Projekat :

University Data Analysis Tool

Baze Podataka

|  |  |  |
| --- | --- | --- |
|  | Studenti : | Begić Edin |
| Sarajevo, |  | Mulić Vedad |
| 05.11.2017. |  | Mušić Mirza |

# 1. Opis teme

Implementacija ovog projekta se svodi na dva dijela. Prvi korak jeste kreiranje određene vrste skladišta podataka (*data warehouse*) koja će biti vezana za bazu podataka Elektrotehničkog fakulteta.

Skladište podataka predstavlja bazu podataka koja je dizajnirana kako bi omogućila razne aktivnosti koje se vežu za poslovnu inteligenciju (*business intelligence*), odnosno postojanje takvih skladišta pomaže krajnim korisnicima da bolje razumiju i poboljšaju performanse organizacije. DW su dizajnirani za izvršavanje kompleksnih upita te analizu prikupljenih podataka, što predstavlja glavnu razliku od tradicionalnih transakcijskih baza podataka.

Da bi se postigao cilj primjene poslovne inteligencije, DW procesiraju podatke koji se prikupljaju iz više izvora podataka. Podaci se dobijaju iz sistema koji su interno razvijeni, plaćenih aplikacija, eksternih dokumenata, itd.

DW se razlikuju od OLTP (*online transaction processing*) sistema. Kao što je navedeno, pomoću DW-a se pravi razlika između rada nad transakcijama i rada na analizi podataka. Iz tog razloga, skladišta podataka su sistemi koji su općenito fokusirani na čitanje podataka (*read-oriented systems*). Omjer čitanja iz baze je dosta veći u odnosu na pisanje i ažuriranje baze podataka.

Prilikom implementacije skladišta podataka, potrebno je izvršiti niz procesa kako bi se efikasno odvijala migracija podataka, što uključuje *ETL* (izdvajanje, transformacija i učitavanje podataka).

Drugi dio projekta je fokusiran na kreiranje BI (*buisness intelligence*) alata. BI predstavlja strategijie i tehnologije korištene od organizacija za analizu podataka i poslovnih informacija. Ključno za svaki takav alat jeste da pruža:

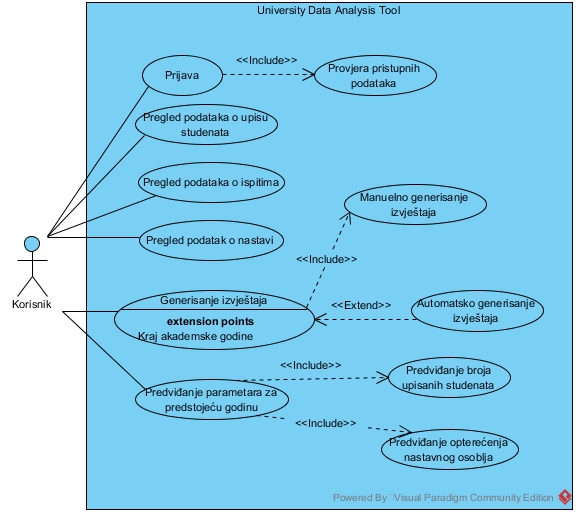
* Historijski pregled vezan za poslovne informacije organizacije
* Pregled trenutnog stanja
* Predviđanja vezana za poslovne informacije

Dobar BI alat fokusiran na univerzitet/fakultet bi trebao biti dovoljno fleksibilan da se koristi u edukacijske svrhe i dovoljno robustan da pruži kvalitetan *data mashup* (spajanje dva ili više skupa podataka u jedan grafički interfejs), te omogući *drill down* podataka.

Alat koji će se implementirati u ovom projektu, služit će fakultetskom osoblju kako bi imali bolji uvid u trenutnu situaciju. Pored toga, alat će omogućiti da se vrše predviđanja za buduće akademske godine u pogledu broja upisanih studenata, broja potrebnog nastavnog osoblja na određenom kursu, itd. Ta predviđanja će se vršiti na osnovu akviziranih podataka iz prošlosti, koji su sačuvani u skladištu podataka.

# 2. Detaljni opis funkcionalnosti

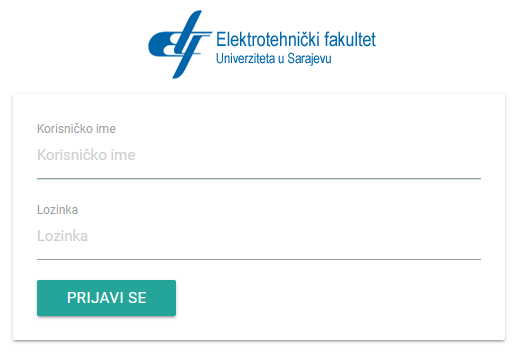
Nakon što su u prethodnom dijelu ukratko navedene funkcionalnosti ovog softverskog rješenja, u ovom poglavlju će se detaljnije opisati osnovne funkcionalnosti.



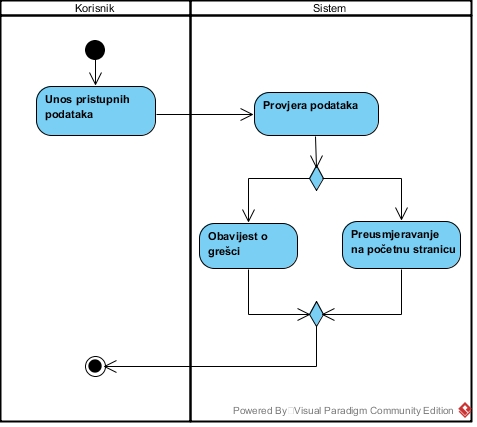
**Slika 1** – Generalni use case dijagram projekta

## 2.1 Prijava korisnika

Prilikom pristupa aplikaciji od korisnika se traži unos korisničkog imena i lozinke. Ukoliko su pristupni podaci ispravni korisnik se preusmjerava na početnu stranicu aplikacije. Podaci koje korisnik unosi se moraju podudarati sa podacima koji se nalaze u bazi podataka fakulteta. Pristup je dozvoljen isključivo uposlenicima studentske službe te administratoru čitavog sistema na fakultetu. Ukoliko neko od nastavnog osoblja želi pristupiti nekom od izvještajam, prvo se mora obratiti studentskoj službi. Primjer forme za prijavu korisnika se može vidjeti na sljedećoj slici.



**Slika 2** – Forma za prijavu korisnika



**Slika 3** – Activity dijagram za prijavu korisnika

## 2.2 Vizualizacija i analiza podataka

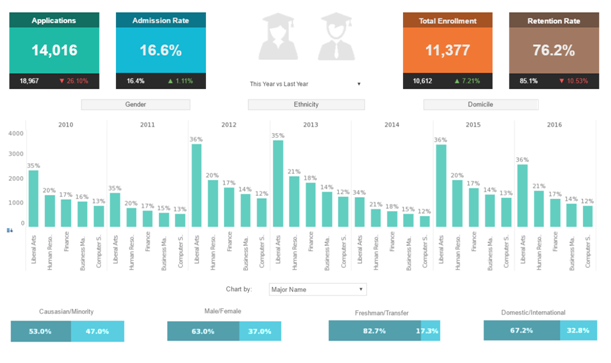
Svrha ove aplikacije jeste omogućavanje detaljnog i pri tome jednostavnog uvida u sve podatke koje su vezane za fakultet. Za sve podatke će biti omogućen “drill down”, tj. pregled sa više detalja. Svi podaci su uvijek aktuelni s tim da postoji mogućnost pregleda podataka u zavisnosti od akademske ili kalendarske godine.

### 2.2.3 Pregled podataka o studentima

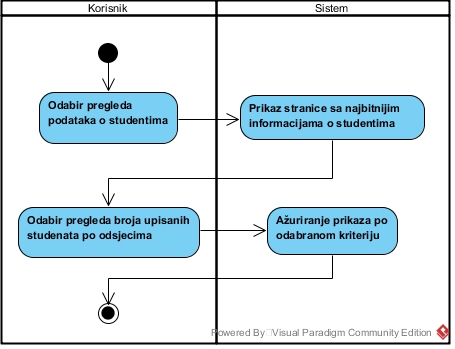
Jedan od najbitnijih entiteta u ovoj aplikaciji je svakako student te će zbog toga biti moguće analizirati ove podatke iz više različitih uglova:

* Broj upisanih studenata u trenutnoj akademskoj godini u zavisnosti od odsjeka
* Procenat studenata koji obnavljaju godinu
* Broj apsolvenata u odnosu na ukupan broj studenata
* Prosjek ocjena za studente u zavisnosti od ciklusa studija, odsjeka i ostalih parametara
* Broj studenata na budžetu

Nakon što su podaci prikupljeni i analizirani prikazuju se krajnjem korisniku putem vizualizacije, kao što se to vidi na slici.



**Slika 4** .- Moguća forma o pregledu podataka o studentima



**Slika 5** – Activity dijagram o pregledu podataka o studentima

### 

### 2.2.3 Pregled podataka o posjećenosti nastavi

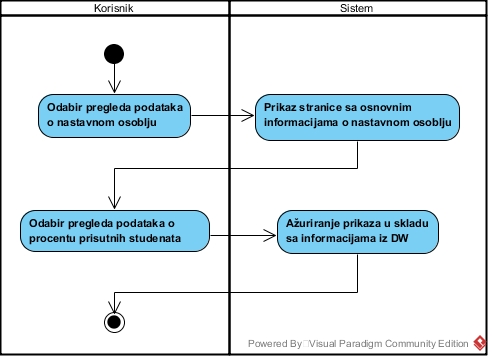
Pored studenata, za potpun pregled podataka o jednom fakultetu, neophodno je imati uvid u podatke o najbitnijim nastavnim aktivnostima, a to su vježbe/tutorijali i predavanjima. Ovi podaci su predstavljeni kroz nekoliko pogleda:

* Pregled podataka o posjećenosti predavanja i vježbi
  + broj studenata koji su prisutni
  + procenat prisutnih studenata
* Posjećenost predavanja u zavisnosti od vremena održavanja
  + dan u sedmici
  + semestar
  + dan u sedmici
* Posjećenost predavanja u zavisnosti od odsjeka
* Pregled najposjećenijih predmeta
* Pregled nastavnog osoblja na čijim predavanjima/vježbama je prisutno najviše (ili najmanje) studenata

Potencijalni izgled dijela aplikacije koji prikazuje podatke o akademskom osoblju bi mogao biti kao na slici



**Slika 6** – Moguća forma pregleda podataka o prisutnosti sudenata



**Slika 7** – Activity dijagram o pregledu podataka o prisutnosti studenata

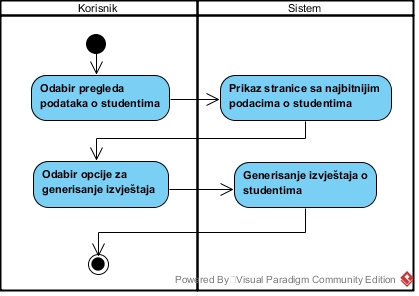
## 2.2.4 Pregled podataka o ispitima

Još jedan bitan aspekt nastavnog procesa na fakultetu jesu ispiti. Analizom velikih količina informacija o održanim ispitima moguće je uočiti neke probleme koji utiču na uspješnost nastavnog procesa. Kada su u pitanju ispiti, moguće je analizirati podatke o izlaznosti studenata u zavisnosti od odsjeka, akademske godine, kalendarske godine i predmeta. Takođe, moguće je analizirati prosječan uspjeh svih studenata na ispitima u zavisnosti od istih parametara.

## 2.3 Generisanje izvještaja

Pored pregleda i analize postojećih podataka, moguće je generisati izvještaje. Neki izvještaji su automatski, i generišu se krajem akademske godine. Ovi izvještaji sadrže statističke informacije o uspjehu studenata u trenutnoj akademskoj godini te određene pokazatelje o promjenama u odnosu na prethodnu akademsku godinu.

Drugi tip izvještaja se generiše “on demand”. Ovi izvještaji se mogu generisati kroz grafički korisnički interfejs te preuzeti u *pdf* formatu.



**Slika 8** – Activity dijagram generisanja izvještaja

## 2.4 Predviđanje trendova u budućnosti

Uz pomoć analize velike količine podataka koji su pohranjeni u *Data Warehouse-*u moguće je napraviti određena predviđanja koja bi dovela do poboljšanja rada fakulteta.

### 2.4.1 Predviđanje broja upisanih studenata

Na osnovu podataka o prethodnim akademskim godinama moguće je predvidjeti koliko će studenata upisati sljedeću akademsku godinu. Ove inforamcije mogu biti ključne kod angažovanja nastavnog osoblja i alokacije resursa (broj računara u laboratorijama, broj nastavnih grupa) za predstojeću akademsku godinu. Predviđanja se predstavljaju kroz grafičke elemente kao što je to slučaj i sa prethodnim funkcionalnostima.

### 2.4.2 Predviđanje opterećenja nastavnog osoblja

Moguće je napraviti predikcije o tome na kojem kursu je neophodno proširiti nastavni ansambl kako bi se kurs mogao izvoditi na adekvatan način.

### 2.5 Dodatne funkcionalnosti:

Pored osnovnih funkcionalnosti sistema, postoji još nekoliko dodatnih koje uveliko olakšavaju korištenje University Analysis Tool- a a to su:

1. Uvoz eksternih dokumenata
2. Filteri za analizu/predikciju
3. Usporedba rezultata
4. Obavještavanje korisnika
5. Vremensko raspoređivanje procesa

### 2.5.1 Uvoz eksternih dokumenata

Korisnicima je pružena mogućnost da se učitaju eksterni fajlovi ukoliko se želi izvršiti analiza podataka koji trenutno nisu locirani unutar postojećeg skladišta podataka.

Ovo je omogućeno tako što se uvezeni eksterni podaci mapiraju sa već postojećim, kreira se njihov model i dolazi do generisanja izvještaja na osnovu opcionalnih analiza ili predikcija.

### 2.5.2 Filteri za analizu/predikciju

Kako bi se dobio što detaljniji uvid u rezultate analize/predikcije moguće je izvršiti filtriranje rezultata. Ovime je uveliko olakšano da korisnici izvuku tražene informacije bez dodatnog napora uloženog u očitavanje kompletnog izvještaja u vidu vizuelnog dijagrama.

### 2.5.4 Obavještavanje korisnika

Ukoliko se analiza/predikcija automatski pokreću i ukoliko se vrši sinhronizacija u realnom vremenu, dolazi do potrebe za obavještavanjem odgovorne osobe ako navedeni procesi nad podacima ne daju zadovoljavajuće rezultate kako bi nadležne osobe mogle pravovremeno reagovati. Također, moguće je postaviti uslov prilikom analize/predikcije i ukoliko uslov bude zadovoljen doći će do obavještavanja korisnika o nastaloj situaciji.

### 2.5.5 Vremensko raspoređivanje procesa

Zbog promjenljivosti podataka dolazi i do promjene rezultata analize/predikcije, pa se javila potreba da se omogući automatizacija većine procesa. Ukoliko u određenom trenutku dođe do promjene nad podacima koji su bitni korisniku, isti će biti obaviješten o toj promjeni dostavljanjem e-maila s generisanim izvještajem promjena.

# 3. Tehnologija

Tehnološki stek projekta je sadržan od tri komponente:

1. **Spring -** koristiti će se za potpuno struktuiranje, održavanje i upravljanje vitalnim procesima web aplikacije
2. **Android -** koristiti će se za pravljenje mobilne aplikacije koja će sadržavati iste funkcionalnosti kao i web verzija
3. **AngularJS -** koristiti će se za kreiranje i upravljanje vizuelnim dinamičkim elementima web aplikacije

3.1 Spring

Spring Framework je Java framework koji pruža sveobuhvatnu infrastrukturnu podršku za razvoj Java aplikacija. Osnovne funkcionalnosti Spring- a mogu biti iskorištene za bilo koju Java aplikaciju, ali postoje ekstenzije za razvoj web aplikacija koje se nalaze na vrhu Java EE platforme. Ovaj framework je među korisnicima Java programskog jezika postao popularan kao dodatak ili čak zamjena za Enterprise Javabeans (EJB) model a u prilog tome ide i činjenica da je open-source.

Ono po čemu je Spring poznat je DI (dependency injection). Prilikom razvoja kompleksne Java aplikacije, klase bi trebale biti nezavisne što je više moguće od ostalih klasa kako bi se iskoristila mogućnost da se kasnije te klase ponovo iskoriste i da se testiraju neovisno od ostalih prilikom unit testova. DI pomaže u integraciji ovih klasa ali isto tako i u njihovom očuvanju neovisnosti.

3.2 Android

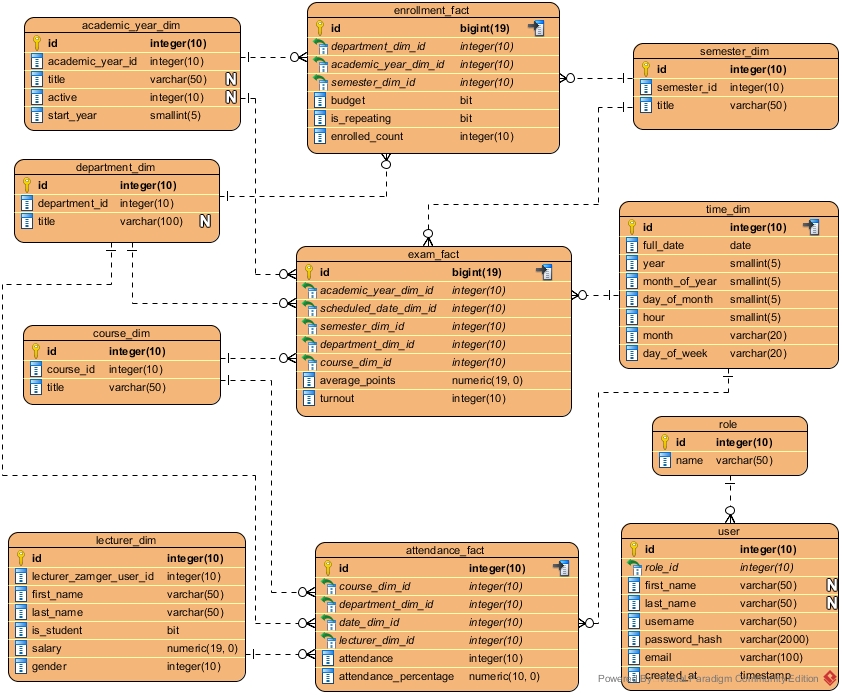
Android predstavlja operativni sistem koji se najčešće koristi na mobilnim uređajima i omogućava razvoj inovativnih aplikacija i igara u Java jezičkom okruženju. Android je kao i Spring *open-source* i dostupne su ogromne količine materijala u vidu biblioteka za razvoj.

3.3 AngularJS

AngularJS je strukturalni framework za dinamične web aplikacije. Omogućava korištenje HTML- a kao osnovnog jezika i dopušta proširenje HTML sintakse da bi se jasnije izrazile aplikacijske vizuelne komponente.

AngularJS je ono što bi HTML trebao biti, da bude kreiran za aplikacije. HTML je odličan deklarativni jezik za statične dokumente ali postoji neusklađenost statičnih dokumenata i dinamičkih aplikacija. Navedena neusklađenost je riješena sa okruženjem i bibliotekama koje sadrže kolekciju funkcija koje su korisne prilikom razvoja web aplikacija.

# 4. ER dijagram



**Slika 9** – ER dijagram *Data Warehouse-*a