GTCApp

Go To Chill Application

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 27.04.2021. | 1.0 | Inicijalna verzija | Nikola K. Nikola P. Sasa S. Petar P. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc71577636)

[2. Opseg dokumenta 5](#_Toc71577637)

[3. Reference 5](#_Toc71577638)

[4. Predstavljanje arhitekture 5](#_Toc71577639)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc71577640)

[6. Pogled na slučajeve korišćenja 5](#_Toc71577641)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc71577642)

[6.2 Kratak opis slučajeva korišćenja 8](#_Toc71577643)

[6.2.1 Predlog lokacija 8](#_Toc71577644)

[6.2.2 Obaveštavanje korisnika 8](#_Toc71577645)

[6.2.3 Ažuriranje naloga 8](#_Toc71577646)

[6.2.4 Dodavanje prijatelja 8](#_Toc71577647)

[6.2.5 Pregled sadržaja 8](#_Toc71577648)

[6.2.6 Pretraga lokacija i događaja 8](#_Toc71577649)

[6.2.7 Prijava 8](#_Toc71577650)

[6.2.8 Autentifikacija 8](#_Toc71577651)

[6.2.9 Kreiranje i verifikacija naloga 8](#_Toc71577652)

[6.2.10 Brisanje korisničkih naloga 8](#_Toc71577653)

[6.2.11 Odobravanje lokacije 8](#_Toc71577654)

[6.2.12 Deljenje lokacija i događaja 9](#_Toc71577655)

[6.2.13 Reportovanje 9](#_Toc71577656)

[6.2.14 Komentarisanje 9](#_Toc71577657)

[6.2.15 Ocenjivanje 9](#_Toc71577658)

[6.2.16 Brisanje lokacije 9](#_Toc71577659)

[6.2.17 Deljenje lokacija i događaja 9](#_Toc71577660)

[6.2.18 Kreiranje novog događaja 9](#_Toc71577661)

[6.2.19 Dodavanje lokacije 9](#_Toc71577662)

[6.2.20 Prijavljivanje na događaj 9](#_Toc71577663)

[6.2.21 Azuriranje događaja 9](#_Toc71577664)

[6.2.22 Slanje obavestenja o dodatom događaju. 9](#_Toc71577665)

[6.2.23 Prijavljivanje korisnika na određenu kategoriju događaja. 10](#_Toc71577666)

[6.2.24 Slanje obavestenja o izmenjenom događaju. 10](#_Toc71577667)

[Akteri koji iniciraju slucaj koriscenja: Sistem, prijavljeni korisnik 10](#_Toc71577668)

[6.2.25 Provera i uklanjane reportovanog sadrzaja. 10](#_Toc71577669)

[7. Pogled na logičku arhitekturu sistema 10](#_Toc71577670)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10](#_Toc71577671)

[7.1.1 Korisnički interfejs 11](#_Toc71577672)

[7.1.2 Aplikaciona logika 11](#_Toc71577673)

[7.1.3 Pristup podacima 11](#_Toc71577674)

[7.1.4 QML 11](#_Toc71577675)

[7.1.5 PHP 11](#_Toc71577676)

[7.1.6 C++ 11](#_Toc71577677)

[7.1.7 MySQL 11](#_Toc71577678)

[8. Pogled na procese 11](#_Toc71577679)

[8.1 Procesi 11](#_Toc71577680)

[8.1.1 Android aplikacija 12](#_Toc71577681)

[8.1.2 Desktop aplikacija 12](#_Toc71577682)

[8.1.3 PHP API 12](#_Toc71577683)

[8.1.4 Web Server 12](#_Toc71577684)

[8.1.5 MySQL Server 12](#_Toc71577685)

[9. Pogled na raspoređivanje sistema 12](#_Toc71577686)

[9.1 Klijent 13](#_Toc71577687)

[9.2 Web server 13](#_Toc71577688)

[9.3 DBMS server 13](#_Toc71577689)

[10. Pogled na implementaciju sistema 13](#_Toc71577690)

[10.1 Model domena 13](#_Toc71577691)

[10.2 Šema baze podataka 14](#_Toc71577692)

[10.3 Komponente sistema 15](#_Toc71577693)

[10.3.1 Komponente korisničkog interfejsa 15](#_Toc71577694)

[10.3.2 Komponente aplikacione logike 15](#_Toc71577695)

[*10.3.2.1* *Komponente pozadinske logike aplikacije* 15](#_Toc71577696)

[*10.3.2.2* *Komponente za komunikaciju sa Web API servisima* 16](#_Toc71577697)

[10.3.3 Komponente API servisa 18](#_Toc71577698)

[*10.3.3.1* *Servisi za pristup bazi podataka* 18](#_Toc71577699)

[10.3.3.2 Servisi za slanje mailova 19](#_Toc71577700)

[10.3.4 Servisi za uploadovanje slika 19](#_Toc71577701)

[11. Performanse 20](#_Toc71577702)

[12. Kvalitet 20](#_Toc71577703)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je specifikacija zahteva u pogledu detaljnog opisa slučajeva korišćenja GTCApp sistema.

# Opseg dokumenta

Dokument se odnosi na GTCApp sistem koji će biti razvijen od strane SquadTeam-a. GTCApp predstavlja skraćenicu za Go To Chill Application. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja za lokacije i događaje socijalnog karaktera.

# Reference

Spisak korišćene literature:

1. GTCApp – Predlog projekta, D01\_Predlog\_Projekta, V1.0, 2021, SquadTeam.
2. GTCApp – Planirani raspored aktivnosti na projektu, D03\_Raspored\_Aktivnosti V1.0, 2021, SquadTeam.
3. GTCApp – Plan realizacije projekta, D03\_Plan\_Realizacije, V1.0, 2021, SquadTeam.
4. GTCApp – Vizija sistema, D02\_Vizija\_Sistema, V1.0, 2021, SquadTeam.

# 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 ograničenja arhitekture

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

1. GTCApp će biti implementirana kao android aplikacija (korisnicki deo) i desktop aplikacija(administratorski deo) zasnovana na C++ jeziku i MySQL bazi podataka.
2. Klijentski deo GTC aplikacije će biti optimizovan za sledeće Android uredjaje: Kitkat 4.4 i noviji.
3. Administratorski deo GTCApp ce biti C++ desktop aplikacija optimizovana za Windows operativni sistem.
4. 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.

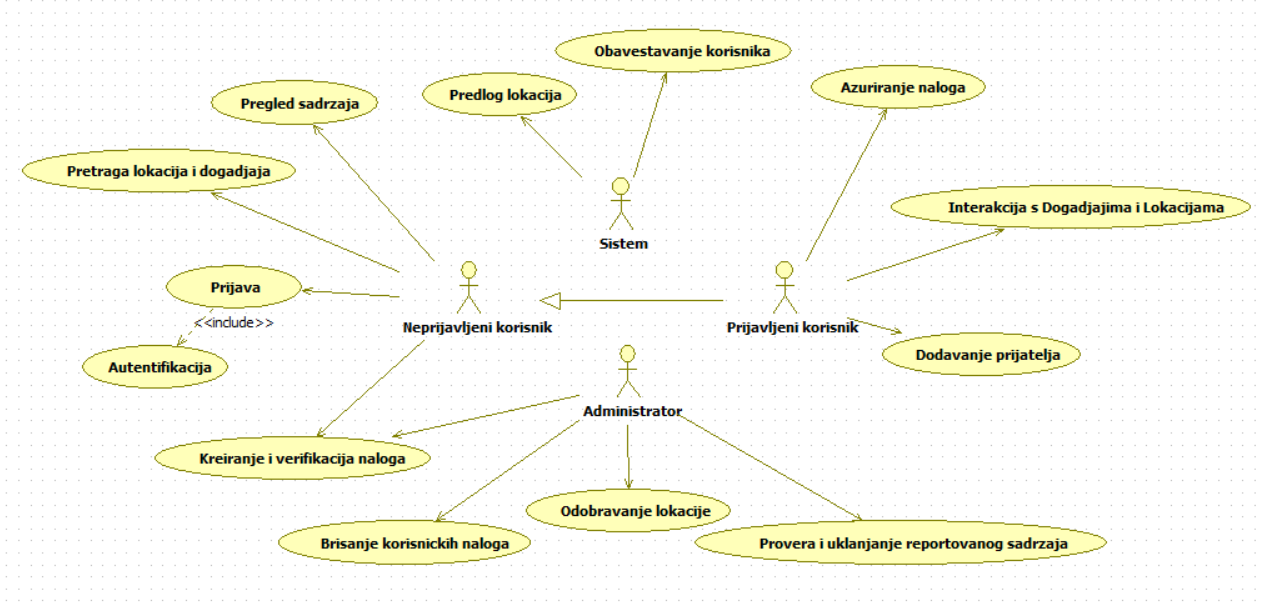
Slučajevi korišćenja GTCApp sistema su:

* *Pregled informacija*
  + Pregled osnovnih podataka o lokacijama
    - Pregled lokacija po tipu
    - Predložene lokacije
  + Pregled osnovnih podataka o događajima
    - Pregled događaja na osnovu lokacije
  + *Pregled podataka o korisnicima*
    - Pregled spiska korisnika
    - Pregled podataka o određenom korisniku.
* Prijavljivanje
* Ažuriranje korisničkih podataka
  + Izmena lozinke
  + Izmena korisničkog imena
* *Ažuriranje podataka o lokaciji*
  + Dodavanje nove lokacije
  + Brisanje postojeće lokacije
  + Dodavanje recenczije
  + Dodavanje komentara
* *Ažuriranje podataka o događaju*
  + Dodavanje novog događaja
  + Brisanje postojećeg događaja
  + Dodavanje recenczije
  + Dodavanje komentara

Ove slučajevi korišćenja mogu da iniciraju prijavljeni i neprijavljeni korisnik ili administrator.

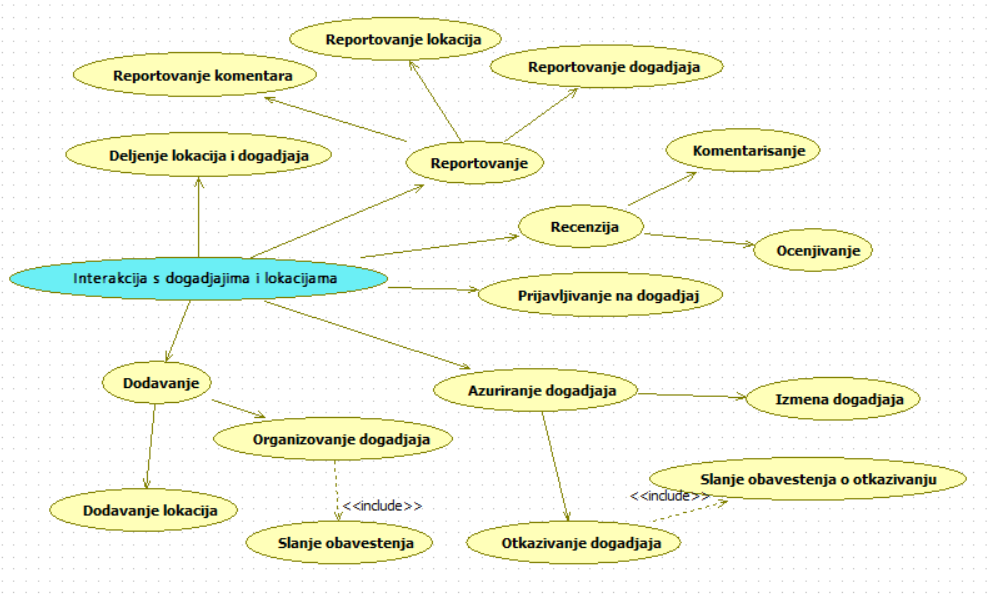
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja GTCApp sistema prikazan je na sledećoj slici:

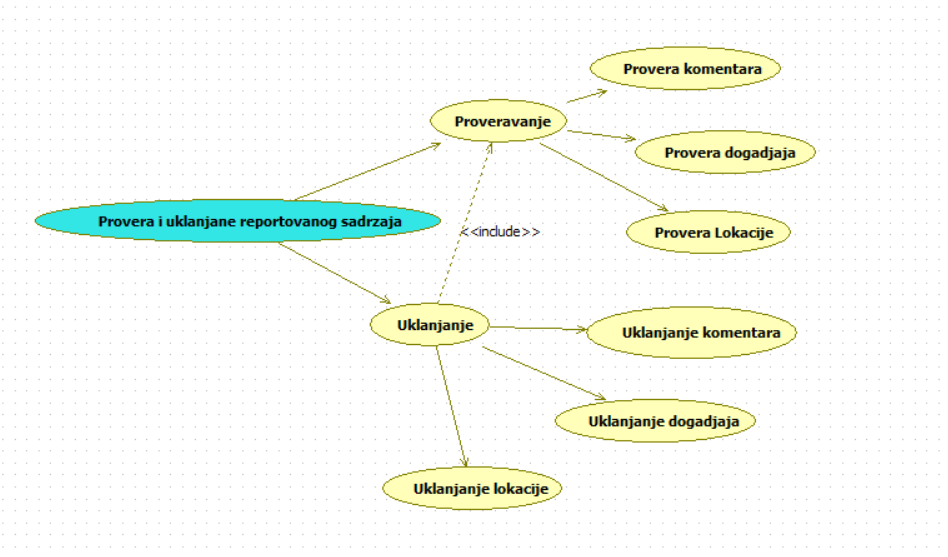


Slučajevi korišćenja koji obuhvataju funkcionalnosti vezane za tipove korisnika.

Detaljni UML dijagram za slučajeve korišćenja *Interakcija s Događajima i Lokacijama* i *Provera i uklanjanje reportovanog sadržaja* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *Provera i uklanjanje reportovanog sadržaja* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Predlog lokacija

Kratak opis: Prilikom otvaranja mobilne aplikacije, korisnik dobija predlog popularnih lokacija.

Akteri koji iniciraju slucaj koriscenja: Neprijavljeni korisnik, Prijavljeni korisnik, Sistem.

### Obaveštavanje korisnika

Kratak opis: Prilikom dodavanja ili izmene sadržaja sistem obaveštava prijavljene korisnike.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik, Sistem.

### Ažuriranje naloga

Kratak opis: Prijavljeni korisnik ima mogućnost izmene ličnih informacija na profilu.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik.

### Dodavanje prijatelja

Kratak opis: Prijavljeni korisnik ima mogućnost dodavavanja drugih korisnika u listu prijatelja.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnici.

### Pregled sadržaja

Kratak opis: Korisnici imaju mogućnost pregleda dostupnog sadržaja.

Akteri koji iniciraju slucaj koriscenja: Neprijavljeni korisnik, Prijavljeni korisnik, Administrator

### Pretraga lokacija i događaja

Kratak opis: Korisnik pretražuje događaje i lokacije po njegovim interesovanjima i mestu boravka.

Akteri koji iniciraju slucaj koriscenja: Neprijavljeni korisnici, prijavljeni korisnici.

### Prijava

Kratak opis: Neprijavljeni korisnik prelazi u prijavljenog.

Akteri koji iniciraju slucaj koriscenja: Neprijavljeni korisnik.

### Autentifikacija

Kratak opis: U zavisnosti od unetih podataka sistem vrši validiranje podataka.

Akteri koji iniciraju slucaj koriscenja: Sistem, Neprijavljeni korisnik.

### Kreiranje i verifikacija naloga

Kratak opis: Korisnik ima mogućnost kreiranja naloga, neophodno je izvršiti verifikaciju email-a.

Akteri koji iniciraju slucaj koriscenja: Neprijavljeni korisnik, Administrator.

### Brisanje korisničkih naloga

Kratak opis: Administrator ima mogućnost brisanja korisničkih naloga iz baze podataka.

Akteri koji iniciraju slucaj koriscenja: Administrator

### Odobravanje lokacije

Kratak opis: Prilikom dodavanja lokacije od strane Prijavljenog korisnika, Administrator dobija zahtev za postavljanje lokacije. Dati zahtev može prihvatiti ili odbaciti.

Akteri koji iniciraju slucaj koriscenja: Administrator

### Deljenje lokacija i događaja

Kratak opis: Korisnik može podeliti događaj ili lokaciju sa drugim korisnicima.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik.

### Reportovanje

Kratak opis: Korisnik ima mogućnost prijavljivanja komentara/lokacija/događaja za koje smatra da nisu prikladni.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik

### Komentarisanje

Kratak opis: Prijavljeni korisnik ima mogućnost ostavljanja komentara na događaje i lokacije.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik

### Ocenjivanje

Kratak opis: Korisnik ima mogućnost ocenjivanja događaja i lokacija, biranjem ocene od 1 do 5.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik.

### Brisanje lokacije

Kratak opis: Nakon dobijene prijave za određenu lokaciju administrator proverava istu i utvrđuje da li će odbaciti prijavu ili će obrisati lokaciju.

Akteri koji iniciraju slucaj koriscenja: Administrator.

### Deljenje lokacija i događaja

Kratak opis: Korisnik ima mogucnost da ostale korisnike obavesti o događajima i lokacijama

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik

### Kreiranje novog događaja

Kratak opis: Kreiranje novog događaja od strane prijavljenog korisnika.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik

### Dodavanje lokacije

Kratak opis: Korisnik dodaje novu lokaciju popunjavajuci formu za dodavanje lokacije.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik, administrator.

### Prijavljivanje na događaj

Kratak opis: Prijavljeni korisnik izrazava zelju za informisanjem o određenoj grupi događaja.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik.

### Azuriranje događaja

Kratak opis: Korisnik kreator događaja menja informacije o događaju.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik.

### Slanje obavestenja o dodatom događaju.

Kratak opis: Obavestavanje korisnika prijavljenih na određenu kategoriju događaja.

Akteri koji iniciraju slucaj koriscenja: Sistem, prijavljeni korisnik.

### Prijavljivanje korisnika na određenu kategoriju događaja.

Kratak opis: Izrazavanje zelje od strane korisnika za informisanje o kreiranju određene kategorije događaja.

Akteri koji iniciraju slucaj koriscenja: Prijavljeni korisnik

### Slanje obavestenja o izmenjenom događaju.

Kratak opis: Obavestavanje korisnika prijavljenih na određeni događaj o izmeni događaja.

### Akteri koji iniciraju slucaj koriscenja: Sistem, prijavljeni korisnik

### Provera i uklanjane reportovanog sadrzaja.

Kratak opis: Korisnici imaju mogucnost reportovanja sadrzaja, posao administratora je da to proveri ukloni ako ima potrebe.

Akteri koji iniciraju slucaj koriscenja: Administrator.

# 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.

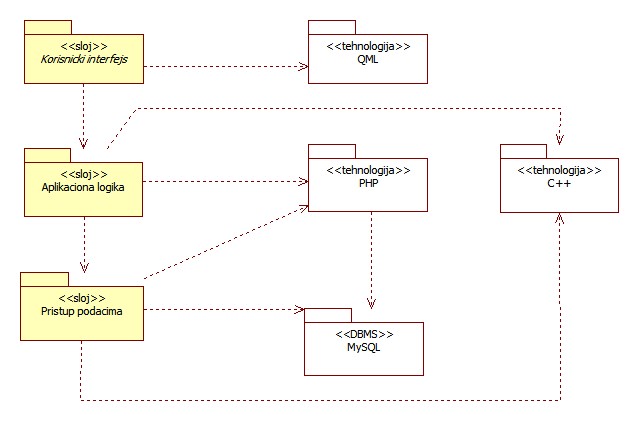
Logički pogled na GTCApp obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži QML stranice i multimedijalni sadržaj koji realizuju grafički dizajn preko kog korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži C++ klase i PHP skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

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

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržane sve QML stranice i multimedijalni sadrzaj.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa QML.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi GTCApp aplikacije. Sadrži PHP skripte kao i C++ klase 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, PHP i C++ paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži PHP skripte i C++ klase zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa C++, PHP i MySQL baza podataka .

### QML

tehnologija

Tehnologija QML-a definiše strukturu grafickog interfejsa aplikacije koja omogućava prikaz informacija i realizaciju formi za unos i ažuriranje podataka.

### PHP

tehnologija

Tehnologija PHP-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte se koriste prilikom registracije korisnika za slanje verifikacionih email-ova i za pristup bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### C++

tehnologija

Tehnologija C++-a obezbeđuje mehanizam za pisanje klasa i funkcija, koje se koriste za realizaciju svih pozadinskih funkcionalnosti aplikacije. Kao sto je komunikacija korisnickog interfejsa sa bazom podataka.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju GTCApp aplikacije.

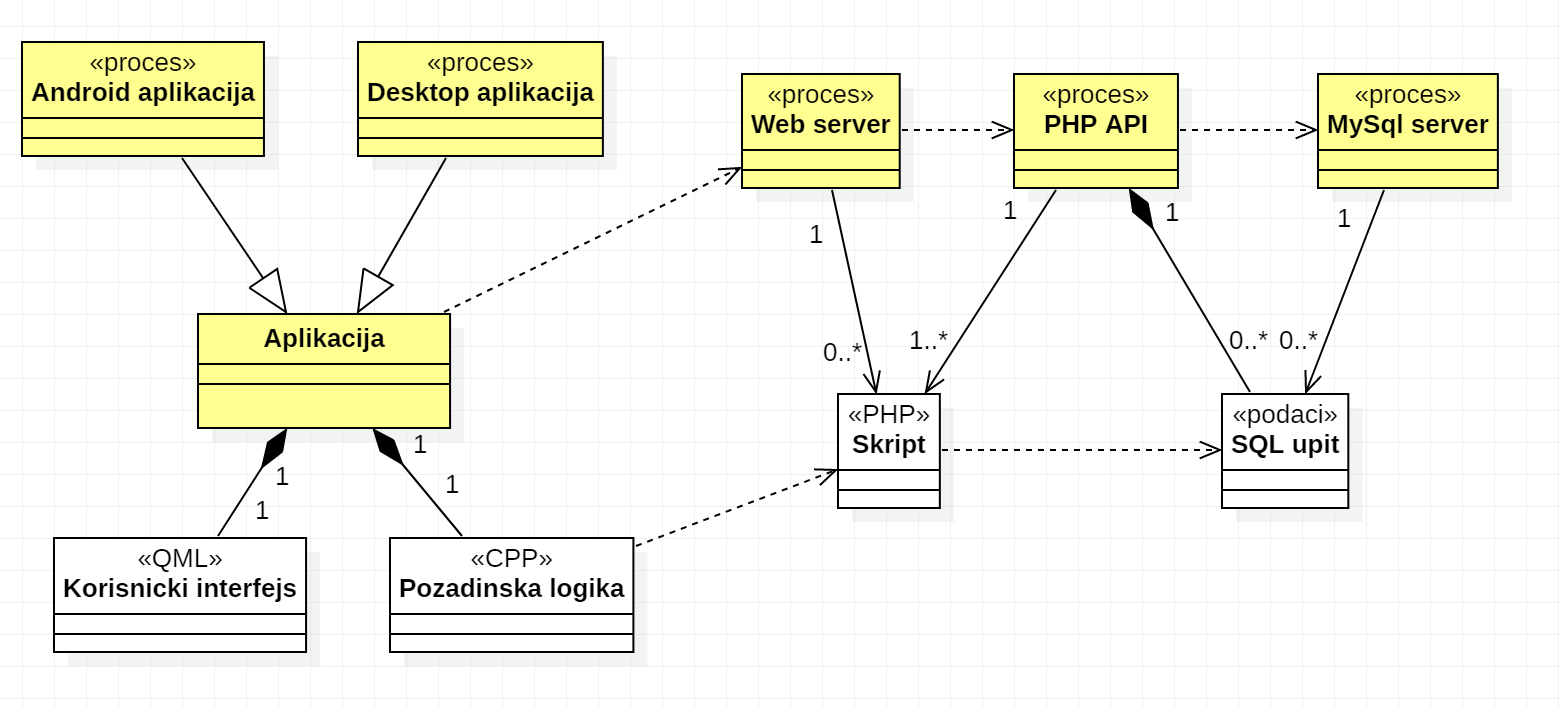
# Pogled na procese

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.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje GTCApp aplikacije.

## Procesi

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



### Android aplikacija

Android aplikacija je proces čija je osnovna funkcija omogućavanja interfejsa prema korisniku. Sastoji se od dva osnovna dela. Prvi deo predstavlja interfejs koji je pisan u QML jeziku. Pozadinska logika predstavlja drugi deo i služi za obradu podataka i komunikaciju sa API-jem. Komunikaciju ostvaruje HTTP zahtevima.

### Desktop aplikacija

Desktop aplikacija je proces koji omogućava administratoru kontrolu celokupnog sistema. Izvršava se na Windows platformi. Komunikaciju sa bazom podataka ostvaruje preko PHP API servisa.

### PHP API

PHP API proces obavlja posao obrade zadatog PHP skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Aplikaciji. Za izvršenje PHP skripta ovaj proces može da zahteva usluge MySQL servera-a. Komunikacija između PHP procesa i MySQL servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

### Web Server

Web server je proces koji ima zadatak da izvršava PHP skripte i time omogućava komunikaciju između baze podataka i aplikacije. Takođe, njegova ulaga je i skladištenje neophodnih fajlova i datoteka.

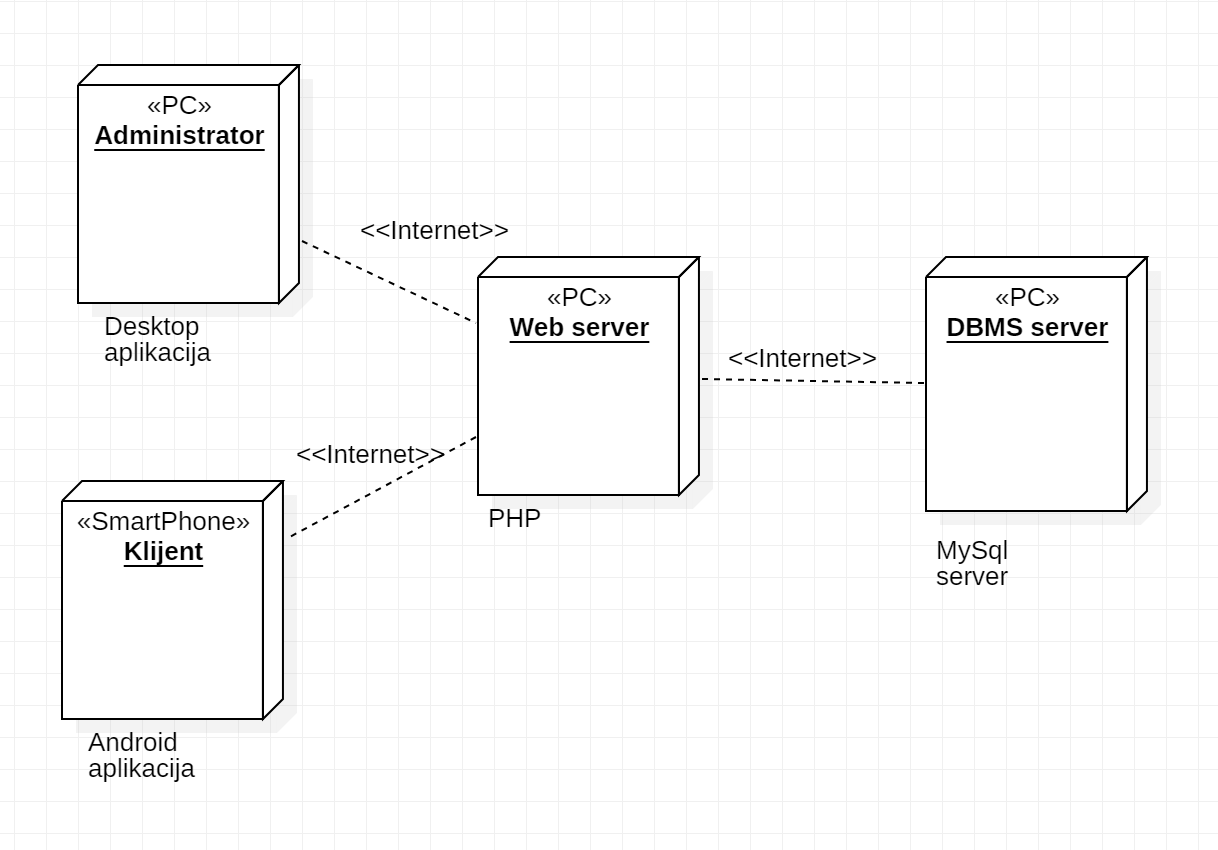
### MySQL Server

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.

# 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 GTCApp sistema.



## Klijent

Pristup GTC aplikaciji se vrši preko pametnih telefona sa Android platformom. Potrebno je instalirati aplikaciju. Za povezivanje sa serverom, koristi se internet infrastruktura, što omogućava korišćenje nezavisno od lokacije klijenta.

## Web server

Mašina na kojoj se pokreće Web server može opsluživati veći broj korisnika koji šalju zahteve putem Internet konekcije. Pored osnovnih funkcoonalnosti, na ovom Web serveru se izvršavaju dodatne PHP skripte koje omogućavaju određene servise klijentskoj aplikacijji. Web server je putem interneta povezan sa MySql bazom podataka.

## DBMS server

DBMS server je mašina na kojoj se izvršava MySql Server. Njegova uloga je omogućavanje osnovnih funkcionalnosti za upravljanje bazom podataka. Pristup DBMS serveru imaju samo određene skripte koje predstavljaju API za klijentsku aplikaciju.

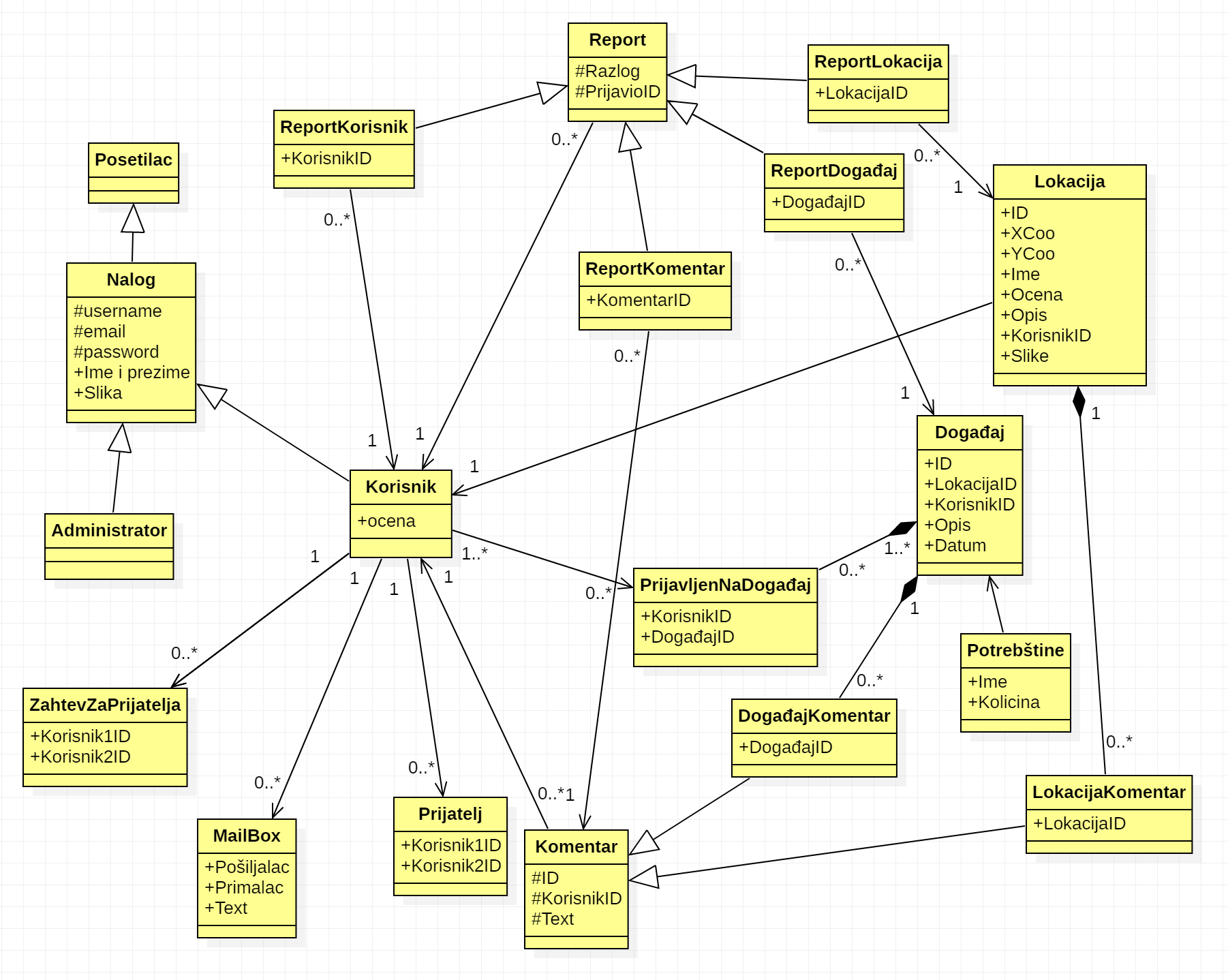
# Pogled na implementaciju sistema

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

## Model domena

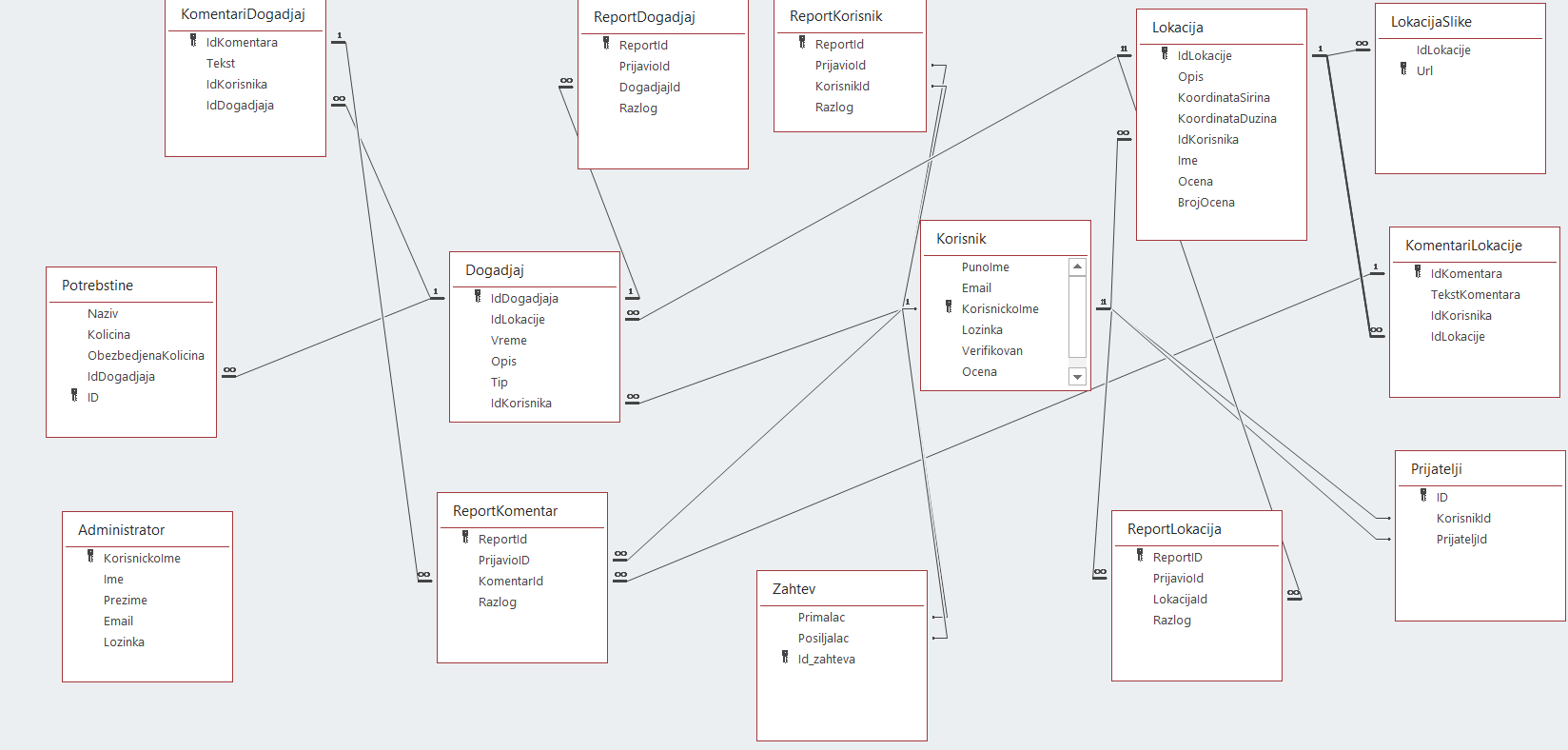
Model domena za koji se GTCApp sistem projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

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 *MySQL Migration Toolkit*-a.



## Komponente sistema

Komponente GTCApp sistema će biti date u tri osnovne grupe i biće prikazane po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML diagrami komponenti i diagrami klasa.

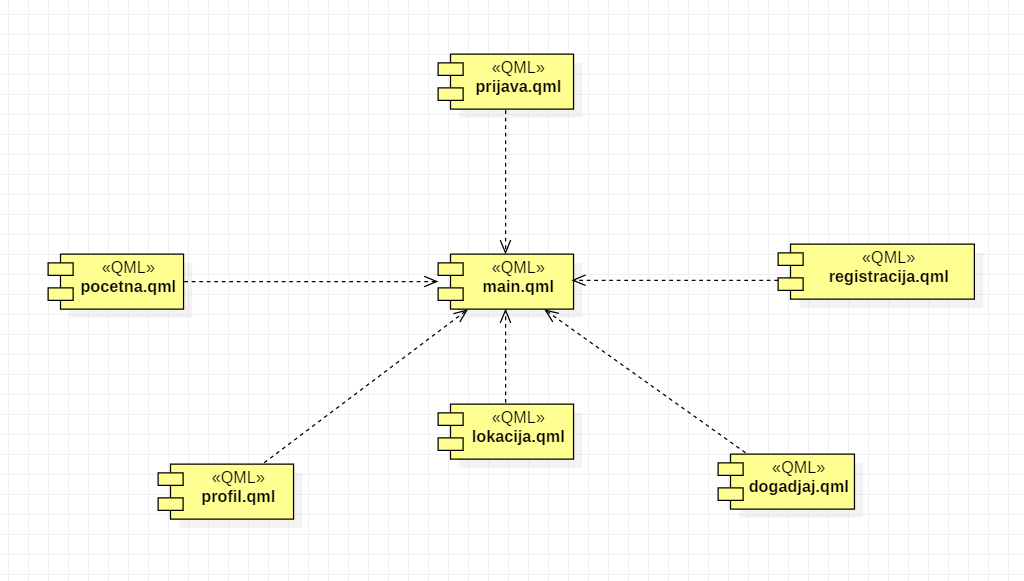
Osnovne grupe:

1. Komponente korisničkog interfejsa
2. Komponente pozadinske logike
3. Komponente API servisa

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je rađen u QML programskom jeziku.

U nastavku su dati dijagrami komponenata korisničkog interfejsa:



Komponenta **main.qml** je osnovna komponenta prikaza koja se otvara prilikom pokretanja aplikacije i služi za otvaranje ostalih komponenti.

Komponenta **pocetna.qml** korisniku daje početni prikaz pri pokretanju aplikacije.

Komponenta **prijava.qml** predstavlja korisnički interfejs za prijavljivanje.

Komponenta **registracija.qml** predstavlja korisnički interfejs za kreiranje novog naloga.

Komponenta **dogadjaj.qml** daje prikaz svih detalja o određenom događaju.

Komponenta **lokacija.qml** daje prikaz informacija i slika o dodatoj lokaciji.

Komponenta **profil.qml** omogućava korisniku pregled podataka o svom profili i izmenu istih.

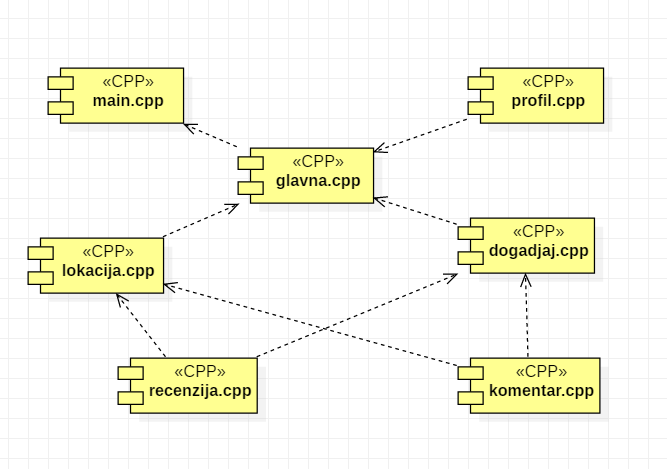
### Komponente aplikacione logike

Komponente koje realizuju pozadinsku logiku se mogu podeliti u više grupa.

U nastavku će biti dati UML dijagrami i objašnjenja istih.

#### *Komponente pozadinske logike aplikacije*

Na sledećem dijagramu su prikazane osnovne komponente koje predstavljaju pozadinsku logiku aplikacije.



Komponenta **main.cpp** se pokreće pri startovanju aplikacije.

Komponenta **glavna.cpp** služi za obradu asinhronih događaja i pozivanje drugih komponenti.

Komponenta **profil.cpp** objedinjuje sve funkcionalnosti vezane za korisnički nalog.

Komponenta **lokacija.cpp** služi kao model za lokaciju koja se pamti u bazi.

Komponenta **dogadjaj.cpp** predstavlja model za dogadjaj.

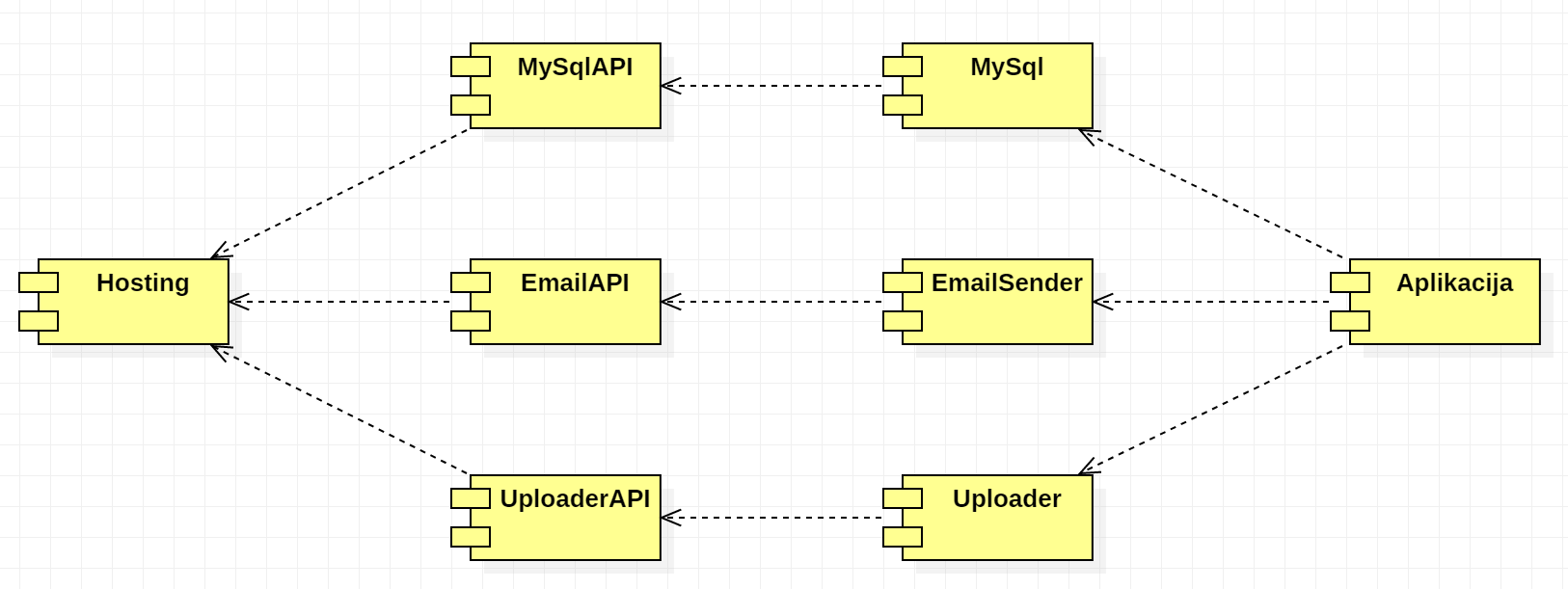
Komponenta **recenzija.cpp** obuhvata funkcionalnosti ocenjivanja.

Komponenta **komentar.cpp** predstavlja model komentara i sadrži osnovne funkcionalnosti za postavljanje i prikaz.

#### *Komponente za komunikaciju sa Web API servisima*

Ove komponente omogučavaju povezivanje android i desktop aplikacije sa Web servisima.

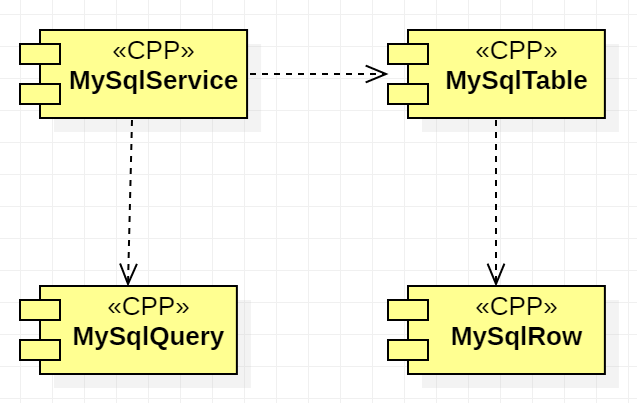
U nastavku je prikazan opšti UML diagram:



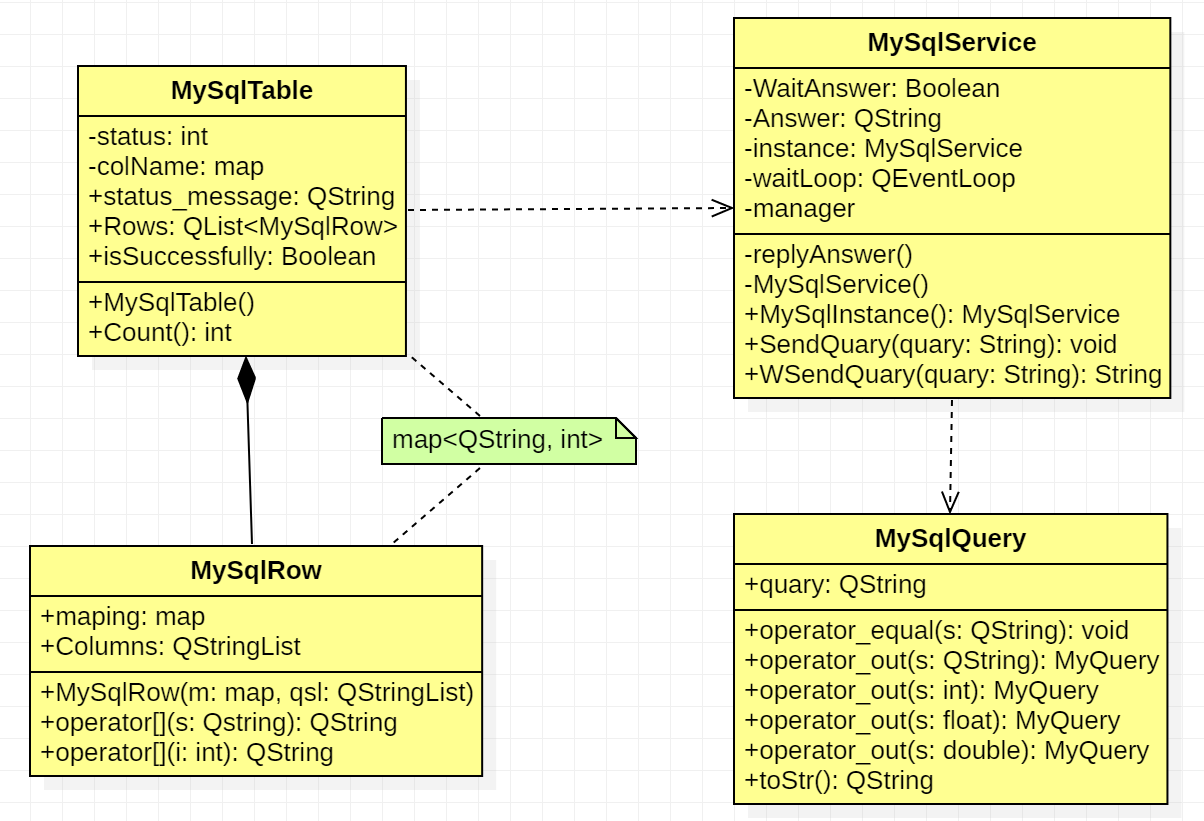
U nastavku će biti detaljno prikazana sva tri segmenta: MySql, EmailSender i Uploader.

##### *Komponente za komunikaciju sa MySql bazom podataka*

Na sledećem dijagramu komponenata su prikazane sve komponente koje služe za komunikaciju sa API komponentama na Web serveru.

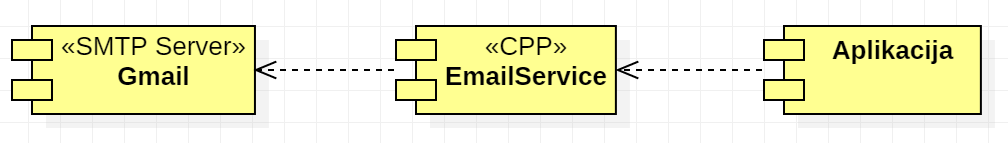


Komponenta **MySqlService** je glavna i služi za ostvarivanje komunikacije sa Web API-jem. Ima mogućnosti da šalje zahteve, prima i obrađuje rezultate. Ostale tri služe za formatiranje i serializaciju podataka. Na sledećem dijagramu klasa se mogu videti detalji implementacije.

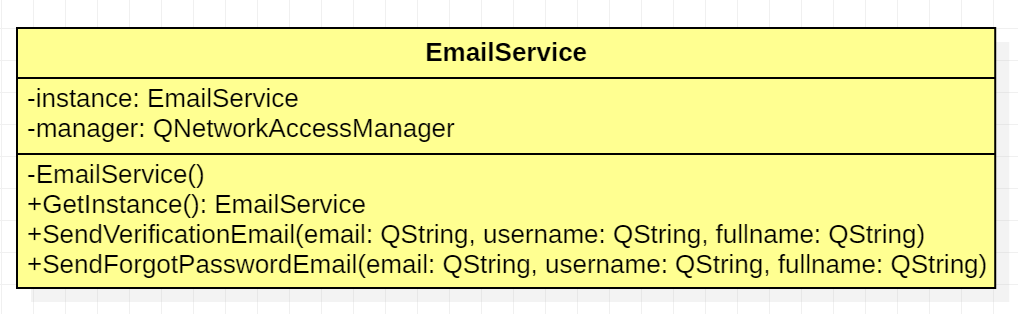


##### *Komponente za slanje Email-ova*

Za slanje emailova koristi se komponenta EmailService. Ova komponenta komunicira direkno sa Web API-jem i pruža uslugu slanja neophodnih mailova. Prikazan je UML diagram komunikacije.



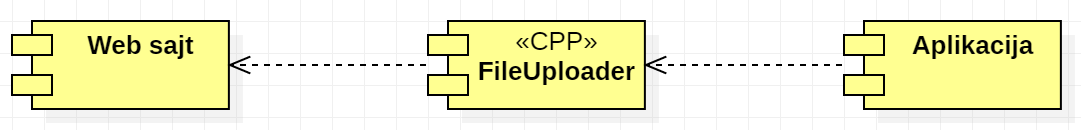
U nastavku je prikazan interfejs ove komponente.



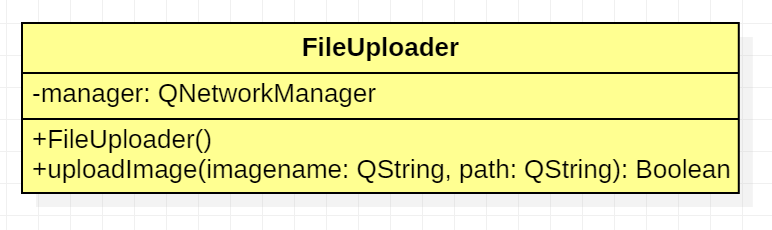
Ova komponenta je realizovana po Singleton obrazcu.

##### *Komponente za Upload-ovanje slika*

Za uploadovanje slika se koristi komponenta FileUploader. Predstavlja posrednika između Web API-ja i Aplikacije. Dijagram komponenata:



U nastavku je prikazan interfejs komponente FileUploader.



FileUploader omogućava čuvanje korisničkih slika sa profila kao i slika dodatih u aplikaciji na našem web sajtu. U bazi podataka se čuva samo ime dodate slike.

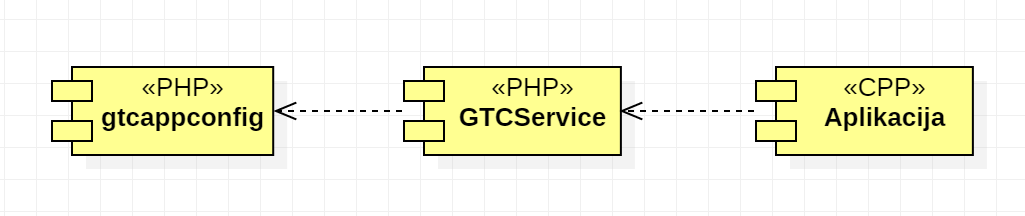
### Komponente API servisa

Komponente API servisa se dele u tri grupe.

1. Servisi za pristup bazi podataka.
2. Servisi za slanje mailova.
3. Servisi za uploadovanje slika.

#### *Servisi za pristup bazi podataka*

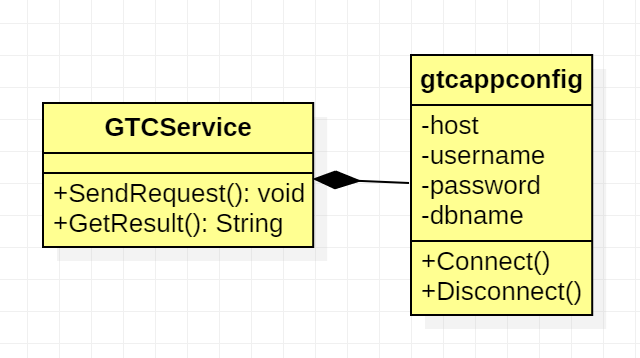
Na sledećem UML diagramu su prikazane komponente koje obezbeđuju API za komunikaciju između aplikacije i baze podataka.



Aplikacija šalje POST web api zahtev komponenti **GTCService**. U zahtevu se nalazi upit za izvršavanje.

GTCService korisni komponentu gtcappconfig za konektovanje na bazu podataka. Nakon konektovanja izvršava dobijeni upit. Rezultate upita formatira u unapred definisani format i vraća kao odgovor aplikaciji.

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

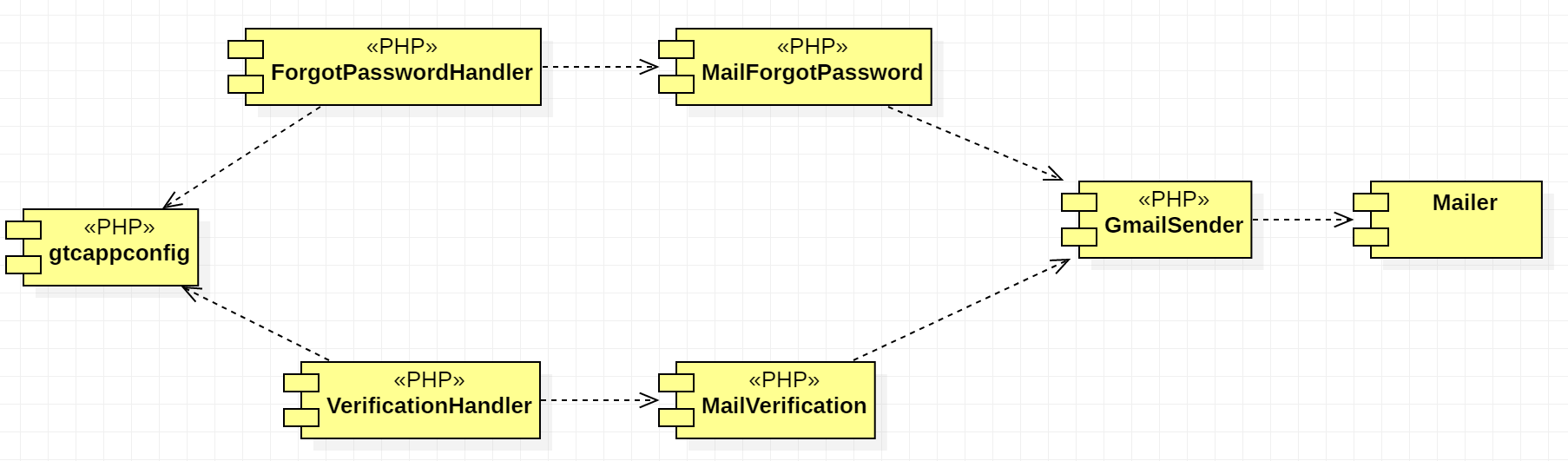


Navedene funkcije obavljaju sledeće zadatke:

* Connect() – Vrši konektovanje na bazu podataka u zavisnosti od parametara.
* Disconnect() – Vrši diskonektovanje sa baze podataka.
* SendRequest() – Šalje SQL upit, koji je prosleđen iz aplikacije, ka bazi podataka.
* GetResult() – Obrađuje odgovor sa baze i prosleđuje aplikaciji u odgovarajućem formatu.

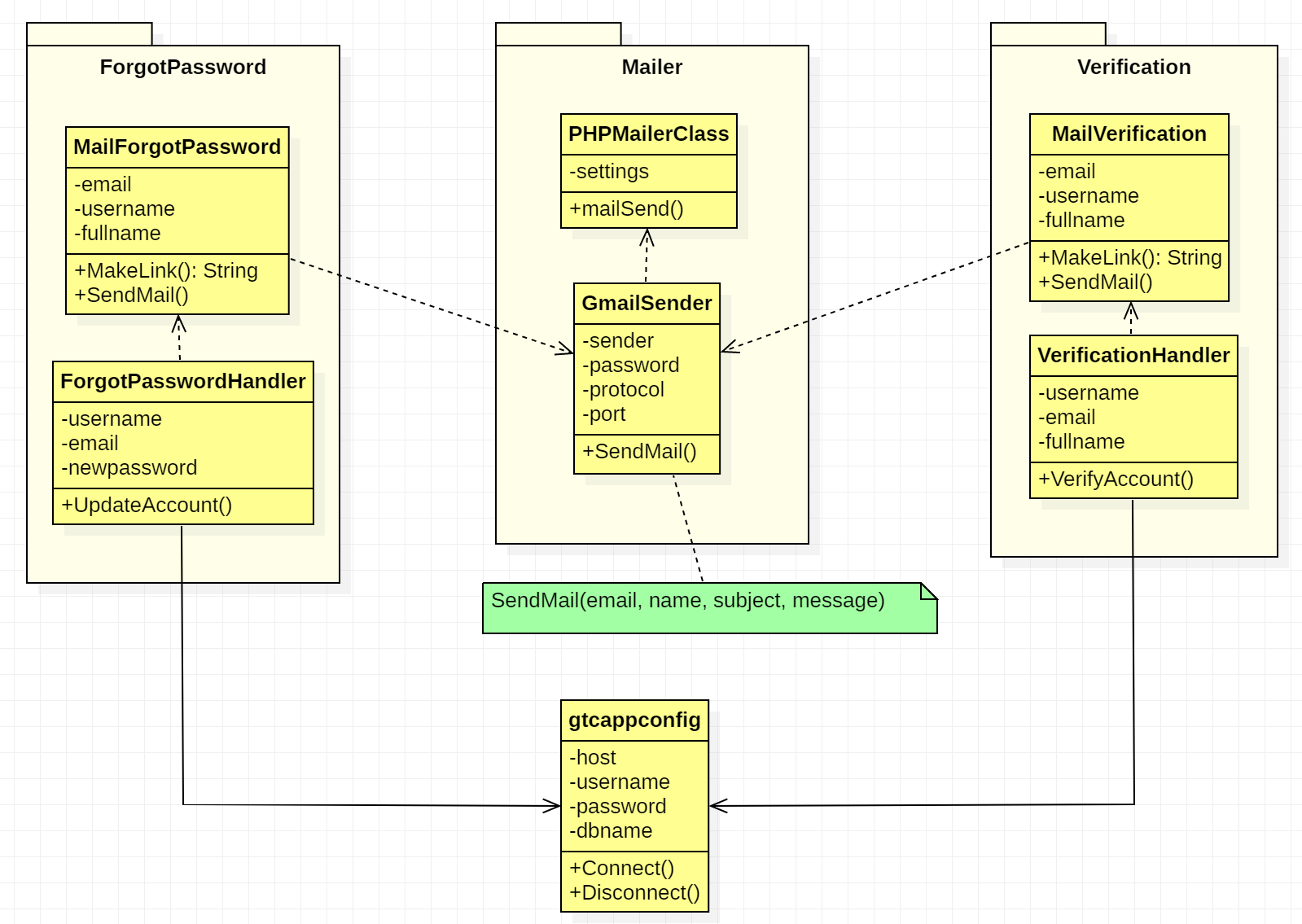
#### Servisi za slanje mailova

Na sledećem UML diagramu se mogu videti komponente koje omogućavaju slanje mailova korisnicima aplikacije.



Iz aplikacije se šalje zahtev komponentama **MailForgotPassword** i **MailVerification** u zavisnosti od potrebe, sa potrebnim podacima. Ove komponente kreiraju odgovarajući link i prosleđuju ga kao deo poruke komponenti **GmailSender**. GmailSender poseduje sva neophodna podešavanja za slanje maila. Dobijenu poruku šalje na uneti mail. Link koji je prethodno kreiran vodi do Handler komponente koja dalje vrši obradu i ažuriranje naloga. Za ažuriranje je neophodno uspostaviti konekciju sa bazom podataka pa se zato poziva komponenta gtcappconfig.

Realizacija ovih komponenti se detaljnije može videti na sledećem dijagramu klasa:

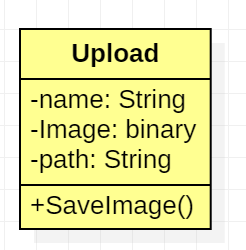


### Servisi za uploadovanje slika

Na sledećem UML diagramu su prikazane komponente servisa za upload-ovanje slika.



Komponenta Upload prihvata POST zahtev kroz koji je poslata slika u binarnom obliku i čuva u posebnom direktorijumu na Web sajtu. Dijagram klasa Upload komponente:



# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi:

1. Sistem će da podrži do 1000 simultanih pristupa API servisima.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 3 sekunde.
3. Učitavanje stranica ne sme trajati duže od 4 sekunde.

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

# Kvalitet

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

1. Usluge GTCApp aplikacije će biti dostupne 24/7.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.
3. Administrator mora u roku od 24 sata da odobri ili odbije zahteve.