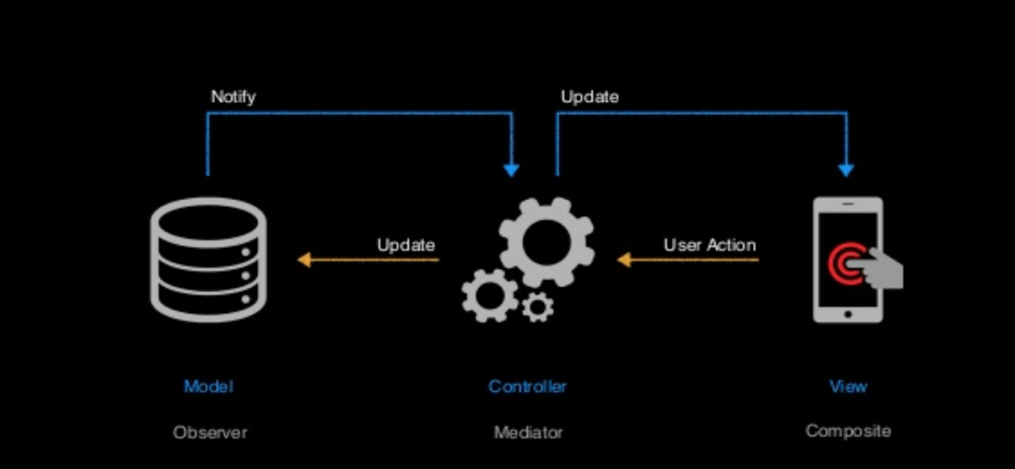
**ARHITEKTURA I DIZAJN**

Arhitektura programske potpore je jedan od najbitnijih koraka u oblikovanju programske potpore. Osnovna uloga arhitekture programske potpore je apstrakcija sustava na visokom nivou. Arhitektura programske potpore mora strukturirati projekt i samu programsku podršku. Izrađuje se prije detaljne specifikacije i osnovna je za komunikaciju dionika. Prednost definiranja arhitekture su što smanjuje cijenu oblikovanja, razvoja i održavanja programskog proizvoda.

S obzirom na to da je naš zadatak napraviti mobilnu aplikaciju koja ima bazu podataka na serveru, odlučili smo se za MVC (engl. Model-View-Controller) arhitekturu. MVC arhitektura je prilagođena objektno orijentiranoj paradigmi pa time dodatno olakšavamo oblikovanje i održavanje programskog proizvoda.

Arhitektura MVC se sastoji od 3 dijela:

1. **Model** - služi za dohvaćanje, umetanje i izmjenu informacija u bazi podataka
2. **View** - služi za prikaz informacija koje korisnik može vidjeti, u našem slučaju view se odnosi na .xml datoteke koje oblikuju izgled mobilne aplikacije
3. **Controller** - služi kao posrednik između viewa i modela, sve što korisnik napravi registrira se na controlleru, također on upravlja sa viewem i razgovara s modelom

****

Slika 1 MVC

Na slici 1 vidi se način rada MVC-a, korisnik radi promjene na viewu(upisuje podatke, traži pristup nekim podacima itd.) sve te promjene se registriraju na controlleru koji njegove zahtjeve obrađuje i po potrebi ažurira podatke modela i prikazuje ih na viewu. U niti jednom trenutku korisnik ne priča izravno s bazom podataka ili baza podataka sa korisnikom,tj. controller je ovdje medijator između njih dvoje.

**BAZA PODATAKA**

Za potrebe našeg sustave i mobilne aplikacije koristi ćemo relacijsku bazu podataka čija struktura uvelike olakšava modeliranje događaja i entiteta iz stvarnog svijeta i njihovih podataka. Kvant relacijske baze podataka je relacija, odnosno tablica koja je opisana svojim imenom i skupom pripadajućih atributa. Sve relacije u bazi su svedene na 3. normalnu formu stoga u bazi nemamo redundantnih podataka.

Slijedi prikazi i opis svih relacija i njihovih atributa te pripadajućih primarnih i stranih ključeva:

**Korisnik**

* korisnikID: VARCHAR(255) - autogenerirani primarni ključ korisnika
* ime: VARCHAR(255) - ime korisnika
* prezime: VARCHAR(255) - prezime korisnika
* email: VARCHAR(255) - email korisnika
* zaporkaHASH: VARCHAR(255) - hešriana zaporka korisnika
* dob: SMALLINT(127) - dob korisnika
* slikaID: VARCHAR(255) - ključ slike spremljen u drugoj tablici
* opis: VARCHAR(255) - opis korisnika
* datumRegistracije: TIMESTAMP - datum kada se korisnik prvi puta registrirao
* jeValidiran: BOOLEAN - je li korisnik verificirao svoj korisnički račun
* brojTelefona: VARCHAR(15) - telefonski broj korisnika

**PK {**korisnikID**}**

**K {**email**}**

**FK {**slikaID**}**

Entitet "Korisnik" opisuje svakog korisnika aplikacije. Entitet "Korisnik" ima 2 jedinstvena ključa: korisnikID i email. KorisnikID je primarni ključ entiteta i generira se pri registraciji svakog korisnika. Svaki Entitet "Korisnik" sadrži osnovne podatke o korisniku: ime, prezime, email, dob, opis, datumRegistracije, brojTelefona, je li korisnik verificirao svoj račun. U bazi podata se ne pamti zaporka nego HASH zaporke. SlikaID je strani ključ entiteta "Slika".

**Poruka**

* porukaID: VARCHAR(255) - autogenerirani primarni ključ poruke
* posiljateljID: VARCHAR(255) - korisnikID korisnika koji je poslao poruku
* primateljID: VARCHAR(255) - korisnikID korisnika koji je primio poruku
* sadrzaj: VARCHAR(255) - sadržaj poruke
* datum: TIMESTAMP - datum kada je poruka poslana

**PK {**porukaID**}**

**FK {**primateljID,posiljateljID**}**

Entitet "Poruka" opisuje svaku poslanu poruku. Svaka poruka ima jedinstveni ključ porukaID. PorukaID se generira prilikom slanja svake poruke. Poruka ima sadržaj i datum kada je poslana. Svaka poruka ima korisnikID posiljatelja i primatelja iste.

**Slika**

* slikaID**:** VARCHAR(255) - autogenerirani primarni ključ slike
* slikaBLOB: BLOB - binarni objekt koji predstavlja sliku

**PK {**slikaID**}**

Entitet "Slika" predstavlja sliku. Svaka slika ima jedinstveni ključ slikaID. Entitet "Slika" ima i atribut slikaBLOB koji predstavlja binarni objekt slike.

**KorisnikStatistika**

* korisnikID: VARCHAR(255)-kljuc korisnika
* ukBrojPoslovaPonudenih: INT - broj poslova koje je korisnik ponudio
* ukBrojPoslovaOdradenih: INT - broj poslova koje je korisnik odradio
* ukNovacPotrosen: INT - ukupna količina novaca koju je korisnik potrošio
* ukNovacZaraden: INT - ukupna količina novaca koju je korisnik zaradio
* prosjecnaOcjena: Double - prosječna ocjena korisnika

**PK{**korisnikID**}**

Za svakog korisnika se vodi njegova statistika. Za svakog korisnika se pamti ukupan broj poslova koje je taj korisnik ponudio, ukupan broj poslova koje je taj korisnik odradio, ukupna kolicna novca koju je taj korisnik potrošio, ukupna količina novca koju je taj korisnik zaradio i prosječna ocjena korisnika.

**Posao**

* posaoID: VARCHAR(255) - autogenerirani primarni kljuc posla
* naslov: VARCHAR(255) - naslov posla
* opis: VARCHAR(1023) - opis posla
* lokacija: VARCHAR(255) - lokacija gdje će se posao održavati
* vrijeme: TIMESTAMP(255) - vrijeme kada će se posao održati
* trajanje: LONG - okvirno vrijeme potrebno za odrađivanje posla
* ponudeniNovac: INT - novac ponuđen za odrađivanje posla
* posaoGotov: BOOLEAN - zastavica je li posao odrađen
* kategorijaID: VARCHAR(255) - kljuc za kategoriju u koju spada posao

**PK{**posaoID**}**

**FK{**kategorijaID**}**

Entitet "Posao" opisuje svaki posao. Entitet PosaoID je primarni ključ entiteta i generira se pri izradi svakog posla. Svaki posao ima naslov,opis posla,lokaciju gdje se održava,vrijeme kad će se posao odraditi, okvirno vrijeme trajanja posla, ponuđeni novac za odradu posla, zastavicu je li posao već odrađen i kategoriju u koju posao spada.

**Kategorija**

* kategorijaID: VARCHAR(255) - primarni ključ kategorije
* naziv: VARCHAR(255) - naziv kategorije
* opis: VARCHAR(255) - opširan opis što sve kategorija podrazumijeva

**PK{**kategorijaID**}**

Entitet "Kategorija" opisuje svaku kategoriju koju posao može imati. Svaka kategorija ima svoj naziv i svoj opis.

**Korisnik-Posao**

* posaoID: VARCHAR(255) - posaoID posla
* poslodavacID: VARCHAR(255) - korisnikD korisnika koji je ponudio posao
* posloprimacID: VARCHAR(255) - korisnikID korisnika koji je odradio posao
* sklopljenPosaoTrenutak: TIMESTAMP - trenutak kada su poslodavac i posloprimac sklopili posao
* odrađenPosaoTrenutak: TIMESTAMP - trenutak kada je posao odrađen

**PK {**PosaoID**}**

**FK {**poslodavacID,posloprimacID**}**

Entitet "Korisnik-Posao" povezuje korisnike (posloprimce i poslodavce s poslom) i pamti kada je posao sklopljen između poslodavca i posloprimca i kada je posao odrađen.

**Ocjena**

* korisnikDaoID: VARCHAR(255)
* korisnikPrimioID: VARCHAR(255)
* posaoID: VARCHAR(255)
* ocjena: SMALLINT(5)
* datumOcjenjen: TIMESTAMP

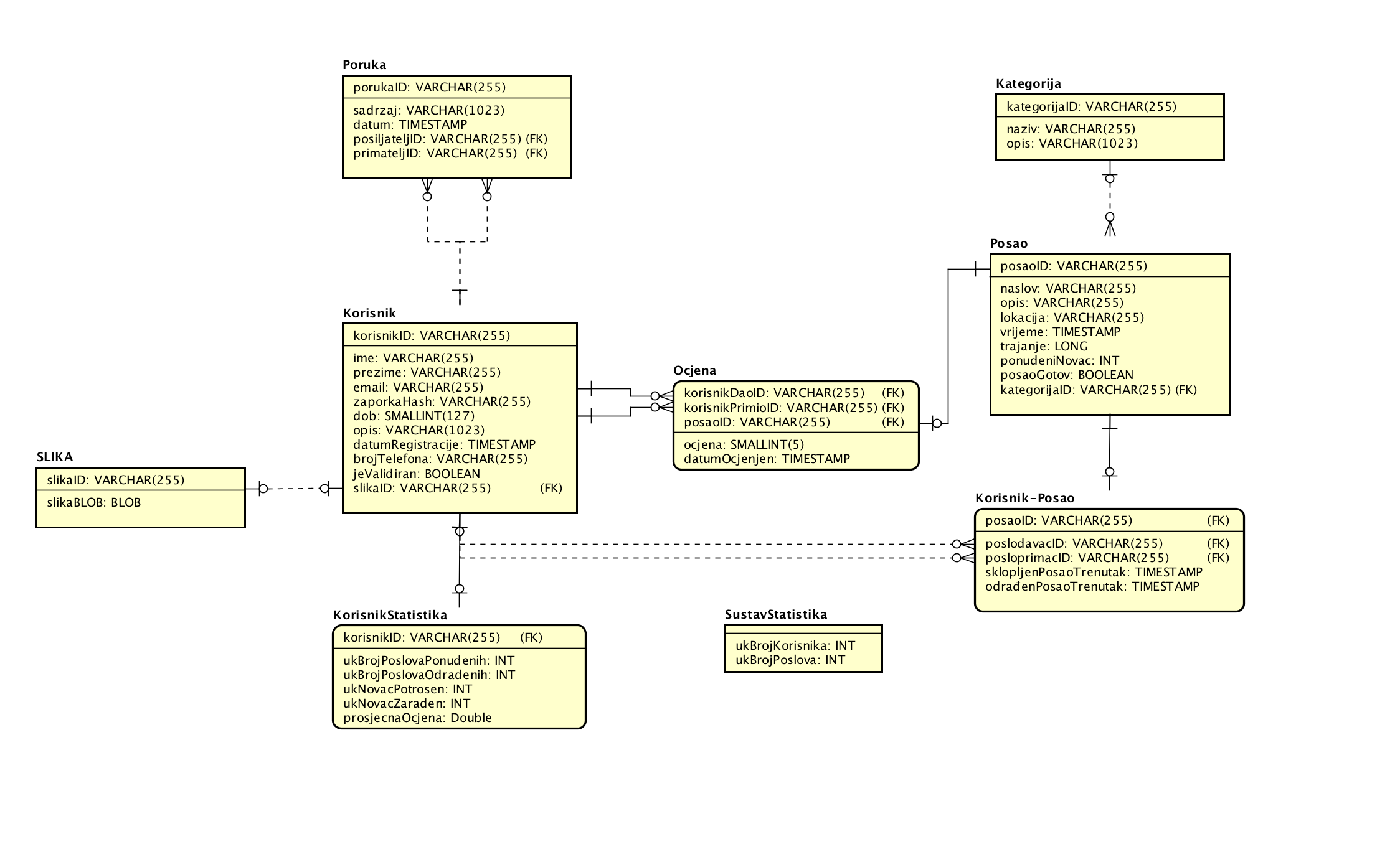
**PK{**korisnikDaoID,korisnikPrimioID,posaoID**}**

Entitet "Ocjena" opisuje svaku ocjenu korisnika. Entitet pamti koji je korisnik kojem korisniku dao ocjenu i za koji posao. Korisnik može ocijeniti drugog samo ako je između njih sklopljen posao. Ocjena je cijeli broj između 1 i 5. Entitet pamti trenutak kada je ocjena dodijeljena.

**SustavStatistika**

* ukBrojKorisnika: INT
* ukBrojPoslova: INT

Entitet "SustavStatistika" pamti osnovne informacije o sustavu. Pamti se ukupan broj korisnika aplikacije i ukupan broj poslova u bazi podataka.

**ER DIJAGRAM**