti da bi iz njega uvezao procedure opisane u poglavlju 2.3.

Pre pokretanja simulacije, korisnik bira i podatke o kretanju cene koji će biti simulirani. Podrazumevano su izabrani istorijski podaci o promeni cene koji su javno dostupne informacije, ugrađene u server i dostupne korisnicima za korišćenje. Zbog svoje prirode, promene cene u nekom vremenskom periodu su najobičnija lista gde svaki element predstavlja jednu promenu na tržištu i vreme kad se ta promena desila. Format liste je .csv fajl čiji je format javno dostupan da bi se korisnicima omogućilo da prave svoje liste promena koje mogu dostaviti serveru da ih koristi za simulaciju. Dostavljene liste događaja za simulacije se perzistentno pamte na nivou korisnika. Korisnik takođe može i da traži od sistema da mu nacrtava vizuelni prikaz svojih podataka kao i istorijskih podataka za neki valutni par u nekom vremenskom periodu za lakše uočavanje karakterističnih situacija.

Pored istorijskih podataka i podataka koje korisnik lično dostavi, on može i izvršiti simulaciju za rangiranje na nekoj rang listi. Rang liste imaju svoje liste događaja koje nisu vidljive korisnicima. Zadaće ih administrator prilikom pravljenja liste. Rang liste su namenjene korisnicima da vide kako se njihove strategije ponašaju u nepoznatom okruženju u poređenju sa drugim korisnicima da bi se podstaklo poboljšanje postojećih i izmišljanje novih strategija. Pregled postojećih rang lista je dostupan i bez autentifikacije ali učestvovanje u rangiranju nije.

Nakon što se simulacija pokrene, korisnik može da vidi njen status. Simulacija može biti u toku, prekinuta zbog greške ili završena. Prekinute i završene simulacije rezultuju relevantnim ispisima koji se sastoje od opisa greške i informacija o uspehu respektivno, kao i svih ispisa koje korisnički kod učini.

## 3. Kategorije korisnika

Sistem razlikuje 3 kategorije korisnika: gost, registrovani korisnik i administrator. Administrator i registrovani korisnik se moraju autentifikovati pre nego što pristupe funkcijama sistema koje zahtevaju odgovarajući tip korisnika. Svaki navedeni tip korisnika ima pristup svim funkcijama sistema koje imaju korisnici navedeni iznad njega.

### 1) Gost

Gost je korisnik koji nema svoj nalog u sistemu i time ima ograničene mogućnosti korišćenja sistema. On može pristupiti formi gde, uz unos traženih ličnih podataka, može poslati zahtev za registraciju. Ima i pristup formi za autentifikaciju kao i mogućnost da vidi postojeće rang liste na platformi i da čita uputstva i dokumentaciju za korišćenje igrališta.

### 2) Registrovani korisnik

Registrovani korisnik je korisnik čiji je zahtev za registraciju odobren od strane administratora. Kao takav, on ima pristup igralištu gde može da pristupi svojim sačuvanim strategijama, da ih menja kao i da pravi nove. Na igralištu može i da pokreće simulacije svojih strategija u okruženju koje simulira tržište koristeći stvarne istorijske podatke iz izabranog vremenskog perioda na izabranom valutnom paru kao i podatke koje lično pripremi za ispitivanje ponašanja strategije u specifičnim situacijama i **ivičnim slučajevima**.

### 3) Administrator

Administrator je poseban korisnik koji ima ulogu da odobrava zahteve za registraciju novih korisnika, kao i da kreira rang liste. On takođe ima mogućnost da dodaje nove i ima uvid u već postojeće rang liste kao i pristup podacima o svim registrovanim. Jedini način da neko postane administrator je da ga postojeći administrator učini istim. Administrator se ne može rangirati na rang listama.

## 4. Opis proizvoda

### 1) Arhitektura sistema

Sistem je zamišljen da bude veb platforma čija je pristupna tačka veb server. Ovaj server pravi veb stranice na kojima se prikazuju forme za pristup funkcionalnostima sistema kao i povratne informacije

koje su rezultat izvršavanja tih funkcionalnosti. Perzistentno skladištenje podataka o korisnicima, njihovim strategijama i rang listama se implementira u formi servera baze podataka u kojoj se čuvaju potrebne informacije s kojim bi komunicirao veb server, šaljući upite bazi i koristeći rezultate tih upita da formira prikaze informacija korisnicima. Za izradu veb servera je izabran radni okvir *Django*.

## 2) Pregled karakteristika

Korist za korisnika	Karakteristika koja je obezbeđuje
Nepostojanje ograničenih vremensih okvira u kojima korisnik pristupa funkcionalnostima sistema	Sistem je stalno aktivan i dostupan na internetu. Korisnik može da <b>gađa veb server</b> i koristi funkcionalnosti koje sistem nudi dok god poseduje internet konekciju.
Platformaska nezavisnost sistema sa kog korisnik pristupa funkcijama sistema	Interfejs za rad je zasnovan na HTML, CSS i Javascript tehnologijama i može se prikazivati u bilo kom modernom veb pretraživaču.
Smanjena potrošnja energije i potreba za procesorskom snagom na korisničkom uređaju	Sistem nudi uslugu simuliranja strategije na svom računaru, ne na klijentskom. Klijent ne mora da poseduje moćan računar da bi izvršavao simulacije svojih strategija na dugačkim <b>vremensim</b> periodima.
Sigurnost i poverljivost podataka	Korisnički podaci (lični podaci i sačuvane strategije) su zaštićeni i zahtevaju prethodnu autorizaciju da bi im se pristupalo. Sistem razlikuje administratora od običnog registrovanog korisnika i vrši proveru nivoa privilegija pre dozvoljavanja pristupa podacima.
Lako administriranje	Kao i sve ostale funkcionalnosti sistema, i administriranje se radi preko veb portala koji je dostupan nakon uspešne autentifikacije administratora. Nije potrebno da administrator bude fizički prisutan već je dovoljno da bude povezan na server preko interneta.

## 5. Funkcionalni zahtevi

### 1) Registracija korisnika

Svaki korisnik ima mogućnost da pristupi formi gde može da unese potrebne podatke i pošalje zahtev za registraciju.

### 2) Autentifikacija korisnika

Svaki korisnik koji nije to prethodno uradio može da pristupi formi gde unosi potrebne podatke za autentifikaciju. Obični registrovani korisnici i administratori se prijavljuju preko iste forme pri čemu je na serveru da utvrdi nivo privilegija na osnovu unetih podataka. Nakon uspešne autorizacije, korisniku se omogućava pristup funkcionalnostima kojima ima pristup.

### 3) Rad sa strategijama

Za rad sa strategijama je potrebno da registrovani korisnik prethodno izvrši autentifikaciju sebe.

#### i. Pregled strategija

Korisnik može da vidi strategije koje je prethodno napravio. Ovo podrazumevano vidi kada izvrši autentifikaciju

#### ii. Pravljenje nove strategije

Korisnik može da napravi novu strategiju tako što zada ime strategije koje mora biti jedinstveno na nivou liste postojećih strategija koje taj korisnik ima napravljenim.

#### iii. Brisanje postojeće strategije

Korisnik može da izabere neku postojeću strategiju i da je obriše iz svoje liste uz dodatnu potvrdu da želi to da uradi.

#### iv. Perzistentno pamćenje strategija

Korisnik može u bilo kom trenutku da sačuva trenutno stanje svojih strategija da mu se restaurira sledeći put kad otvori pregled strategija

#### v. Pregled jedne strategije

Korisnik može da izabere neku postojeću strategiju i dobije prikaz hijerarhije modula koji je čine.

#### vi. Dodavanje modula u strategiju

Za izabranu strategiju, korisnik može da uključi modul u istu tako što pošalje lokalni fajl na server ili stvori prazan fajl.

#### vii. Izmena postojećeg modula

Za izabranu strategiju, korisnik može da izabere jedan modul u okviru njene hijerarhije. Tada dobija prikaz code editor interfejsa sa sadržajem tog modula. Ima mogućnost da izmeni taj modul i opciju da sačuva izmene u modul.

#### viii. Brisanje modula iz strategije

Za izabran modul iz izabrane strategije, korisnik može da izda komandu za brisanje modula. Nakon što potvrdi da to zaista želi da radi, modul biva obrisano iz strategije.

#### 4) Rad sa podacima o tržištu

Za rad sa podacima o tržištu je potrebno da je registrovani korisnik prethodno izvršio autentifikaciju sebe tako što je pristupio formi za autentifikaciju i uneo ispravne podatke svog naloga.

##### i. Dodavanje svojih podataka o kretanju

Korisnik izabere .csv fajl koji sadrži podatke o kretanju koje želi da pošalje na server. Nakon slanja ima pristup tim podacima.

##### ii. Izmena svojih podataka u kretanju

Korisnik može da izabere neki od svojih fajlova sa podacima koje je poslao na server. Ti podaci mu se prikazu u formi gde može da ih izmeni i sačuva ili odbaci izmene.

##### iii. Brisanje svojih podataka o kretanju

Pored izmene, za izabran fajl sa listom promena korisnik može da zada komandu za brisanje. Nakon dodatne potvrde da želi to da uradi, fajl se briše iz sistema.

##### iv. Vizuelizacija podataka

Korisnik može da izabere neki od svojih podataka ili istorijske podatke u nekom periodu za neki par i da traži od sistema da ih prikaže vizuelno.

#### 5) Simuliranje

Za rad sa simuliranjem je potrebno da je registrovani korisnik prethodno izvršio autentifikaciju sebe tako što je pristupio formi za autentifikaciju i uneo ispravne podatke svog naloga.

##### i. Pokretanje simulacije

Korisnik bira strategiju koju želi da simulira. Zatim bira podatke sa kojima želi da radi, gde može da izabere istorijske podatke na nekom od ponuđenih parova u vremenskom periodu koji zada, da koristi neki od fajlova sa podacima koje je on dostavio ili da izabere neku od postojećih rang lista da radi simulaciju za rangiranje. Kad izabere strategiju i podatke, može da izda komandu za pokretanje simulacije. Simulacija se izvršava u pozadini a on može da radi druge stvari sa sistemom. Jedan korisnik u jednom trenutku može da ima samo jednu aktivnu simulaciju.

##### ii. Pregled postojećih simulacija

Korisnik može da vidi status pokrenute simulacije kao i sve simulacije koje je sačuvao. Kada se simulacija završi, njen rezultat se može videti tako što se izabere u ovom pregledu. Rezultat ostaje sačuvan. Korisnik može da briše sačuvane simulacije.

#### 6) Pregled rang lista i dokumentacije i uputstva

Pored formi za registraciju i autentifikaciju, neautentifikovan korisnik može da pregleda postojeće rang liste kao i dokumentaciju i uputstvo za korišćenje sistema.

#### 7) Administratorske funkcije

##### i. Prihvatanje novih korisnika i blokiranje starih korisnika

Administrator može da prihvati ili odbije novog korisnika, takodje ima spisak svih korisnika i mogućnost uklanjanja (blokiranja).



## ii. Dodavanje rang liste

Korisnik koji se autentifikovao kao administrator može da napravi novu rang listu tako što zada jedinstveno ime i podatke koji se simuliraju za rangiranje na isti nači kao što korisnik zadaje podatke za simuliranje.

## 6. Pretpostavke i ograničenja

Sistem je ovde specifikiran tako da se nuđenje veb usluga i simulacije korisničkih strategija izvršavaju na istom računaru. U optimalnom slučaju, ta dva dela bi bila odvojena u posebne fizičke komponente sistema da bi se održala dostupnost veb servera u periodima kada se izvršava intenzivno simuliranje. To razdvajanje nije eksplicitno učinjeno u ovoj verziji projekta zbog neiskustva projektanta i njegove nemogućnosti da sagleda posledice takve odluke na težinu faze implementacije projekta u ovom trenutku, ali se ovde eksplicitno spominje kao podsetnik za potencijalno dodavanje u nekoj narednoj verziji.

## 7. Nefunkcionalni zahtevi

### 1) Sistemski zahtevi

Za funkcionisanje sistema, neophodno je da veb server poseduje instaliranu Python implementaciju uz radni okvir *Django*. Za simuliranje je potreban samo Python. Veb stranice koriste HTML tehnologiju pa je neophodno prilagoditi ih tako da ne postoje veća odstupanja od ciljnog prikaza stranica u različitim veb pretraživačima (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).

### 2) Ostali zahtevi

Veb stranice se mogu prikazivati i na mobilnim uređajima. Bilo bi dobro kad bi veb stranice koje veb server pravi mogle da se prilagođavaju veličini ekrana na kome se prikazuju. (*Responsive Web Design*)

## 8. Zahtevi za korisničkom dokumentacijom

Strategy Playground je platforma koja nudi interfejs za rad sa programskim okruženjem. Kao takva, za upotrebu zahteva dokumentaciju aplikativnog programskog interfejsa koji nudi kao i uputstvo za korišćenje koje sadrži opis rada radnog okvira da korisnik razume i makar jedan primer koji može da sprovede korisnika kroz rad i opiše dostupne funkcionalnosti. Uputstvo za generalno korišćenje programskog jezika Python nije potrebno uključiti ovde pošto ih postoji dovoljno na internetu već je dovoljno ostaviti jedan ili više linkova ka sajtovima gde korisnici mogu da nauče više o samom jeziku.

## 9. Plan i prioriteti

Različiti delovi sistema su od različitog značaja pa se ovde definiše zamišljen red za izradu implementacija funkcionalnosti. Više pozicije na listi predstavljaju stvari sa višim prioritetom.

- Autentifikacija i registracija korisnika.
- Osnovni rad sa simulacijom.  
Korisnik pošalje fajl sa kodom na server i server odradi simulaciju sa njim.
- Pamćenje strategija kao celina koje sadrže više modula i fajl sistem za svaku
- Mogućnost snabdevanja svojim podacima o kretanju cene
- Vizuelizacija podataka o kretanju cene
- Administratorske funkcije i rang liste
- Podrška za *Responsive Web Design*

## 10. Istorija izmena

Datum	Verzija	Kratak opis	Autori
11. 03. 2023.	1.0	Inicijalna verzija	Nikola Đokić
19.03.2023.	1.1	Izmenjena verzija	Ana Vraneš