**Dizajn specifikacija**

**Association Rule Mining – Seattle Police Department Incident Response**

**PA 1**

5. 5. 2019.

**Tehnički fakultet Sveučilišta u Rijeci - Računarstvo**

**Sažetak**

Tema ovog projekta je Association Rule Mining. To je metoda za otkrivanje relacija između vrijednosti atributa u velikim bazama podataka. Pronalaze se pravila o vezama između vrijednosti atributa pomoću kojih se može predvidjeti vrijednosti atributa na temelju vrijednosti drugog atributa. Dobili smo bazu podataka u kojoj su evidentirane reakcije policije na pozive građana u Seattleu. Nadalje, trebamo otkriti vezu između različitih događaja te njihovih atributa.

**Ciljana publika**

U našem slučaju ciljana publika su profesor i asistent koji pregledavaju naše rješenje te predanu dokumentaciju. U stvarnom svijetu to bi bili potencijalni kupci koji su zainteresirani za naš proizvod. Također, ovaj dokument je namijenjen i svim inženjerima koji izvode validaciju sustava te menadžerima.

**Članovi projektnog tima**

1. Projektni manager – Luka Babić

2. Stručnjak za specifikaciju – Borna Gilja

3. Stručnjak za oblikovanje – Kristijan Knežić

4. Stručnjak za testiranje – Luka Šarlija

5. Stručnjak za integraciju – Marin Markanjević

**Verzija kontrole dokumenta**

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Primarni autor(i)** | **Opis verzije** | **Datum završetka** |
| Radna | Kristijan Knežić | Početak pisanja plana | 05.05.2019. |
| Radna | Kristijan Knežić | Ispravak dokumenta | 09.06.2019. |

**Sadržaj**

[1. Uvod 4](#_Toc7991856)

[1.1. Sažetak faze dizajna 4](#_Toc7991857)

[1.2. Matrica praćenja zahtjeva 4](#_Toc7991858)

[2. Dizajn arhitekture sustava 5](#_Toc7991859)

[2.1. Odabrana arhitektura sustava 5](#_Toc7991860)

[2.2. Diskusija alternativnih dizajna 5](#_Toc7991861)

[2.3. Opis sučelja sustava 7](#_Toc7991862)

[3. Detaljan opis komponenata 8](#_Toc7991863)

[3.1. Main 8](#_Toc7991864)

[3.2. Prijava 8](#_Toc7991865)

[3.3. Sučelje 8](#_Toc7991866)

[3.4. Upit 8](#_Toc7991867)

[3.5. Algoritam 9](#_Toc7991868)

[3.6. Događaj 9](#_Toc7991869)

[4. Dizajn korisničkog sučelja 10](#_Toc7991870)

[4.1. Opis korisničkog sučelja 10](#_Toc7991871)

[4.1.1. Grafički elementi sučelja 10](#_Toc7991872)

[4.1.2. Objekti i akcije 11](#_Toc7991873)

# Uvod

## Sažetak faze dizajna

U ovom dokumentu sadržano je detaljno objašnjenje dizajna projekta. Dokument objašnjava na koji će se način implementirati aplikacija te koje će funkcionalnosti imati. Također opisuje na koji će se način ispuniti traženi zahtjevi navedeni u Specifikaciji dizajna. Sadrži objašnjenje zašto je izabran korišteni pristup dizajna u odnosu na druge. Opisuje dizajn korisničkog sučelja.

Trenutni, ovjde korišteni, dizajn ostvaren je međusobnim idejama članova tima. Brainstorm sesijom određeni su mogući problemi tj. scenariji. S obzirom na to biran je dizajn. Iznešene su prednosti i mane svake ideje. Određene su klase te je pomoću CRC kartona dobiven model za koji smo odredili način implementacije, to jest korišteni dizajn.

## Matrica praćenja zahtjeva

|  |  |  |
| --- | --- | --- |
|  | **Zahtjev** | **Komponenta (Klasa)** |
| 1. | Pokretanje aplikacije | Main, Prijava, Sučelje |
| 2. | Prijava u aplikaciju | Prijava |
| 3. | Odabir parametara | Upit |
| 4. | Obrada podataka | Algoritam |
| 5. | Prikaz podataka | Događaj, Sučelje |

# Dizajn arhitekture sustava

## 2.1. Odabrana arhitektura sustava

Komponente sustava podijeljene prema specifičnim zahtjevima:

**Pokretanje aplikacije** – za ispunjenje ovog zahtjeva koristi se klasa *Main*, klasa *Prijava* te klasa *Sučelje*. Namjena komponente *Main* je integracija svih ostalih komponenta u cijelinu tj. pokretanje cijele aplikacije. Sadrži funkciju koja poziva ostale komponente da bi korisnik dobio grafičko sučelje. Komponenta *Prijava* služi da bi se korisnik mogao prijaviti valjanim podacima u aplikaciju. Te komponenta *Sučelje* koja služi za prikaz podataka.

**Prijava u aplikaciju –** koristi se klasa *Prijava*. Namjena joj je korisniku prikazati mjesto za upis korsničkog imena i lozinke te provjeriti ispravnost istih. Ako je sve ispravno komponenta *Prijava* pomogućuje daljnji poziv komponente *Sučelje*.

**Odabir parametara pretraživanja baze** – koristi se klasa *Upit*. Namjena joj je da prema korisnikovom odabiru prosljedi odabrane parametre komponenti *Algoritam*.

**Obrada podataka** – koristi se klasa *Algoritam*. Namjena komponente je da pretraži bazu podataka povezanu na aplikaciju s obzirom na odabrane parametre dobivene od komponente *Upit*. Obrađeni podaci se prosljeđuju komponenti *Sučelje*.

**Prikaz podataka** – koristi se klasa *Događaj* i klasa *Sučelje.* Komponenta *Događaj* sadrži podatke za svaki određeni događaj. Pomoću komponente *Sučelje* iste te događaje prikazujemo korisniku.

## 2.2. Diskusija alternativnih dizajna

Prilikom odabira dizajna dolazi do alteranativnih rješenja dizajna. Neka od njih su:

* korisnik za pretaživanje učitava bazu. Za implementaciju ovog rješenja bilo bi potrebno koristiti dodatnu komponentu/klasu *Unos* koja bi povezivala aplikaciju s bazom podataka. Nedostaci i razlog zašto ovakav dizajn nije korišten je zbog jednostavnosti te zbog korištenja uvijek iste baze podataka. Ovime smo omogučili da bazu podataka prilagodimo aplikaciji te je ona uvijek povezana s istom.
* korisniku se prikazuju svi događaji nakon čega može koristiti filtere, odnosno odabrati parametre. Ovakav pristup nije korišten zbog toga što bi morali slati zahtjev bazi prvi puta za prikaz svih događaja te slati zathev drugi puta za prikaz događaja s obzirom na odabrane parametre. Trenutni dizajn je korišten zbog logičkog slijeda, korisnik odabire parametre te se samo jednom šalje zahtjev prema bazi za prikaz traženih događaja. Za implementaciju ovakvog alternativnog rješenja nebi bilo potrebe za uvođenje novih komponenata već bi samo bilo potrebno izmjeniti *Upit i Sučelje.*
* korisnik ne mora obaviti prijavu, može se prijaviti kao gost. To bi značilo da bi u stvarnom svijetu svatko imao pristup podacima. Implementacija je moguća izmjenom komponente *Prijava.* Ne koristimo je trenutno zbog sigurnosti sustava.
* kao jedan od alternativnih dizajna je bio da korisnik vidi odvojeno prvo grafičko sučelje za upit, gdje bi odabrao parametre. Nakon što bi potvrdio odabir vidio bi drugo odvojeno grafičko sučelje gdje bi mu se prikazali rezultati. Za razliku od ovog dizajna, pozitivna strana korištenog je što korisnik istovremeno ima pregled na parametre i na rezultat što mu omogućava lakše snalaženje.

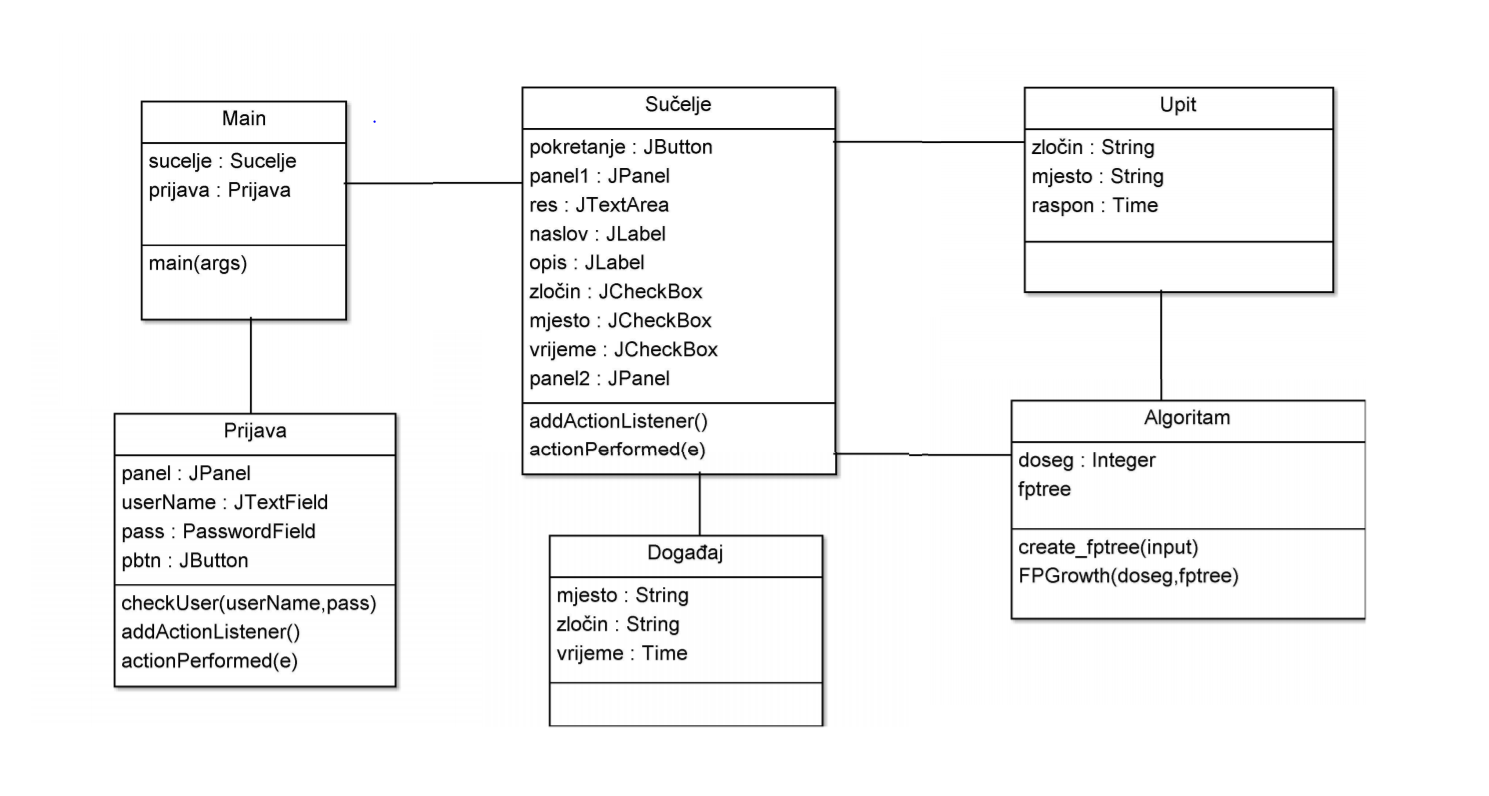
Potreba za nadogradnjom:

* mogućnost nadogradnje odabira parametara za pretraživanje. Ako bi bilo potrebno u budućnosti moguće je implementirati dodatne funkcionalnosti odabira parametara za pretraživanje događaja. Da bi isto bilo napravljeno potrebno je izmjeniti klasu *Upit i Sučelje*.

Održavanje sustava:

* Ukoliko se u budoćnosti javi potreba za izmjenom baze podataka, moguće je implementirati klasu *Unos* pomoću koje bi korisnik mogao birati pretraživanje po određenoj bazi podataka. U trenutnom slučaju to nema smisla zbog toga što je korištena samo jedna baza podataka.

## 2.3. Opis sučelja sustava



*Class dijagram*

Detaljnije podatke o sučelju sustava vidimo u Class dijagramu.

# 3. Detaljan opis komponenata

## 3.1. Main

Glavni zadatak Main komponente je integracija svih ostalih komponenata u cjelinu. Smatramo je procedurom. Postoji zbog potrebe smislenog povezivanja ostalih komponenata. Funkcija joj je pozivati ostale komponente kada je potrebno. Prilikom pokretanja aplikacije Main vrši interakciju s komponentom Prijava. Ako je prijava uspješno odrađena poziva komponentu Sučelje.

## 3.2. Prijava

Komponenta Prijava je procedura, korisniku prikazuje sučelje gdje upisuje podatke za prijavu u sustav te ih provjerava. Ova komponenta postoji zbog autentifikacije korisnika, a funkcija joj je provjeriti da li su podaci za prijavu dobiveni od korisnika ispravni. Ako jesu dopusiti mu dalje interakciju sa sustavom ili u suprotnom blokirati. Komponenta ovisi o komponenti Main koja ju poziva. Također komponenta Prijava će interaktirati s Main ovisno je li prijava uspješno izvršena. Komponenta sadrži podatke koje uspoređuje s unosom korisnika te na temelju toga provjerava ispravnost prijave.

## 3.3. Sučelje

Sučelje postoji zbog potrebe za prikazom podataka te odabira parametara pretraživanja. Funkcija mu je omogućiti korisniku odabir željenih parametara te ih proslijediti komponentom Upit. Također druga funkcija Sučelja je jednom kad Algoritam pronađe tražene događaje prikazati iste. Komponenta interaktira s Upitom, Algoritmom, Događajem. Komponenta Sučelje biva pozivana od komponente Main.

## 3.4. Upit

Svrha komponente Upit je prosljediti parametre do komponente Algoritam. Funkcija je spremiti parametre koje je korisnik odabrao na Sučelju te ih prosljediti komponenti Algoritam. Komponentu Upit možemo smatrati spremnikom podataka. Ovisi o Sučelju te interaktira s Algoritmom.

## 3.5. Algoritam

Svrha komponente je obrada podataka. Vrsta komponente je proces. Funkcija mu je pronaći događaje iz baze koji odgovaraju odabranim parametrima. Komponenta implementira FP-growth algoritam. Ovisi o Upitu, tj. podacima dobivenim od komponente Upit. Interaktira sa s komponentom Događaj te s komponentom Sučelje.

## 3.6. Događaj

Svrha ove komponente je da služi kao spremik podataka za neki događaj iz baze podataka. Funkcija joj je da olakša prikaz jednom dobivenih događaja koje je pronašla komponenta Algoritam. U njoj su spremljeni svi podaci o određenom događaju. Interaktira sa Sučeljem gdje se vrši prikaz događaja.

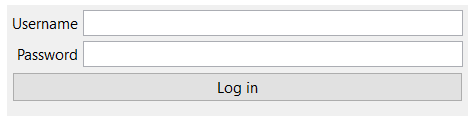
# 4. Dizajn korisničkog sučelja

## 4.1. Opis korisničkog sučelja

### 4.1.1. Grafički elementi sučelja

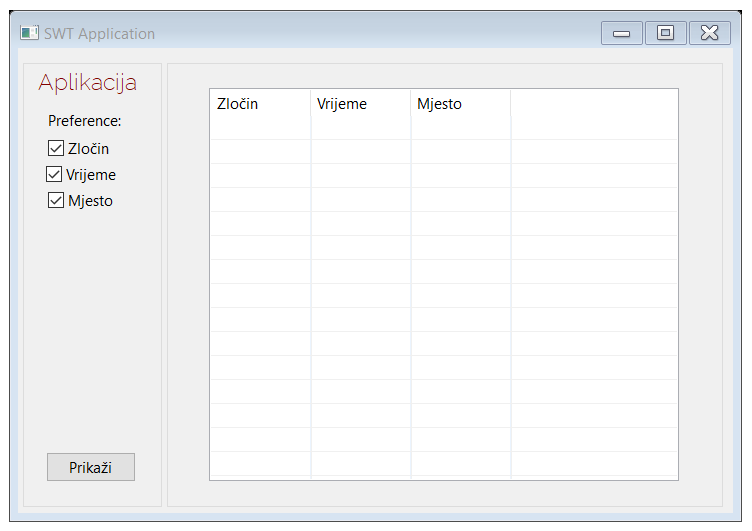
Za pokretanje aplikacije korisnik će na računalu imati ikonu aplikacije na koju će dvoklinkuti kako bi prokrenuo istu. Ovime je osiguran zahtjev pokretanja aplikacije.

*Primjer sučelja za prijavu*



Prilikom pokretanja aplikacije od korisnika se zahtjeva unos korisničkog imena i lozinke. Pritiskom gumba „Log in“, ako je ispravna prijava korisnik dolazi do glavnog sučelja. Zahtjev za prijavu u aplikaciju.

*Primjer glavnog sučelja*



Na lijevoj strani sučelja korisnik može odabrati parametre po kojima želi pretražiti događaje u bazi podataka. Pritiskom na gumb „Prikaži“ pokrenuti će se algoritam za pretragu po odabranim parametrima. Zahtjev za odabirom parametara pretraživanja.

Nakon pokrenute pretrage korisniku se daje do znanja da se pretraga vrši. Za to vrijeme se u pozadini koristi algoritam za pretragu. Zahtjev obrade podataka.

Na desnoj strani sučelja prikazati će se rezultat jednom kada se algoritam pretrage izvrši. Zahtjev prikaza podataka.

### 4.1.2. Objekti i akcije

Objekt s kojima korisnik interaktira je Događaj. Sadrži sve potrebne podatke o nekom određenom zločinu koji je pronađen pretragom u bazi podataka.

Korisniku se od akcija redom nude sljedeće:  
 - pokretanje aplikacije

- odabir parametara

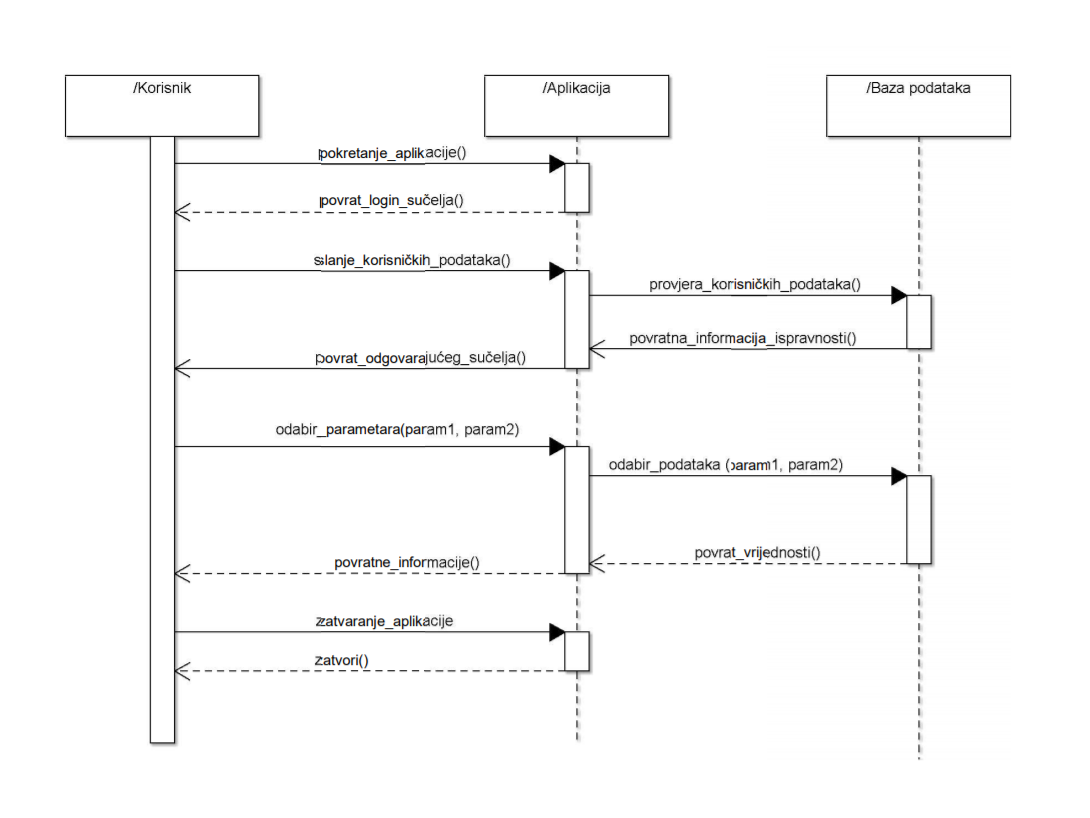
- prikaz podataka

- zatvaranje aplikacije

Akcije koje vrši aplikacija prema bazi podataka su:

- pretraga podataka s obzirom na odabir parametara

- provjera podataka za prijavu korisnika



*Sequence dijagram*