

Rehab
Aplikacija za odvikavanje od pušenja
Arhitekturni projekat
Verzija 2.0

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

Pregled izmena

Datum	Verzija	Opis	Autor
23.04.2017.	1.0	Inicijalna verzija	Marija Stamenković Bojan Milosavljević Mirjana Mičić
01.06.2017	2.0	Izvršna verzija	Marija Stamenković Bojan Milosavljević Mirjana Mičić

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

Sadržaj

1.	Cilj dokumenta	5
2.	Opseg dokumenta	5
3.	Reference	5
4.	Predstavljanje arhitekture	5
5.	Ciljevi i ograničenja arhitekture	5
6.	Pogled na slučajeve korišćenja	5
6.1	Dijagrami slučajeva korišćenja	6
6.2	Kratak opis slučajeva korišćenja	7
6.2.1	Pokretanje aplikacije prvi put	7
6.2.2	Unos datuma prestanka	7
6.2.3	Unos cene pakle	7
6.2.4	Unos količine popušenih cigareta dnevno	7
6.2.5	Unos razloga za prestanak pušenja	7
6.2.6	Pregled progresa	7
6.2.7	Pregled uštede	7
6.2.8	Pregled proteklog vremena od prestanka	8
6.2.9	Pregled zdravstvenog napretka	8
6.2.10	Pokretanje igrice	8
6.2.11	Pregled motivacija	8
6.2.12	Beleženje krize	8
6.2.13	Pregled saveta za prenebegavanje krize	8
6.2.14	Promena informacija	8
6.2.15	Promena datuma prestanka	8
6.2.16	Promena cene pakle	8
6.2.17	Promena količine popušenih cigareta dnevno	8
6.2.18	Promena razloga za prestanak pušenja	8
7.	Pogled na logičku arhitekturu sistema	9
7.1	Pregled arhitekture – organizacija paketa i podsistema u slojeve	9
7.1.1	Korisnički interfejs	9
7.1.2	Aplikaciona logika	9
7.1.3	Pristup podacima	10
7.1.4	Android studio	10
7.1.5	Java	10
8.	Pogled na procese	10
8.1	Procesi	10
8.1.1	Aplikacija	10
8.1.2	Logika aplikacije	10
8.1.3	SQLite baza	11
9.	Pogled na raspoređivanje sistema	11

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

9.1	Korisnik	11
9.2	Aplikacija REHAB	11
9.3	SQLite	11
10.	Pogled na implementaciju sistema	11
10.1	Model domena	11
10.2	Komponente sistema	12
10.2.1	Komponente korisničkog interfejsa	12
10.2.2	Komponente aplikacione logike	13
10.2.3	Detaljni dijagrami komponenata aplikacione logike	13
10.2.4	Komponente za pristup podacima	15
11.	Performanse	16
12.	Kvalitet	16

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

Arhitekturni projekat

1. Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Rehab aplikacije.

2. Opseg dokumenta

Dokument se odnosi na Rehab aplikaciju koja će biti razvijena od strane TripleM tima. Namena sistema je pomoć pušačima svih dobi pri odvikavanju od pušenja. U tom cilju, Rehab pruža korisne informacije i savete, prati napredak korisnika-zdravstveni i finansijski boljitak i omogućava korisniku pregled napretka.

3. Reference

Spisak korišćene literature:

1. Rehab – Predlog projekta, Triple M-Rehab-01, V2.0, 2017, Triple M team.
2. Rehab – Planirani raspored aktivnosti na projektu, V1.0, 2017, Triple M team.
3. Rehab – Plan realizacije projekta, V1.0, 2017, Triple M team
4. Rehab – Vizija sistema, V2.0, 2017, Triple M team
5. Rehab – Specifikacija zahteva, V2.0, 2017, Triple M team

4. Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

5. Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Rehab aplikacija će biti implementirana kao android aplikacija zasnovana na java skripting jeziku i SQLite bazi podataka.
2. Rehab aplikacija će biti optimizovan za telefone koji imaju Android 4.0 ili novije verzije androida.
3. Svi zahtevi u pogledu performansi dati u moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

6. Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva.

Slučajevi korišćenja Rehab aplikacije su:

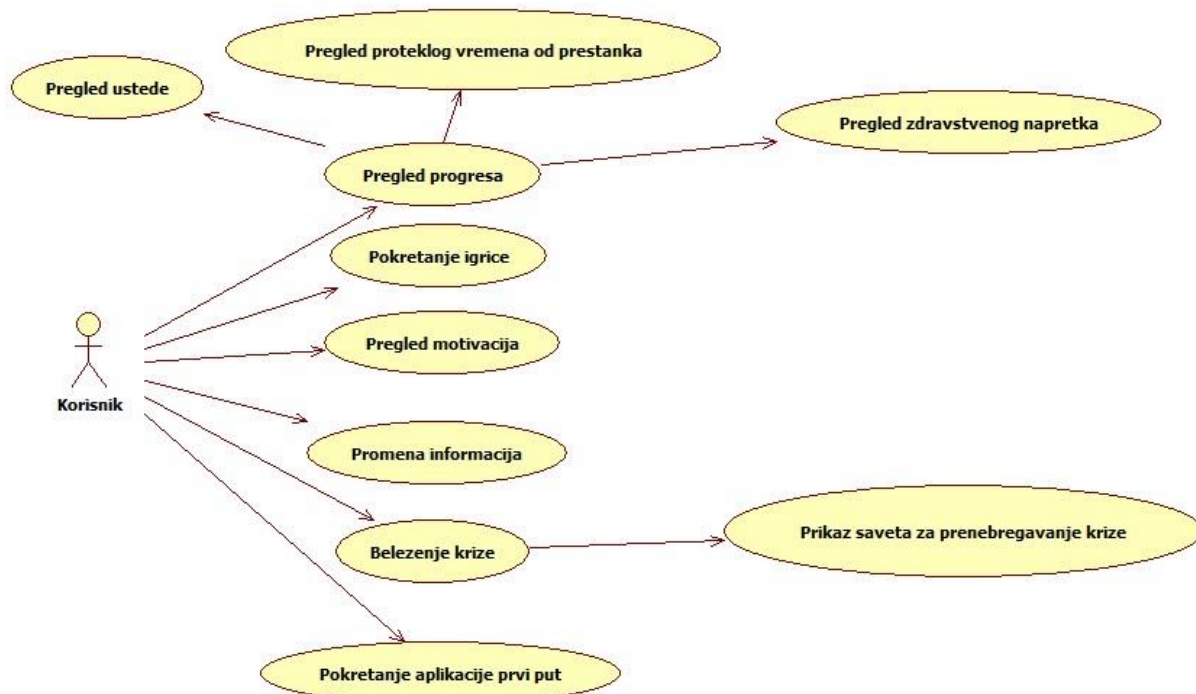
- Pokretanje aplikacije prvi put
- Pregled progresa
 - Pregled uštede
 - Pregled proteklog vremena
 - Pregled zdravstvenog napretka
- Pokretanje igrice
- Pregled motivacija
- Promena informacija
- Beleženje krize
 - Pregled saveta za prenebegavanje krize

Ove slučajevi korišćenja može da inicira korisnik.

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

6.1 Dijagrami slučajeva korišćenja

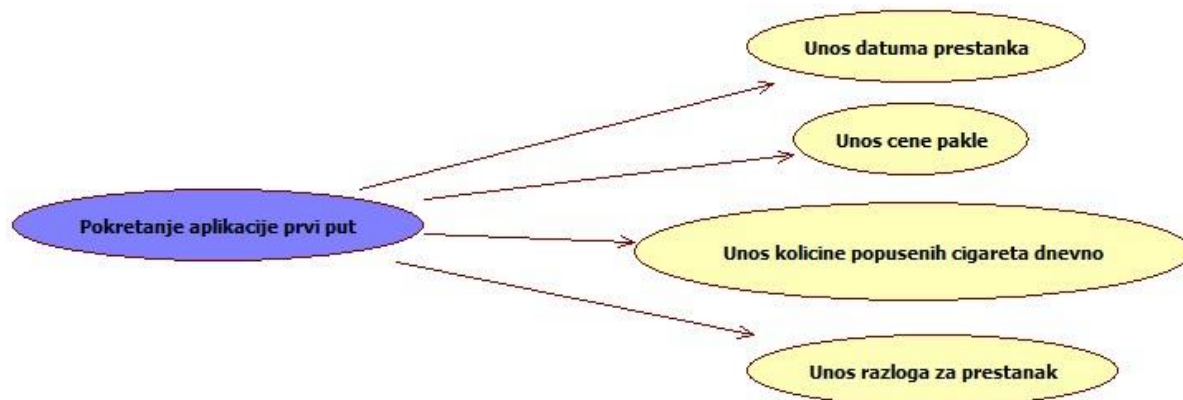
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Rehab aplikacije prikazan je na sledećoj slici:



Slučajevi korišćenja *pokretanje aplikacije prvi put* i *promena informacija* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

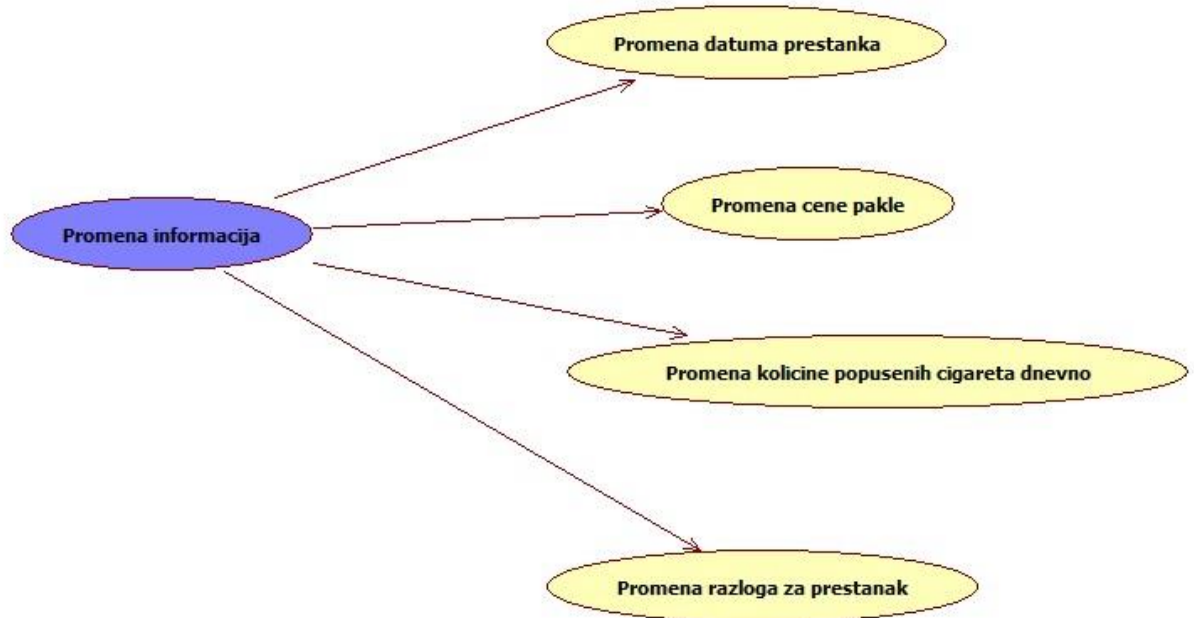
Detaljni UML dijagram za slučaj korišćenja *pokretanje aplikacije prvi put* je prikazan na sledećoj slici:

Detaljni UML dijagram za slučaj korišćenja *kreiranje, brisanje i arhiviranje članova* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *promena informacija* je prikazan na sledećoj slici:

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine



6.2 Kratak opis slučajeja korišćenja

6.2.1 Pokretanje aplikacije prvi put

Kratak opis: Korisnik prvi put pokreće aplikaciju nakon instalacije.

Akteri: Korisnik.

6.2.2 Unos datuma prestanka

Kratak opis: Korisnik unosi datum prestanka pušenja.

Akteri: Korisnik.

6.2.3 Unos cene pakle

Kratak opis: Unosi se cena pakle cigareta koje je korisnik konzumirao.

Akteri: Korisnik.

6.2.4 Unos količine popušenih cigareta dnevno

Kratak opis: Korisnik unosi koliko je cigareta dnevno konzumirao.

Akteri: Korisnik

6.2.5 Unos razloga za prestanak pušenja

Kratak opis: Unose se razlozi zbog kojih je korisnik odlučio da prestane sa pušenjem

Akteri: Korisnik.

6.2.6 Pregled progressa

Kratak opis: Omogućavanje korisniku da pregleda informacije o svom napretku u vidu zdravlja, finansija i proteklog vremena od poslednje popuštene cigarete.

Akteri: Korisnik

6.2.7 Pregled uštede

Kratak opis: Pregled ostvarene uštede, od trenutka prestanka pušenja do trenutnog datuma.

Akteri: Korisnik.

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

6.2.8 *Pregled proteklog vremena od prestanka*

Kratak opis: Pregled proteklog vremena u formatu DD/MM/YYYY mm:HH.

Akteri: Korisnik.

6.2.9 *Pregled zdravstvenog napretka*

Kratak opis: Pregled zdravstvenog napretka korisnika od trenutka poslednje popuštene cigarete u skladu sa SZO statistikom.

Akteri: Korisnik.

6.2.10 *Pokretanje igrice*

Kratak opis: Iniciranje mini-igrice.

Akteri: Korisnik.

6.2.11 *Pregled motivacija*

Kratak opis: Prikaz aktivnosti u okviru koje se nalaze motivacione poruke i ohrabrenja kao i razlozi za prestanak pušenja.

Akteri: Korisnik.

6.2.12 *Beleženje krize*

Kratak opis: Korisnik beleži trenutak u kom nastaje kriza(potreba za cigaretom).

Akteri: Korisnik.

6.2.13 *Pregled saveta za prenebrgavanje krize*

Kratak opis: Prikazuju se saveti za olakšavanje kriznog trenutka.

Akteri: Korisnik.

6.2.14 *Promena informacija*

Kratak opis: Promena informacija koje je korisnik uneo pri prvom pokretanju aplikacije.

Akteri: Korisnik

6.2.15 *Promena datuma prestanka*

Kratak opis: Modifikovanje ranije unetih podataka.

Akteri: Korisnik

6.2.16 *Promena cene pakle*

Kratak opis: Modifikovanje ranije unetih podataka.

Akteri: Korisnik

6.2.17 *Promena količine popušenih cigareta dnevno*

Kratak opis: Modifikovanje ranije unetih podataka.

Akteri: Korisnik

6.2.18 *Promena razloga za prestanak pušenja*

Kratak opis: Modifikovanje ranije unetih podataka.

Akteri: Korisnik

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

7. Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

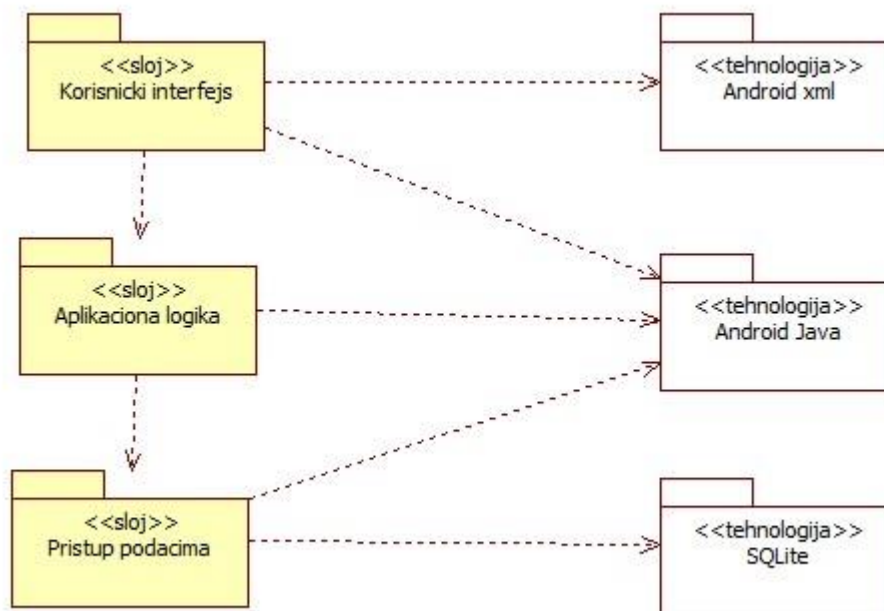
Logički pogled na Rehab aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* koristi Android studio koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji se piše u Javi zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži Java/SQLite kod koji predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve



7.1.1 Korisnički interfejs

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržani elementi Android studia koji podržava izradu izgleda aplikacije.

Sloj korisničkog interfejsa zavisi od Android studia.

7.1.2 Aplikaciona logika

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi. Sadrži java kod koje realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima.

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

7.1.3 Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži funkciju zaduženu za pribavljanje, dodavanje i ažuriranje podataka.

7.1.4 Android studio

Tehnologija Android studio definiše grafički interfejs aplikacije sa svim potrebnim funkcionalnostima koja aplikacija zahteva.

7.1.5 Java

Tehnologija Java koristi se za pisanje koda aplikacije za funkcionisanje aplikacije i rada sa bazom podataka.

8. Pogled na procese

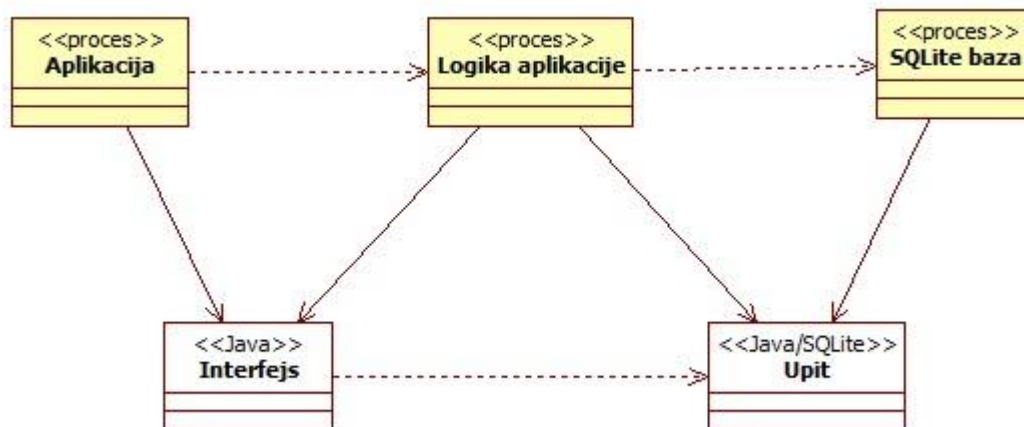
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesu i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Android aplikacija zasnovane na Javi imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Android sistema. Sa stanovišta projektanta Android aplikacije nije potrebno voditi računa o načinu rada Android sistema.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Rehab kao Android aplikacije.

8.1 Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Rehab aplikacije.



8.1.1 Aplikacija

Enkapsulira u sebi grafičke elemente aplikacije i pruža podršku za rad sa samom aplikacijom. Zavisí od logike aplikacije.

8.1.2 Logika aplikacije

Logika aplikacije je proces koji izvršava funkcionalnosti aplikacije, počev od prikaza korisničkog interfejsa, preko vršenja potrebnih izračunavanja, do logike za mini-igricu. U ovaj proces spadaju sve aktivnosti koje se izvršavaju, kao i pomoćne klase za potrebna izračunavanja. Zavisí od SQLite baze od koje dobija neophodne podatke.

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

8.1.3 SQLite baza

U ovaj proces spada klasa koja pruža funkcije za unos, modifikaciju i čitanje podataka iz baze. Baza je lokalna, tako da nema potrebe za proveravanjem njenih performansi.

9. Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja REHAB aplikacije.



9.1 Korisnik

Pristup aplikaciji REHAB se obavlja preko smartphone-a na kojima se izvršava aplikacija. Svi podaci čuvaju se u lokalnoj SQLite bazi i prikazuju na zahtev korisnika.

9.2 Aplikacija REHAB

Telefon na kome se izvršava aplikacija opslužuje jednog korisnika koji koristi funkcionalnosti aplikacije. Pored osnovnog procesa koji realizuje funkcionalnost aplikacije, postoji i proces koji realizuje prikaz interfejsa aplikacije.

9.3 SQLite

U ovom delu se realizuje pristup bazi podataka, unos, modifikovanje i preuzimanje podataka. SQLite je lokalna baza koja se čuva na telefonu.

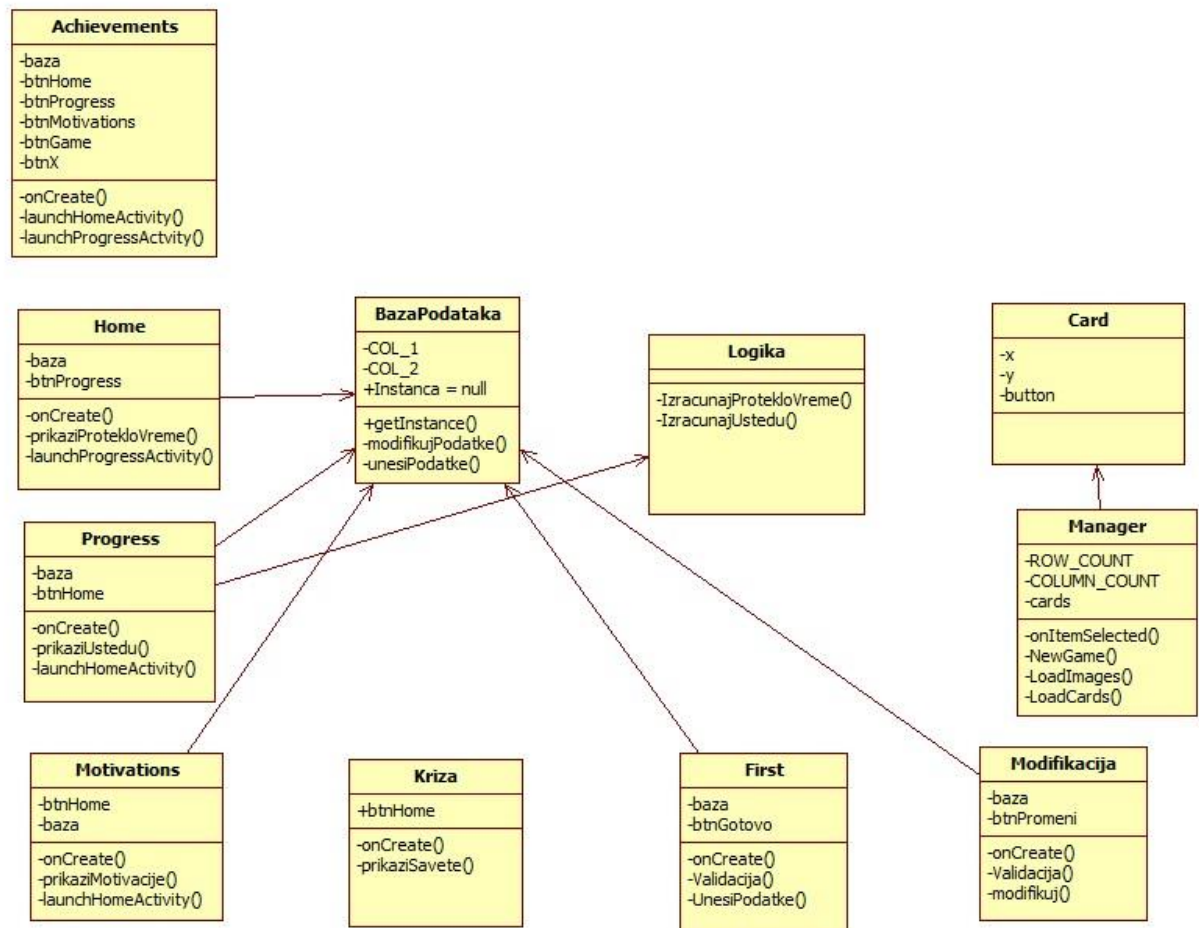
10. Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju REHAB aplikacije ovaj odeljak sadrži model domena i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

10.1 Model domena

Model domena za koji u REHAB aplikaciji projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine



10.2 Komponente sistema

Komponente sistema REHAB aplikacije su Java klase čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde se koristi pristup bazi podataka stavili smo stereotip <<Java/SQLite>>.

10.2.1 Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama tj. paketima:



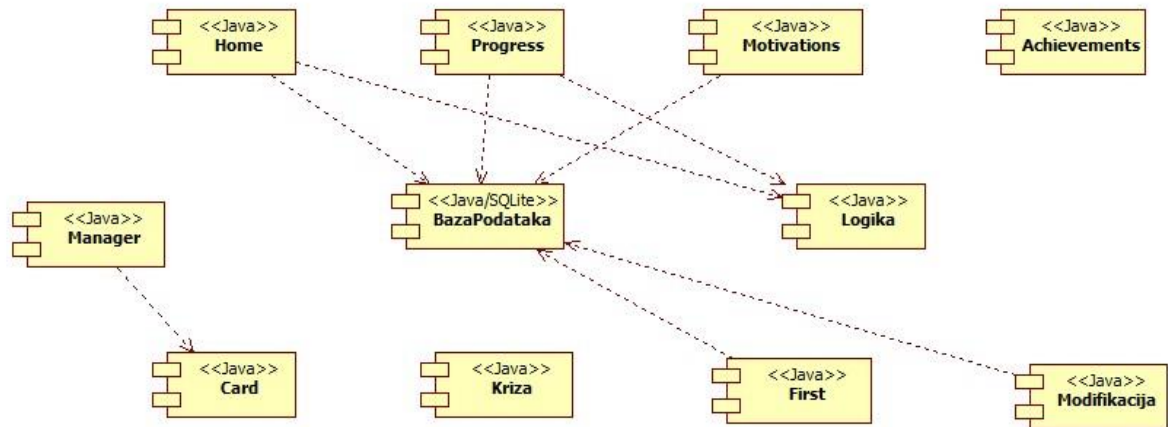
Komponenta **Activity.xml** implementira izgled različitih aktivnosti aplikacije, čiji se sadržaj menja u zavisnosti od unetih podataka.

Komponenta **Class.java** predstavlja logiku kojom se generišu komponente koje čine interfejs aplikacije.

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

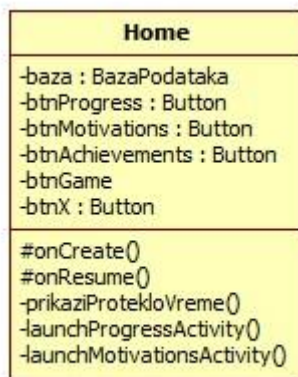
10.2.2 Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.php** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



10.2.3 Detaljni dijagrami komponenata aplikacione logike

10.2.3.1 Home



Klasa Home.class modeluje prvu, pocetnu stranicu aplikacije, omogucava prikaz pojedinih podataka i komunikaciju sa drugim stranicama kao i sam prikaz osnovne aktivnosti.

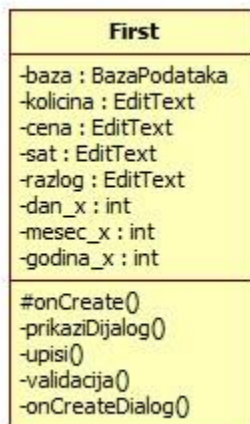
Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

10.2.3.2 Progress



Klasa Progress modeluje aktivnost koja služi za prikaz napretka korisnika, omogućava povezivanje sa drugim stranicama i koristi bazu podataka kao i objekte klase Logika za vršenje potrebnih izračunavanja, i prikazuje rezultate. Klase Motivations, Achievements i Kriza rade na sličnom obrascu kao i Progress zato ih nećemo dalje razmatrati.

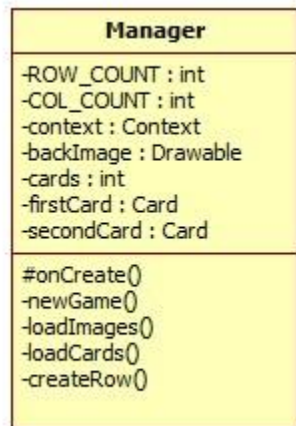
10.2.3.3 First



Klasa First služi za modelovanje aktivnosti koja se prikazuje samo jednom, kad korisnik prvi put instalira aplikaciju, radi unosa podataka neophodnih za pravilan rad aplikacije. Ima metode za prihvatanje korisnikovog unosa, validaciju istog i korišćenje baze za upis podataka. Klasa Modifikacija radi na sličnom principu, samo što ne upisuje nego modifikuje podatke iz baze te je nećemo dalje uzimati u razmatranje.

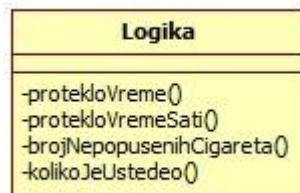
Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine

10.2.3.4 Manager



Klasa Manager koristi se za upravljanje igricom. Sadrzi sve neophodne metode za logiku i prikaz elemenata igrice, koristi se klasom Card, i kao i svaka druga klasa koja modeluje izgled aktivnosti, nasledjuje klasu AppCompatActivity.

10.2.3.5 Logika



Klasa Logika je jedinstvena u nasem sistemu, koristi se bazom podataka radi vrsenja neophodnih izracunavanja, koja se u razlicitim aktivnostima koriste radi pracenja korisnikovog napretka.

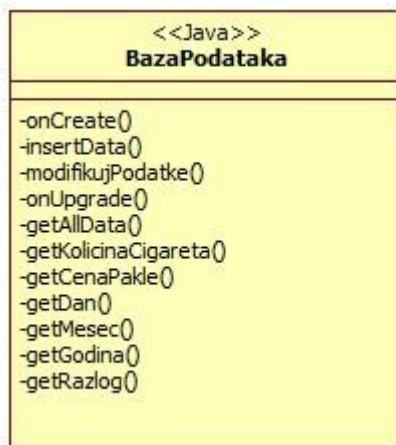
10.2.4 Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru klase **BazaPodataka.java**. Instanca ove klase se koristi u aktivnostima koje zahtevaju pristup bazi podataka.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:

Rehab	Verzija: 2.0
Arhitekturni projekat	Datum: 23.04.2017. godine



Navedene funkcije obavljaju sledeće zadatke:

- onCreate – kreiranje baze podataka, kreiranje tabele i definisanje ograničenja nad tabelom i kolonama
- insertData – unos podataka u SQLite bazu
- modifikujPodatke – modifikovanje već postojećih podataka
- onUpgrade – ukoliko dodje do promene baze, mora se promeniti i verzija, za to služi ova funkcija
- getAllData – vraća objekat klase Cursor u kom se nalaze svi podaci iz baze
- getKolicinaCigareta – cita iz baze odgovarajući podatak i vraća ga u vidu String-a
- getCenaPakle – cita iz baze odgovarajući podatak i vraća ga u vidu String-a
- getDan – cita iz baze odgovarajući podatak i vraća ga u vidu String-a
- getMesec – cita iz baze odgovarajući podatak i vraća ga u vidu String-a
- getGodina – cita iz baze odgovarajući podatak i vraća ga u vidu String-a
- getRazlog – cita iz baze odgovarajući podatak i vraća ga u vidu String-a
- getInstance – klasa je Singleton, tako da ovaj metod vraća instancu klase

11. Performanse

Aplikacija REHAB je jednokorisnička aplikacija, odabrana arhitektura u vidu performansi mora da zadovolji samo već navedeni uslov u prethodnim dokumentima da vreme učitavanja svake od aktivnosti ne sme biti preciznije 5s. Iako koristimo bazu podataka, SQLite je lokalna baza, operacije kojima se koristi nisu zahtevne toliko da ih moramo uzeti u obzir, prema tome smatramo da su zahtevane performanse zadovoljene izborom tehnologija na kojima će aplikacija biti razvijana i definisane hardverske platforme.

12. Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. REHAB aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji..
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.