**Running++  
Android aplikacija** **za trčanjeipraćenjestatistike**

**Arhitekturni projekat**

**Verzija 1.0**

**Pregledizmena**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autori** |
| 30.3.2019. | 1.0 | Inicijalna verzija | Aleksandar Stoiljković  Nikola Gocić  Lazar Stoiljković |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 4

2. Opseg dokumenta 4

3. Reference 4

4. Pregled slučajeva korišćenja 5

5. Profili korisnika 6

5.1 Administrator aplikacije 6

5.2 Guest korisnik 7

5.3 Krajnji korisnik(korisnik sa nalogom) 7

6. Opis slučajeva korišćenja 7

6.1 Praćenje aktivnosti(trčanje) 8

6.2 Prikaz trenutnih rezultata 9

6.3 Unos ličnih podataka 9

6.4 Ažuriranje ličnih podataka 9

6.5 Odjavljivanje iz aplikacije 9

6.6 Kreiranje naloga 9

6.7 Logovanje 9

6.8 Prijavljivanje preko Google/Facebook naloga 9

6.9 Pretraga ostalih korisnika 9

6.10 Dodavanje korisnika u listu prijatelja 9

6.11 Brisanje korisnika iz liste prijatelja 9

6.12 Prikaz i sortiranje rezultata 10

6.13 Prikaz dnevnih rezultata 10

6.14 Prikaz nedeljnih rezultata 10

6.15 Prikaz mesečnih rezultata 10

6.16 Sortiranje-Distanca 10

6.17 Soritranje-Vreme Trčananja 10

6.18 Sortiranje-Potrošene kalorije 10

6.19 Sortiranje-Brzina trčanja 10

6.20 Logovanje u administratorski nalog 10

6.21 Brisanje i ažuriranje korisničkih naloga 10

7. Dodatni zahtevi 11

7.1 Funkcionalnost 11

7.2 Upotrebljivost 11

7.3 Pouzdanost 11

7.4 Performanse 11

7.5 Podrška i održavanje 11

7.6 Ograničenja 11

8. Pogled na procese 13

8.1 Procesi 13

8.1.1 Android uređaj 13

8.1.2 MySQL Server 13

9. Pogled na raspoređivanje sistema 13

9.1 Klijent 13

9.2 Server 14

9.3 DBMS 14

10. Pogled na implementaciju sistema 14

10.1 Model domena 15

10.2 Šema baze podataka 15

1510.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 16

10.3.2 Komponente aplikacione logike 17

10.3.3 Komponente za pristup podacima 19

11. Performanse 20

12. Kvalitet 20

**Arhitekturni projekat**

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Running++ aplikacije.

# Opseg dokumenta

Dokument se odnosi na Running++ Andoid aplikaciju koja će biti razvijena od strane AMGTeam-a.Namena aplikacije je praćenje i prikaz statistike trčanja.

# Reference

Spisak korišćene literature:

1. Running++ - Predlog projekta, PredlogProjekta-Running++, V1.0, 2019, AMG
2. Running++ - Vizija sistema, VizijaSistema-Running++, V1.2, 2019, AMG
3. Running++ - Plan realizacije, PlanRealizacije-Running++, V1.0, 2019, AMG
4. Running++- Planirani raspored aktivnosti, RasporedAktivnosti, V1.0,2019,AMG
5. Running++ - Specifikacija zahteva, SpecifikacijaZahteva-Running++, V1.0, 2019 AMG

# 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.

# Ciljevi i ograničenje arhitekture

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

# Running++ aplikacija će biti implementirana kao Android aplikacija zasnovana na programskom jeziku Java i MySQL bazi podataka.

1. Korisnički deo aplikacije će biti optimizovan za Android uređaje koji ne koriste stariju verziju Android od 4.0 .
2. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva[5].

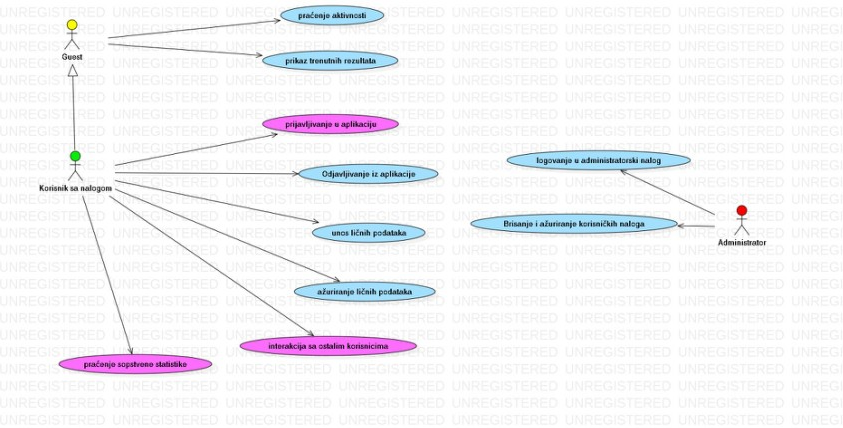
Slučajevi korišćenja Running++ aplikacije su :

* Praćenje aktivnosti(trčanje)
* Prikaz trenutnih rezultata
* Unos ličnih podataka
* Ažuriranje ličnih podataka
* Odjavljivanje iz aplikacije
* *Prijavljivanje u aplikaciju*
  + Kreiranje naloga
  + Logovanje
  + Prijavljivanje preko Google/Facebook naloga
* *Interakcija sa ostalim korisnicima*
  + Pretraga ostalih korisnika
  + Dodavanje korisnika u listu prijatelja
  + Praćenje sopstvene statistike
  + Brisanje korisnika iz liste prijatelja
  + *Sortiranje na osnovu različitih statističkih parametara*
    - Sortiranje-Distanca
    - Soritranje-VremeTrčananja
    - Sortiranje-Potrošene kalorije
    - Sortiranje-Brzina trčanja
  + *Prikaz i sortiranje rezultata*
    - Prikaz dnevnih rezultata
    - Prikaz nedeljnih rezultata
    - Prikaz mesečnih rezultata
* Logovanje u administratorski nalog
* Brisanje i ažuriranje korisničkih nalog

Ove slučajeve korišćenja mogu da iniciraju Administrator, Korisnik sa nalogom ili Guest korisnik.

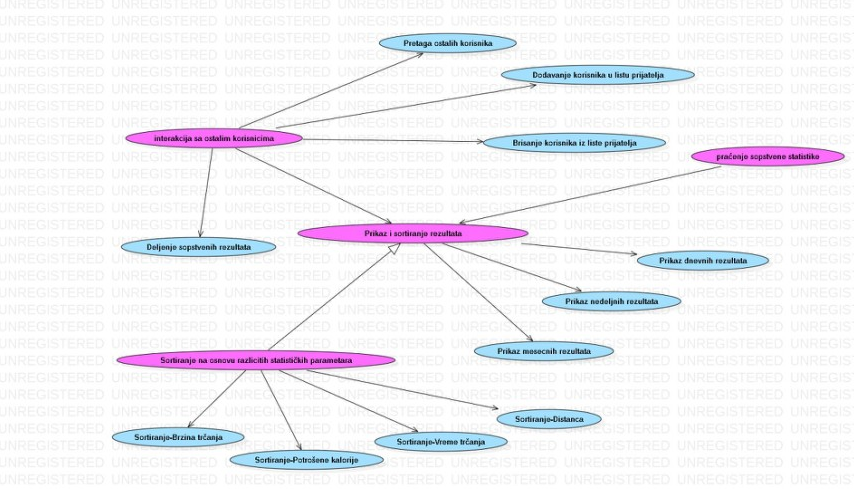
## Dijagrami slučajeva korišćenja

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

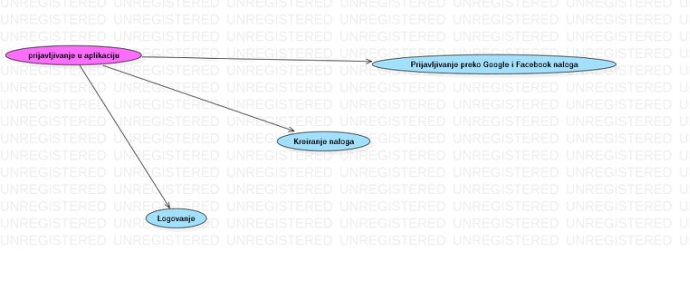


Slučajevi korišćenja *prijavljivanje u aplikaciju i brisanje* i*interakcija sa ostalim korisnicima* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Deteljni UML dijagram za slučaj korišćenja *Interakcija sa ostalim korisnicima* je prikazan na sledećoj slici:



# Detaljni UML dijagram za slučaj korišćenja *Prijavljivanje u aplikaciju* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Praćenje aktivnosti

Kratak opis: Praćenje rute kojom korisnik trči.   
Akteri: Guest korisnik i korisnik sa nalogom

### Prikaz trenutnih rezultata

Kratak opis; Prikaz osnovnih statističkih parametara (brzina trčanja,distanca,broj potrošenih kalorija, vreme trčanja).   
Akteri: Guest korisnik i korisnik sa nalogom

### Unos ličnih podataka

Kratak opis: Unos osnovnih ličnih podataka od strane korisnika koji su bitni za rad aplikacije (pol,brojgodina,broj kilograma, visina)   
Akteri: Korisnik sa nalogom

### Ažuriranjeličnihpodataka

Kratakopis: Izmenaprethodnounetihpodataka  
Akteri: Korisniksanalogom

### Odjavljivanje iz aplikacije

Kratakopis: Odjavljivanjesanaloga u aplikaciji.   
Akteri: Korisniksanalogom

### Kreiranje naloga

Kratakopis: Kreiranjenaloga za novekorisnikekojinemajunalog  
Akteri: Korisniksanalogom

### Logovanje

Kratakopis: Logovanjenakorisničkinalog  
Akteri: Korisnicisanalogom

### Prijavljivanje preko Google ili Facebook naloga

Kratakopis: Prijavljivanjepreko Google ili Facebook naloga  
Akteri: Korisnicisanalogom

### Pretraga ostalih korisnika

Kratakopis:Pretragaostalihkorisnikakojiimajukreirannalognaosnovunjihovogkorisničkogimena u ciljupostizanjakonkretnihciljeva(dodavanjekorisnika u listuprijatelja za kojećeimatimogućnostbrzogpristupa).  
Akteri: Korisnicisanalogom

### Dodavanje korisnika u listu prijatelja

Kratakopis:Dodavanjekorisnika u listuKorisnikaodinteresa  
Akteri:Korisnicisanalogom

### Brisanje korisnika iz liste prijatelja

Kratakopis: Brisanjeodabranogkorisnikaizlisteprijateljadatogkorisnika  
Akteri: Korisnicisanalogom

### Prikaz i sortiranje rezultata

Kratakopis: Prikazisoritranjepostignutihrezultata za konkretnogkorisnikailigrupukorisnika.  
Akteri: Korisnicisanalogom

### Prikaz dnevnih rezultata

Kratakopis: Prikazdnevnihrezultatakorisnika  
Akteri: Korisnicisanalogom

### Prikaz nedeljnih rezultata

Kratak opis:Prikaz nedeljnih rezultata korisnika  
Akteri:Korisnici sa nalogom

### Prikaz mesečnih rezultata

Kratak opis:Prikaz mesečnih rezultata korisnika  
Akteri:Korisnici sa nalogom

### Sortiranje-Distanca

Kratak opis:Soritranje prikazanih rezultata po predjenoj distanci  
Akteri:Korisnici sa nalogom

### Sortiranje-Vreme Trčanja

Kratak opis:Soritranje prikazanih rezultata po vremenu trčanja  
Akteri:Korisnici sa nalogom

### Sortiranje-Potrošene kalorije

Kratak opis:Soritranje prikazanih rezultata po broju potrošenih kalorija  
Akteri:Korisnici sa nalogom

### Sortiranje-Brzina Trčanja

Kratak opis:Soritranje prikazanih rezultata po prosečnoj brzini trčanja  
Akteri:Korisnici sa nalogom

### Logovanje u administratorski nalog

Kratak opis:Logovanje u administratorski nalog  
Akteri:Administratori

### Brisanje i ažuriranje korisničkih naloga

Kratak opis: Administrator ima mogućnost da obriše neki korisnički nalog ili da promeni neke podatke.   
Akteri:Administartor

# Pregled 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.

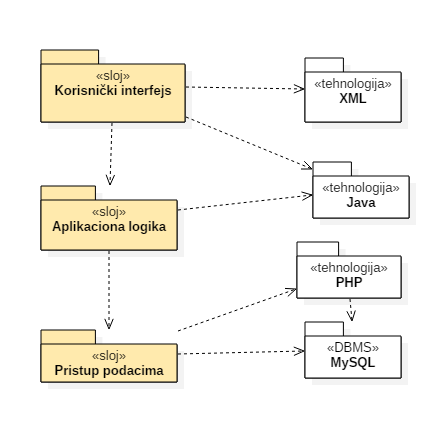
Logički pogled na Running++ aplikacije obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži forme preko kojih korisnici aplikacije komuniciraju sa aplikacijom, multimedijalni sadržaj prisan korisnicima radi poboljšanja izlgleda interfejsa aplikacije.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži Java kod zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži PHP skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregledarhitekture – organizacijapakketa I podsistema u slojeve



### Korisničkiinterfejs

sloj  
Ovajslojrealizujekorisničkiinterfejsaplikacije. U njemususadržanesveformerealizovanepomoću XML tehnologijei Android Studio okruženja. Multimedijalnisadržaj je interaktivnogtipaprilagođenkorisnicima. Ovajslojzavisi od aplikacionelogikekao I od paketa XML I Java.

### Aplikacionalogika

sloj  
Slojaplikacionelogike je srednjisloj u troslojnojarhitekturi Running++ Android aplikacije. Sadržiprogramskikodnapisan u programskomjeziku Java. Ovajslojzavisi od slojapristupapodacimakao I od paketa Java.

### Pristuppodacima

sloj  
Sloj za Pristuppodacima se nalazinadnutroslojnearhitekture Running++ aplikacije I sadrži PHP skriptezadužene za pribavljanje, dodavanje I ažuriranjepodatakakoji se čuvaju u MySQL bazipodataka. Ovajsloj ne zavisi do drugihslojeva, ali je zavisanodpaketa MySQL I PHP.

### XML

tehnologija  
Tehnologija XML-a definišegradivneelementeformikoji se prikazujunauređajimasa Android operativnimsistemom.

### Java

tehnologija  
Tehnologija Java obezbeđujemehanizam za pisanjeprogramskelogikeaplikacijeiizvršavanjefunkcijaaplikacije.

### PHP

tehnologija  
Tehnologija PHP-a obezbeđujemehanizam za pisanje I izvršavanjeskriptinastarniservera. Ove skriptepristupajubazipodataka u ciljupribavljanjaneophodnihinformacija, unospodatakaiažuriranjapodataka.

### MySQL

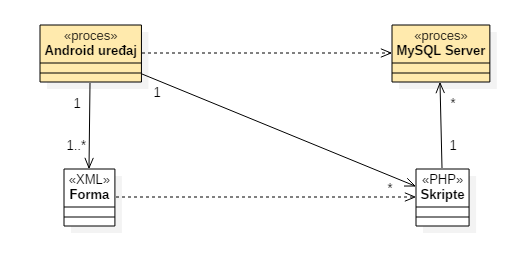
DBMS  
MySQL predstavljasistem za upravljanjebazompodatakakoji se koristi za realizaciju Running++ aplikacije.

# Poglednaprocese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa 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.

Ilustracijeradi u nastavku je datopisprocesauključenih u razvoj Running++ Android aplikacije.

## Procesi

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

### Android uređaj

Android uređaj process kojiizvršavafunkcionalnostiaplikacije za prikazformi n adatom Android uređaju. Android uređajzavisi od MySQL serverakojičuvaiobrađujeneophodnepodatke.

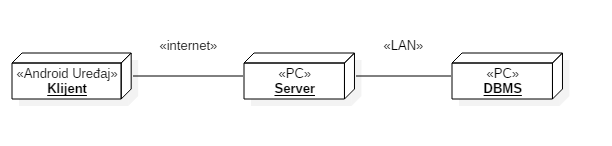
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Poglednaraspoređivanje 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 Running++ aplikacije.



## Klijent

Pristup Running++ aplikaciji se obavlja preko klijentskih uređaja koji izvršavaju aplikaciju. Za povezivanje između klijenta i Servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Server

Računar na kome se izvršava Server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Servera, na ovom računaru mogu da se izvršavaju i procesi PHP koji vrše obradu zadatih PHP skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Serverom u lokalnoj mreži (LAN).

## DBMS

DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

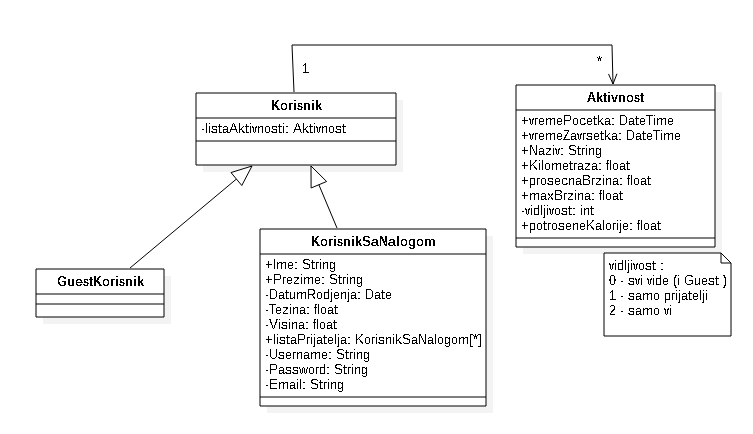
# Poglednaimplementaciju Sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju Running++ aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

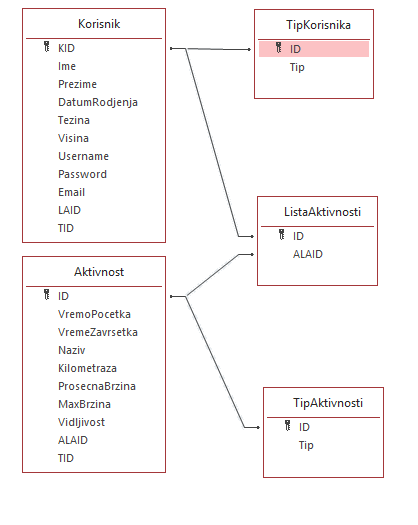
Model domena za koji se Running++aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti koje će biti implementirane.



## Šemabazepodataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *MS Access*-a, dok je migracija na MySQL obavljena pomoću *MySQLMigrationToolkit*-a.



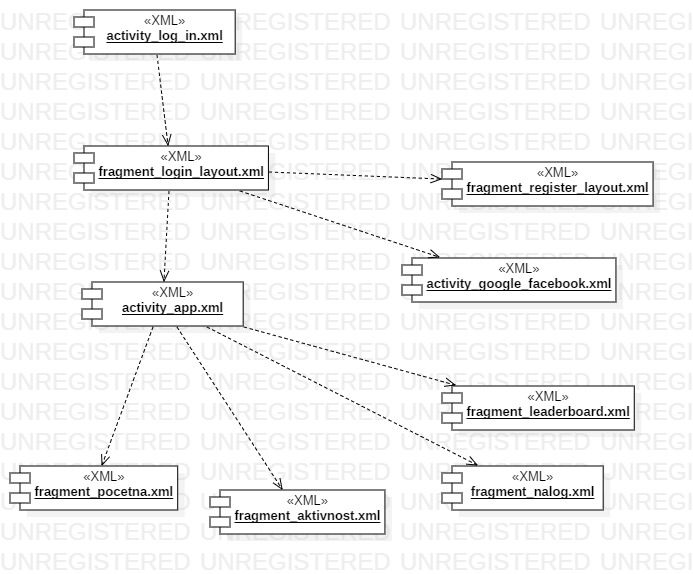
## Komponente Sistema

Komponente Sistema Running++ aplikacijesuprogramskikodovinapisani u programskomjeziku Java, kao I formenapisane u XML-u. Za ilustracijućebitikorišćeni UML dijagramikomponenti I dijagramiklasa.

### Komponente korisničkog interfejsa

Za realizaciju komponenti korisničkog interfejsa biće krišćena XML tehnologija. U nastavku biće predstavljena organizacija XML fajlova koji su korišćeni za realizaciju korisničkog interfejsa Running++ aplikacije.

Dizajn korisničkog interfejsa je obuhvaćen sledećim komponentama:

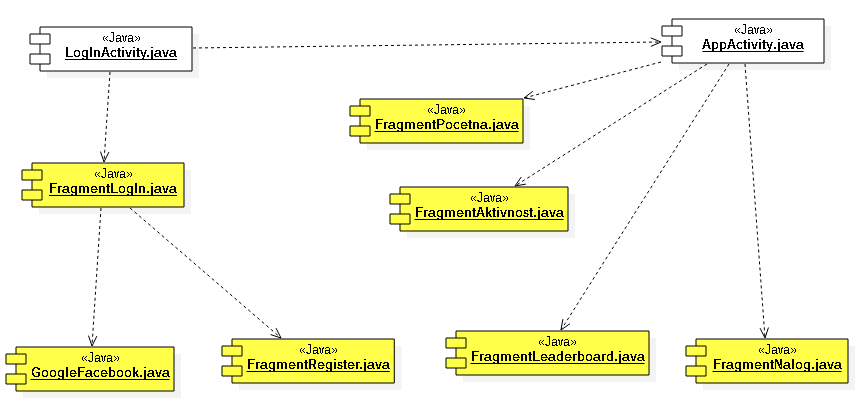
 Opis komponenti koje su predstavljene na prethodnoj slici:

1. fragemnt\_login\_layout.xml - definiše osnovne elemente za prikaz forme za prijavljivanje u aplikaciju.
2. fragment\_register\_layout.xml – fragment za opis kompenenti za registraciju korisnika koji nemaju svoj korisnički nalog
3. activity\_google\_facebook.xml – activity za prijavljivanje u aplikaciju korišćenjem facebook ili google naloga
4. fragment\_pocetna.xml – fragment koji sadrži elemente za opis početne strane aplikacije. Na početnoj strani biće prikazana lista najnovijih aktivnosti korisnika. U okviru jedne aktivnosti biće prikazani podaci o korisniku, slika koja predstavlja rutu koju je korisnik pretrčao i osnovne statističke parametre za tu rutu.
5. fragment\_aktivnost.xml – fragment koji sadrži elemnte za praćenje aktivnosti. U okviru ovog fragmenta nalaziće se mapa za prikaz lokacije korisnika . Takođe omugućeno je dinamičko prikazivanje rute u toku trčanja, kao i osnovnih statističkih podataka(trenutna pretrčana distanca,vreme trčanja, prosečna brzina,maksimalna brzina).
6. fragment\_nalog.xml - fragment u kome se prikazuju informacije o korisniku i njegovim prijateljima.
7. fragment\_leaderboard.xml - fragment za prikazivanje trenutno najboljih rezultata u različitim statističkim kategorijama.

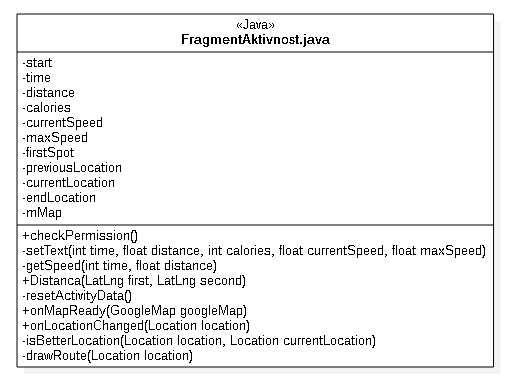
Napomena: Pored ovih osnovnih komponenti koje predstavljaju osnovu za opis korisničkog interfejsa aplikacije Running++ postojaće i dodatne podkomponente(samim tim imaćemo dodatne xml fajlove koje predstavljaju nove fragmente i activity-je ).

### Komponente aplikacione logike

Komponente koje realizuju aplikacion logiku realizovane su korišćenjem Java tehnologije. Na sledećem dijagramu prikazane su komponente za realizaciju aplikacione logike.

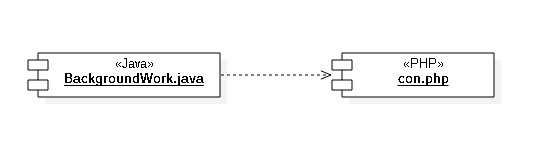
 Opis komponenti koje su predstavljene na prethodnoj slici dat je u nastavku:

1. LoginActivity.java – u ovoj klasi opisana je logika za proveru da li postoji odgovarajući korisnički nalog u bazi i u slučaju da postoji obradjuje se prijavljivanje u aplikaciju. U drugom slučaju obradjuju se sledeći dogadjaji: Registrovanje, Prijavljivanje pomoću Facebook ili Google naloga, Prijavljivanje kao Guest.
2. FragmentRegister.java – u ovoj klasi opisana je logika za registraciju korisnika, unos ličnih podataka i upis podataka u bazu.
3. GoogleFacebook.java – logika za prijavljivanje preko Facebook ili Google naloga
4. FragmentPocetna.java – u ovoj klasi je predstavljena logika za prikaz liste aktivnosti. Takodje ovde je obradjena i logika i za pretraživanje korisnika na osnovu korisničkog imena i mogućnost da se doda korisnik u listu prijatelja. Potrebno je obraditi sledeće dogadjaje u ovoj klasi: Pribavljanje liste aktivnosti i sortiranje na osnovu datuma i vremena obavljanja aktivnosti; Pribavljanje podataka o korisniku(podrazumeva se ime i prezime korisnika) i pribavljanje podataka o aktivnosti(podrazumeva se slika istrcane rute kao i statistički podaci za pretrčanu rutu) ;Prikaz liste aktivnosti ; Pretraga liste korisnika i dodavanje kosrisnika u listu prijatelja
5. FragmentNalog.java – klasa u kojoj je obradjena logika za upravljanje korisničkim nalogom. Pod tim podrazumevamo: ažuriranje korisničkog naloga(pristup bazi i ažuriranje odgovarajućih podataka u bazi), prikaz liste prijatelja(pristup bazi i pribavljanje podataka ), prikaz lične statistike(obradjuje se pristup bazi i pribavljanje potrebnih podataka i po potrebi odgovarajuća sortiranja).
6. FragmentLeaderboard.java – klasa u kojoj se samo vrši pristup bazi pribavljanje odgovarajućih podataka, sortiranje na sonovu određenih kriterijuma i tabelarni prikaz.
7. FragmentAktivnost.java – u ovoj klasi potrebno je obraditi dogadjaje za prikaz mape, prikaz trenutne lokacije i dinamičku obradu podataka u toku obavljanja aktivnosti. Na sledećoj slici predstavljen je dijagram za ovu klasu:

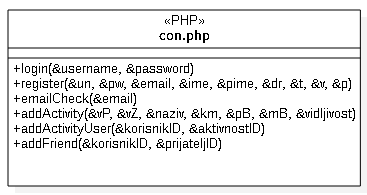


### Komponente za pristup podacima

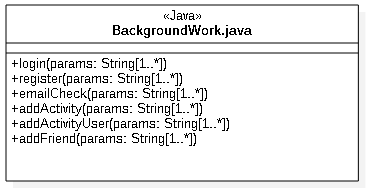
Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru PHP skripte con.php koja se nalzi na serveru. Pomenuoj skripti se pristupa iz BackgroundWork.java klase u javi, tako da su sve funkcije za pristup podacima dostupne celoj aplikaciji.



Na sledećim dijagramima pobrojane su funkcije za pristup podacima iz baze :



* login(&username,&password) – proverava da li dati username i password postoje u bazi podataka i vraća SuccessL odnosno FailedL
* register(&un,&pw,&email,&ime,&pime,&dr,&t,&v,&p) – dodaje u bazu novog korisnika sa prosleđenim podacima i vraća SuccessR odnosno FailedR
* email(&email) - proverava da li u bazi postoj dati email i vraća SuccessE odnosno FailedE
* addActivity(&vP,&vZ,&naziv,&km,&pB,&mB,&vidljivost) - dodaje novu aktivnost sa proslđenim parametrima i vraća SuccesA odnosno FailedA
* addActivityUser(&korisnikID,&aktivnostID) – datom korisniku dodaje aktivnost i vraća SuccesAU odnosno FailedAU
* addFriend(&korisnikID,&prijateljID) – prosleđenom korisniku dodaje prijatelja sa datim ID-jem i vraća SuccessF odnosno FailedF



* login(params: String[1..\*]) – poziva odgovarjuću funkciju i vraća rezultat
* register(params: String[1..\*]) – poziva odgovarjuću funkciju i vraća rezultat
* emailCheck(params: String[1..\*]) – poziva odgovarjuću funkciju i vraća rezultat
* addActivity(params: String[1..\*]) – poziva odgovarjuću funkciju i vraća rezultat
* addActivityUser(params: String[1..\*]) – poziva odgovarjuću funkciju i vraća rezultat
* addFriend(params: String[1..\*]) – poziva odgovarjuću funkciju i vraća rezultat

# Preformanse

Izabrana arhitektura softvera podržava zahteve u pogledu preciznosti trenutne lokacije uređaja i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Trenutna lokacija mora da bude precizna, greska ne sme biti veća od 2 metra.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

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

1. Running++aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati