SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Matija Benotić**

**Tomislav Dubravac**

**Dominik Vitez**

**Domagoj Andlar**

**Lukas Krištić**

Student Attendance System

PROJEKT IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA

TEHNIČKA DOKUMENTACIJA

Link na GitHub repozitorij: <https://github.com/matokc95/StudentAttendanceSystem-SAS>

Trello:

-Project Management board: <https://trello.com/b/4zK88RFV>

-Sprint 1 board: <https://trello.com/b/gd5jYoBF>

-Sprint 2 board: <https://trello.com/b/3m4MkmBW>

-Sprint 3 board: <https://trello.com/b/v1BkPQr1>

-Sprint 4 board: <https://trello.com/b/r1pNxY7l>

Varaždin, 2019.

Sadržaj

[Uvod 3](#_Toc536635954)

[Specifikacija zahtjeva 4](#_Toc536635955)

[Namjena dokumenta 4](#_Toc536635956)

[Ciljevi 4](#_Toc536635957)

[Svrha 4](#_Toc536635958)

[Djelokrug 4](#_Toc536635959)

[Perspektiva proizvoda 5](#_Toc536635960)

[Korisničke karakteristike 5](#_Toc536635961)

[Nefunkcionalni zahtjevi 5](#_Toc536635962)

[Performanse zahtjevi 5](#_Toc536635963)

[Sigurnosni zahtjevi 5](#_Toc536635964)

[Preventivne mjere zaštite 5](#_Toc536635965)

[Pravni zahtjevi 6](#_Toc536635966)

[Funkcionalni zahtjevi 6](#_Toc536635967)

[Profesor 6](#_Toc536635968)

[Student 7](#_Toc536635969)

[Tehnička Dokumentacija 8](#_Toc536635970)

[Arhitektura sustava 8](#_Toc536635971)

[Konceptualno i logičko oblikovanje 8](#_Toc536635972)

[Prijava 8](#_Toc536635973)

[Početni zaslon 9](#_Toc536635974)

[Zaslon prikaza kolegija 9](#_Toc536635975)

[Zaslon za CRUD kolegija 10](#_Toc536635976)

[Zaslon prikaza Seminara 11](#_Toc536635977)

[Zaslon za CRUD seminara 11](#_Toc536635978)

[Zaslon prikaza labosa 12](#_Toc536635979)

[Zaslon prikaza rasporeda predavanja 12](#_Toc536635980)

[Zaslon za praćenje evidencije 13](#_Toc536635981)

[Baza podataka 14](#_Toc536635982)

[Struktura aplikacije 17](#_Toc536635983)

[Modularizacija 17](#_Toc536635984)

[Modul app 18](#_Toc536635985)

[Modul core 18](#_Toc536635986)

[Modul webservice 18](#_Toc536635987)

[Modul passwordrecord 18](#_Toc536635988)

[Dijagram slučajeva aktivnosti 19](#_Toc536635989)

[Funkcionalnosti i pripadajući dijagrami 19](#_Toc536635990)

[Profesor 19](#_Toc536635991)

[Prijava 20](#_Toc536635992)

[Pretraživanje i dodavanje seminara 21](#_Toc536635993)

[Raspored predavanja 23](#_Toc536635994)

[Prisustvo pomoću lozinke 24](#_Toc536635995)

[Dijagrami modula 26](#_Toc536635996)

[Modul app 27](#_Toc536635997)

[Modul webservice 29](#_Toc536635998)

[Modul passwordrecord 30](#_Toc536635999)

[Web servis 30](#_Toc536636000)

[Popis slika 35](#_Toc536636001)

# 

# Uvod

Ovaj dokument prikazuje najvažnije informacije o projektu, opis funkcionalnosti aplikacije, korisnički zahtjevi, opis tehnologija i alata koji su potrebni za izradu sam aplikacije. Aplikacija „Student Attendance System“ je zamišljena da olakša evidenciju studenata na nekom predavanju pomoću pina ili prepoznavanja lica. Kako je evidencija studenata na predavanjima i dalje velika papirologija ova aplikacija bi ubrzala proces evidencije i automatizirala taj proces. Aplikacija nije zamišljena samo da bude od koristi profesoru već i studentima. Svaki student će moći pregledavati izostanke i raspored sati a tako i profesori.

Prilikom izrade mobilne aplikacije korištena je popularna metodologija razvoja „SCRUM“. Proces izrade projekta je podijeljen na sprintove gdje svaki sprint sadrži određen broj dana, a u ovom slučaju projekt je podijeljen na 4 sprint. Funkcionalnosti će se podijeliti po sprintovima kako bi svaki član tima imao svoj zadatak tokom jednog sprinta. Za evidenciju sprinteva koristi se web aplikacija Trello koja je besplatna i vrlo jednostavna. Koristit će se još Microsoft Visual Studio 2017 i programski jezik C# te paltforma za razvoj mobilnih i web aplikacija Firebase.

# Specifikacija zahtjeva

## Namjena dokumenta

Dokument je prvenstveno namijenjen razvojnom timu, a i osobama koje imaju želju da bolje upoznaju aplikaciju te da im se pobliže objasni korištenje iste i koja joj je svrha.

## Ciljevi

Glavna dva cilja ovog inovativnog proizvoda:

1. Većina fakulteta koristi printane potpisne liste kako bi se studenti na predavanjima potpisali i tako ostvarili dolazak koja su obavezna. Ponekad zbog velike predavaone/učione i velikog broja studenata to uzima dosta vremena te studenti i profesor moraju poslije nastave ostajati i čekati da se potpišu svi. Ova aplikacija bi spriječila to čekanje i evidencija prisutnih bi se riješila u 2-3 minute.
2. Potpisne liste su papirologija koja zahtjeva profesorovo dodatno vrijeme što je stvarno nepotrebno jer ova aplikacija uskratila taj posao.
3. Studenti će moći svoje izostanke svakodnevno provjeravati
4. Aplikacija će biti vrlo jednostavna za korištenje

## Svrha

Sličnih aplikacija već postoji u današnjem svijetu koje mogu prikazivati raspored sati, odabir grupa za laboratorijske vježbe ali nijedna ne može evidentirati prisustvo što ju baš to čini kompetentnijom i boljom od ostalih. Profesori više neće morati nositi printane popisne liste kako bi evidentirali prisustvo na predavanjima svaki tjedan. U nastavku ćemo opisati specifikacije aplikacije i ostale važne stvari.

## Djelokrug

Aplikacija Student Attendance System omogućavat će korisnicima da pregled upisanih kolegija, prikaz rasporeda sati za predavanja, seminare i laboratorijske vježbe, prikaz dolazaka odnosno nedolazaka. Korisnik(profesor) će biti u mogućnosti dodavati nove kolegije i odabir termina labosa (student u ovom slučaju). Korisnik(profesor) će moći evidentirati prisutnost tako da mu aplikacija izgenerira pin kojeg studenti unose u aplikaciju(svaki student ima vlastiti profil) i tako evidentiraju dolazak. Drugi način evidencije je pomoću algoritma za prepoznavanje lica tako da profesor slika auditorij i aplikacija sama detektira prisutne.

## Perspektiva proizvoda

Na domaćem tržištu postoji sličnih aplikacija koje također imaju prikaz rasporeda predavanja i unošenja izostanaka ručno ali većina studenata je zaboravna pa to baš na kraju i ne ispadne uvijek najtočnije. To je glavni razlog za razvoj ove naše aplikacije da automatiziramo postupak evidencije i uvedemo novitete prepoznavanja lica u obrazovnu ustanovu što nije tako rasprostranjeno. Student Attendance aplikacija je bazirana na aplikaciji Fakulteta informatike i organizacije koja također može prikazati raspored studenta ali sve student mora sam ručno unositi pa tako i izostanke te studenti ju rijetko koriste. Naša aplikacija bi više koristila studentima.

## Korisničke karakteristike

Minimalne karakteristike korisnika za aplikaciju „Student Attendance System bi bile:

1. Osnovno rukovanje pametnim mobtitelom
2. Upoznat sa radom aplikacije Google Play Store

Preporučene karakteristike:

1. Napredno korištenje pametnih mobitela

## Nefunkcionalni zahtjevi

### Performanse zahtjevi

Pokretanje aplikacije ne bi trebalo prelaziti vrijeme veće od 4 sekunde. Za 20 korisnika vrijeme čekanja aplikacije ne bi trebalo prelaziti 5 sekundi a 99% operacija se treba izvršiti u roku od 2 sekunde.

### Sigurnosni zahtjevi

Korisnici se dijele na profesore i studente i studenti nisu u mogućnosti iz generirati pin kako bi si evidentirali dolazak a tako niti pokretanje kamere za slikanje auditorija. Student ima manje ovlasti od profesora. Registraciju izvršava 3. osoba tako da se korisnici ne mogu sami registrirati. Podaci u bazi podataka(lozinke) su zaštićene od sql injection napada.

### Preventivne mjere zaštite

Profesor bi morao pripaziti kako svoj mobitel ne bi slučajno ostavio u dvorani kako studenti ne bi zloupotrijebili aplikaciju u svoje svrhe dok profesora nema.

Preventivne mjere: Postaviti lozinku na mobitelu prilikom svakog zaključavanja, postaviti otisak prsta kao metodu otključavanja mobitela, uredno odjavljivanje sa aplikacije.

### Pravni zahtjevi

Svi podaci u aplikaciji pripadaju obrazovnoj ustanovi koja kupi aplikaciju i nitko drugi ne može doći do tih podataka niti zloupotrebljavati ih.

## Funkcionalni zahtjevi

Funkcionalni zahtjevi korisnika su podijeljene u dvije kategorije. Korisnici aplikacije se dijele na profesore i studente. Navest ćemo za svakog korisnika posebno sve funkcionalnosti koje može obavljati.

### Profesor

#### Prijava

Za korištenje aplikacije profesor se prvo mora prijaviti u aplikaciju. Podatke za prijavu bi trebao dobiti od obrazovne ustanove (slično kao na Fakulteti organizacije i informatike eduhr podaci). Korisnik odabire kategoriju profesor za prijavu.

#### Prikaz rasporeda predavanja

Profesoru će biti automatski iz generiran raspored predavanja i tako će moći pratiti na aplikaciji kada, u kojoj dvorani i u koliko sati ima nastavu.

#### CRUD kolegija/seminara/laboratorijskih vježbi

Profesor je u mogućnosti kreirati novi kolegij te za novo kreirani kolegij može kreirati nove termine seminara i laboratorijskih vježbi. Isto to može i raditi za već postojeće kolegije. Prilikom kreiranja upisuje naziv, vrijeme početka i završetka, dvoranu, broj dozvoljenih izostanaka.

#### Dodavanje studenata na kolegije

Profesor ima ovlasti da studenta posebno doda na kolegij ako kojim slučajem nije zapisan u bazi.

#### Evidencija dolazaka studenata po kolegiju

Profesor u svakom trenu ima mogućnost pogledati dolaske odnosno nedolaske određenog studenta na kolegiju (predavanje, seminar, labosi).

#### Generiranje pina/lozinke za prisustvo

Profesor ima opciju za generiranje pina/lozinke za provjeru prisustva kako bi studenti mogli unijeti u svoje aplikacije i tako zabilježili dolazak na predavanja. Pin/lozinka je samo za kratko vrijeme aktivna.

#### Autentifikacija prepoznavanjem lica

Profesor može evidenciju izvršiti tako da poslika auditorij gdje algoritam za prepoznavanje detektira i autentificira studente kako bi ima zabilježilo dolazak

#### Odjava

Profesor se može u svakom trenu odjaviti iz apliakcije.

### Student

#### Prijava

Student na početnom zaslonu odabire „student“ i prijavljuje se u sustav tako što unese šifru i mail.

#### Prikaz rasporeda sati

Studentu će biti automatski iz generiran raspored predavanja(svi kolegiji koje je upisao) i tako će moći pratiti na aplikaciji kada, u kojoj dvorani i u koliko sati ima nastavu.

#### Pregled upisanih kolegija (laboratorijskih vježbi, seminara, predavanja)

Student ima mogućnost pregleda upisanih kolegija koji su prikazani kao lista, a tako i za laboratorijske vježbe , seminare i predavanja. Svaka kategorija je na posebnoj aktivnosti prikazana.

#### Predbilježba za laboratorijske vježbe

Svaki student ima mogućnost odabira laboratorijskih vježbi kada ih profesor definira. Student je u mogućnosti poništiti odabir i odabrati drugi termin ali ne može više od jednog termina odabrati

#### Evidencija dolazaka

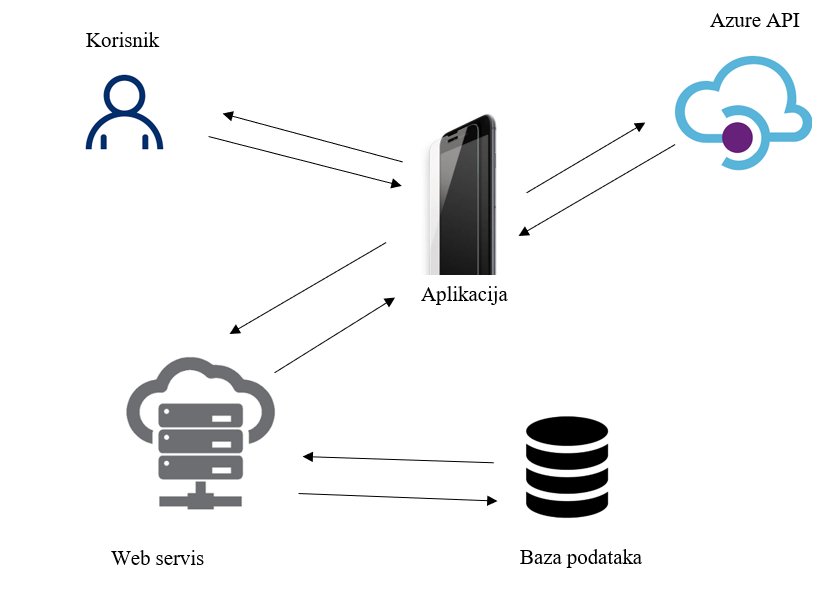
Student ima mogućnost provjere svoje prisutnosti na kolegiju tako što odabere kolegij zatim tip aktivnosti na kolegiju(predavanje, labosi, seminari) i ispiše njegovu prisutnost za svaki tjedan.

#### Registracija

Registraciju bi trebala izvršavati neku u obrazovnoj ustanovi kao što se na upisu na fakultetu radi. Studenti se upišu i zatim ih neko u sustav unosi i nakon toga dobivaju email i lozinku.

# Tehnička Dokumentacija

## Arhitektura sustava



Slika 1. Arhitektura aplikacije

## Konceptualno i logičko oblikovanje

### Prijava

Nakon što se aplikacija učita, prikazuje se početni zaslon koji prikazuje opciju prijavljivanja. Potrebno je unijeti korisničko ime i lozinku. Valjanom prijavom otvara se zaslon početne stranice.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Početni zaslon

Na ovom zaslonu moguće je vidjeti sve mogućnosti koje profesor može ostvariti putem aplikacije. Moguće je pregledati kolegije, seminare i labose za pojedinog profesora te također raspored predavanja i vođenje prisustva na nastavi.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Zaslon prikaza kolegija

Klikom na početnom zaslonu na kategoriju „Moji kolegiji“ profesor može vidjeti popis svojih kolegija za koje je pribilježen. Također u mogućnosti je obrisati pojedini kolegij ili ga urediti što će detaljnije biti prikazano na sljedećem zaslonu.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Zaslon za CRUD kolegija

Kao što smo naveli u prethodnom primjeru, ovaj zaslon nam prikazuje mogućnost odabira izrade, uređivanja i brisanja pojedinog kolegija.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Zaslon prikaza Seminara

Klikom na početnom zaslonu na kategoriju „Moji seminari“ profesor može vidjeti popis svojih seminara za koje je pribilježen. Također u mogućnosti je dodati novu seminarsku grupu, odnosno termin ili ga urediti što će detaljnije biti prikazano na sljedećem zaslonu.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Zaslon za CRUD seminara

Klikom na prethodnom zaslonu na opciju „Dodaj“ otvara nam se trenutni zaslon te se nalaze mogućnosti dodavanja seminara. Traže se informacije poput naziva seminara, dvorane, vrijeme održavanja i dan u tjednu uz opciju pritiska gumba „Spremi promjene“.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Zaslon prikaza labosa

Klikom na početnom zaslonu na kategoriju „Moji labosi“ profesor može vidjeti popis svojih labosa za koje je pribilježen. Također u mogućnosti je dodati novi laboratorijski termin ili ga urediti na isti način kao što smo prethodno naveli za seminare i kolegije.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

### Zaslon prikaza rasporeda predavanja

Klikom na opciju „Raspored predavanja“ profesor po određenom datumu može provjeriti svoj raspored predavanja.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

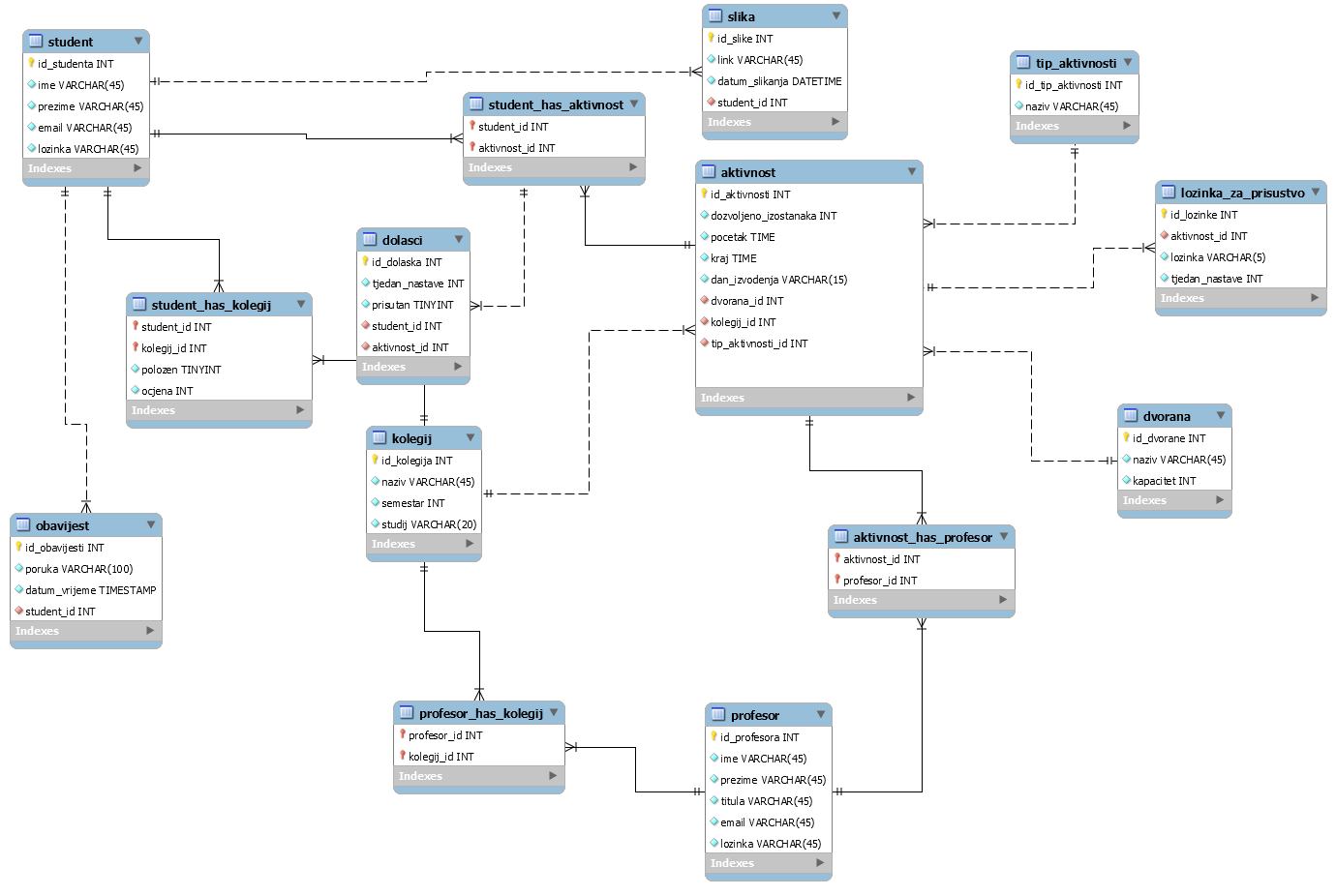
### Zaslon za praćenje evidencije

Odabirom opcije „Prisustva“, profesoru se izlistava popis studenata ispod kojih se može vidjeti da li je student bio prisutan na predavanju/seminaru/labosu.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

# Baza podataka



Slika 2. ERA model

Za potrebe projekta izrađen je era model od ukupno 13 tablica, 9 glavnih i 4 pomoćne. Glavne tablice su : student, obavijest, dolasci, kolegij, aktivnost, slika, profesor, dvorana i lozinka\_za\_prisustvo.

U tablicu student upisuju e podaci o studentu, a tablica sadrži sljedeće atribute :

id\_stuenta(INT) – primarni ključ tablice

ime (VARCHAR) – ime studenta

prezime (VARCHAR) – prezime studenta

email (VARCHAR) – službeni e-mail studenta

lozinka (VARCHAR) – lozinka studenta za prijavu u sustav fakulteta

Tablica student povezana je neobvezujućom vezom 1 naprema više sa tablicama slika i obavijest te je pomoću pomoćnih tablica vezom više naprema više povezana tablicama aktivnost i kolegij.

Tablica slika sadrži podatke o slikama studenta koje se koriste kako bi se student pomoću prepoznavanja lica mogao zabilježiti za prisustvo na nastavi. Tablica sadrži atribute :

id\_slike (INT) – primarni ključ tablice

link (VARCHAR) – putanja do slike

datum\_slikanja (DATETIME) – datum kada je slika napravljena

student\_id (INT) – vanjski ključ na tablicu student

Tablica obavijest sadrži podatke o obavijesti koje student primi, kao npr. „Imate pravo na još jedan izostanak“. Tablica sadrži sljedeće atribute:

id\_obavijesti (INT) – primarni ključ tablice

poruka (VARCHAR) – tekst obavijesti koju student primi

datum\_vrijeme (TIMESTAMP) – datum i vrijeme kada je obavijest poslana

student\_id (INT) – vanjski ključa na tablicu student

Tablica kolegij sadrži informacije o kolegijima koji se predaju na fakultetu. Neobvezujućom vezom 1 naprama više povezana je s tablicom aktivnost te je pomoću pomoćnih tablica vezom više prema više povezana sa tablicama student i profesor. Tablica kolegij sadrži atribute:

id\_kolegija (INT) – primarni ključ tablice

naziv (VARCHAR) – naziv kolegija

semestar (INT) – semestar u kojemu se kolegij izvodi

studij (VARCHAR) – naziv studija na kojemu se izvodi kolegij

pomoćna tablica student\_has\_kolegij osim vanjskih ključeva na tablice student i kolegij sadrži atribute:

polozen (TINYINT) – označava je li student položio kolegij

ocjena (INT) – ocjena s kojom je kolegij položen (ako je položen)

Tablica aktivnost sadrži podatke o aktivnostima kolegija (predavanja, seminare, labaratorijske vježbe, …). Tablica je povezana neobvezujućim vezama više naprema jedan s tablicama kolegij i dvorana dok je neobvezujućom vezom jedan naprema više povezana s tablicom lozinka\_za\_prisutstvo. Tablica je povezana vezom više prema više s tablicama profesor i student. Tablica aktivnost posjeduje atribute:

id\_aktivnosti (INT) – primarni ključ tablice

tip\_aktivnosti (VARCHAR)- označuje je li predavanje, seminar ili neka druga aktivnost vezana uz kolegij

dozvoljeno\_izostanaka (INT) – ukupan broj dozvoljenih izostanaka s aktivnosti

pocetak (TIME) – vrijeme početka aktivnosti

kraj (TIME) – vrijeme završetka aktivnosti

dan\_izvodenja (VARCHAR) – dan u tjednu kada se aktivnost izvodi

dvorana\_id (INT) – vanjski ključ na dvoranu

kolegij\_id (INT) – vanjski ključ na kolegij

Tablica lozinka\_za\_prisustvo pohranjuje podatke o lozinkama koje studenti unose kako bi zabilježili prisustvo na nekoj aktivnosti vezanoj za kolegij. Tablica sadrži atribute:

id\_lozinke (INT) – primarni ključ tablice

aktivnost\_id (INT) – vanjski ključ na tablicu aktivnost

lozinka (VARCHAR)- lozinka

tjedan\_nastave(INT)- trenutni tjedan izvođenja nastave

U tablici dvorana pohranjeni su podaci o dvoranama fakulteta. Tablica sadrži atribute :

id\_dvorane (INT) – primarni ključ tablice

naziv (VARCHAR) – naziv dvorane

kapacitet (INT) – broj sjedećih mjesta u dvorani

Tablica profesor sadrži podatke o profesorima i asistentima na kolegiju. Vezama više naprema više povezana je s tablicama aktivnost i kolegij. Tablica sadrži atribute:

id\_profesora (INT) – primarni ključ tablice

ime (VARCHAR) – ime profesora

prezime (VARCHAR) – prezime profesora

titula (VARCHAR) – titula profesora

email (VARCHAR) – službeni email profesora

lozinka (VARCHAR) – lozinka za prijavu u sustav fakulteta

# Struktura aplikacije

Struktura se sastoji od tri modula (app, core, webservice i passwordrecord). Unutar app modula nalaze se paketi i klase potrebni za sam prikaz podataka, obradu podataka, listeneri za web servis i nekoliko paketa. Unutar core modula nalaze se klase koje su potrebne app i core modulu. Nalaze se upravo tu jer se većina tih klasa koristi kod parsiranja podataka sa web servisa, a slijedno tome koriste se i za kreiranje, ažuriranje ili slanje podataka unutar app modula. Modul web service služi za dohvaćanje i slanje podataka na php web servis. Modul passwordrecord služi za evidenciju prisustva pomoću lozinke koju generira profesor.

# Modularizacija

Modularno programiranje je tehnika projektiranja softvera koja naglašava odvajanje funkcionalnosti programa u neovisne, zamjenjivie module, tako da svaki sadrži sve što je potrebno za izvršavanje samo jednog aspekta željene funkcionalnosti. Naš projekt se sastoji od pet modula (app, core, webservice, passwordrecord i facerecognition). Unutar app modula nalaze se paketi i klase potrebni za sam prikaz podataka, obradu podataka, listeneri za web servis i nekoliko paketa. Unutar core modula nalaze se klase koje su potrebne app i core modulu. Nalaze se upravo tu jer se većina tih klasa koristi kod parsiranja podataka sa web servisa, a slijedno tome koriste se i za kreiranje, ažuriranje ili slanje podataka unutar app modula. Modul web service služi za dohvaćanje i slanje podataka na php web servis. Modul passwordrecord služi za evidenciju prisustva pomoću lozinke koju generira profesor. Modul facerecognition služi za evidentiranje prisustva studenata pomoću prepoznavanja lica.

Modulsko sučelje izražava elemente koji su navedeni i potrebni za modul. Elementi definirani u sučelju mogu se otkriti drugim modulima. Implementacija sadrži radni kod koji odgovara elementima deklariranim u sučelju. U našem slučaju, sučelje sadrži metodu za prosljeđivanje podataka u modul i metodu za prosljeđivanje podataka iz modula.

Modularno programiranje usko je povezano sa strukturiranim programiranjem i objektno orijentiranim programiranjem, a svi imaju isti cilj olakšati izgradnju velikih softverskih programa i sustava dekompozicijom na manje komade.

## Modul app

Sastoji se od paketa poput „adaptersprofesor“, „adaptersstudent“, „uiprofesor“, „uistudent“ te klasa koje predstavaljaju aktivnosti: „MainActivity“, „ListOfSeminars“, „ListOfLabs“, „Login“, „AddSeminar“ i „AddLab“...

**Napomena**! Registracija korisnika se provodi zasebnom desktop aplikacijom. Zamišljeno je da se ova aktivnost izvršava unutar fakulteta.

## Modul core

Služi kao posrednik za klase koje trebaju modulima app, webservice i passwordercord za komunikaciju. To su u pravilu entitetne klase.

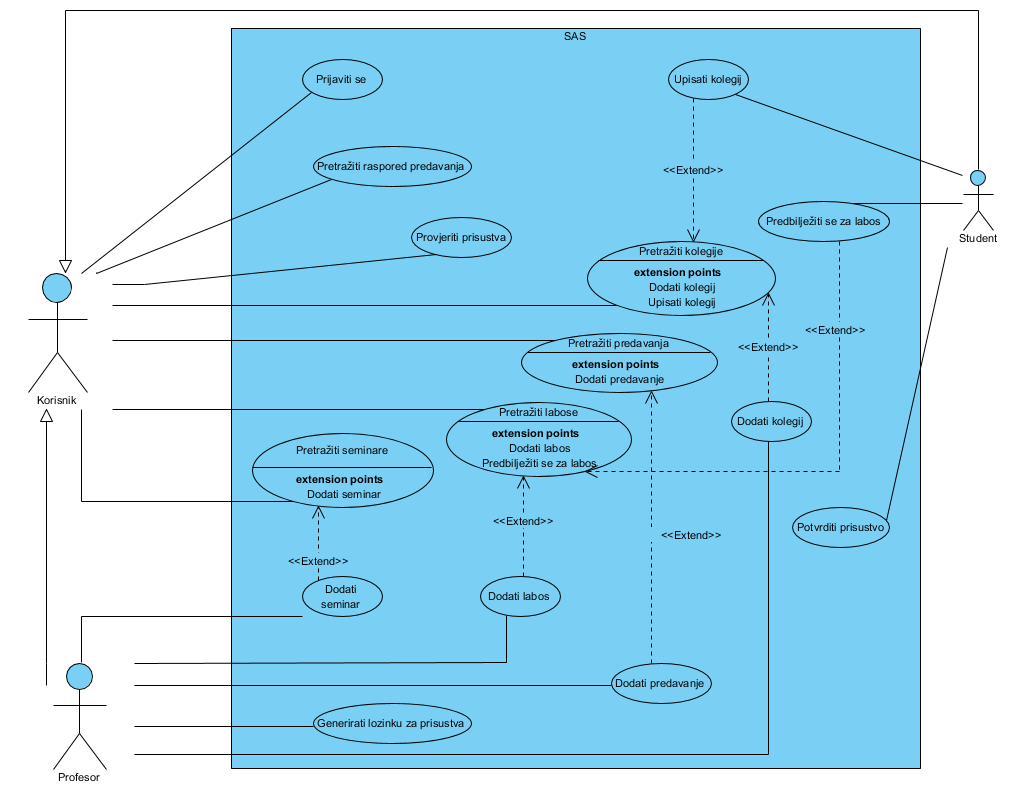
## Modul webservice

U sklopu ovog modula imamo paket „responses“ unuta kojeg se nalazi klasa „SasWebServiceResponse“ (čini strukturu odgovora koji primamo od web servisa). Uz paket tu se nalaze klasa „SasWebService“ te sučelja „SasWebServiceCaller“ i „SasWebServiceHandler“. Interface „SasWebService“ sadrži retrofit anotacije te putanje sa parametrima za kreiranje HTTP zahtjeva.“SasWebServiceHandler“ je interface sa metodom za prosljeđivanje dobivenog rezultata od web servisa prema aplikaciji, tj. koristi se za proslijeđivanje poruke odgovora. Klasa „SasWebServiceCaller“ sastoji se od metoda za prosljeđivanje parametara iz aplikacije prema HTTP zahtjevu koji se generira, ali i za handlanje odgovora koji nam web servis vraća i njihovo prosljeđivanje listeneru, tj konkretno sučelju SasWebServiceHandlerer.

## Modul passwordrecord

Modul „passwordrecord“ sadrži pakete „loaders“, „responses“, „profesor“, „student“ i klase „SasWebService“, SasWebServiceCaller“ i „SasWebServiceHandler“. Paket „profesor“ sadrži klasu „GeneratePassword“ kojom profesor generira lozinku na temelju podataka koji se dobe iz „app“ modula preko sučelja.. Paket „student“ sadrži klasu „SubmitAttendance“ koja vraća „app“ modulu studenta koji je prisutan preko sučelja. Preostale klase služe za komunikaciju sa Web servisom.

# Dijagram slučajeva aktivnosti



Slika 3. Dijagram slučajeva korištenja

Dijagram slučajeva korištenja se sastoji od 2 učesnika (profesor i student), međutim neke funkcionalnosti su identične za oba učesnika pa se uvodi pojam generalizacije, odnosno učesnik „korisnik“. Korisnik se prijavljuje u aplikaciju, pretražuje kolegije, pretražuje predavanja, labose i seminare te može pretražiti raspored predavanja i uvid u prisustva na nastavi. Pojedinačno, profesor može dodavati seminare, labose, predavanja i kolegije te generirati lozinku za pojedinu aktivnost. Pojedinačno, student može upisati željeni kolegij, predbilježiti se za labose te potvrditi prisustvo na određenoj aktivnosti.

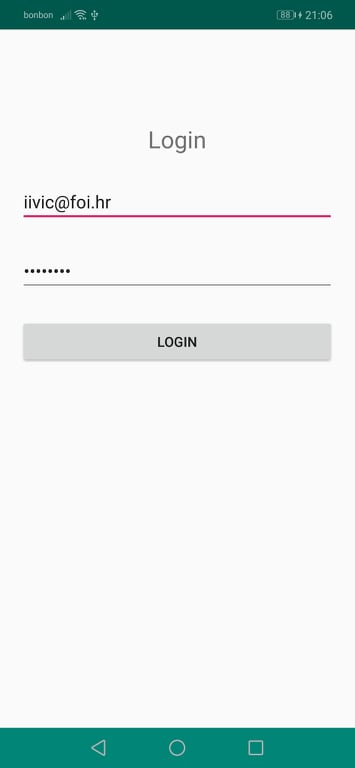
# Funkcionalnosti i pripadajući dijagrami

Funkcionalnosti aplikacije su podjeljene na dvije uloge: profesor i student, te će tako biti razmatrane i odvojene u nastavku.

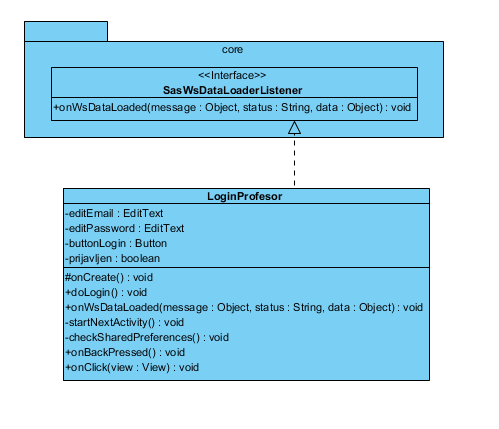
## Profesor

Profesor ima na raspolaganju pretraživanje i dodavanje kolegija, pretraživanje i dodavanje predavanja, pretraživanje i dodavanje seminara, pretraživanje i dodavanje labosa, pregled rasporeda predavanja, mogućnost odabira načina pomoću kojeg će se studenti predbilježiti za prisustvo na nastavi. U nastavku će te funkcionalnosti biti opisane i prikazane pripadajućim dijagramima.

### Prijava



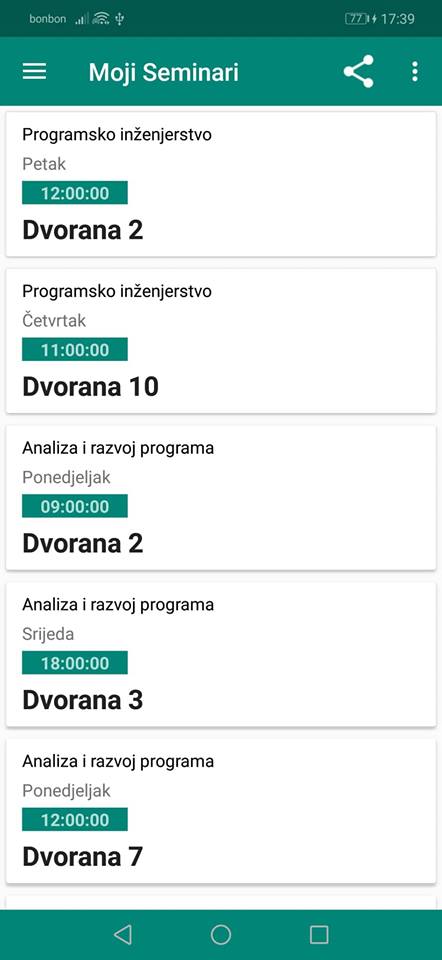
#### Dijagram klasa



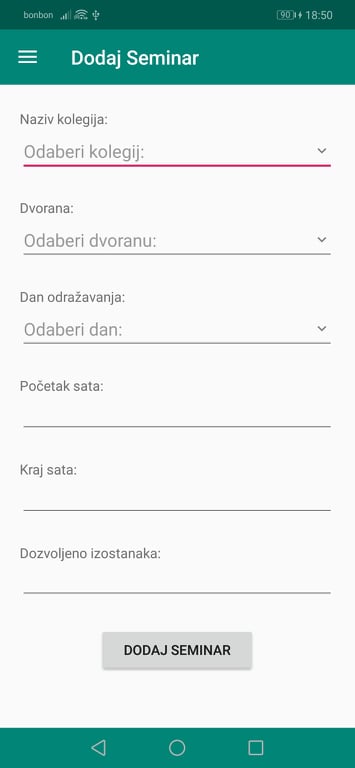
Slika 4. Dijagram klasa prijave

Funkcionalnost se sastoji od jedne aktivnosti i jednog sučelja. Core je modul gdje se nalaze entitetne klase i sučelja. „LoginProfesor“ je klasa za autentifikaciju korisnika kod prijave i ta klasa realizira sučelje „SasWsDataLoaderListener“.

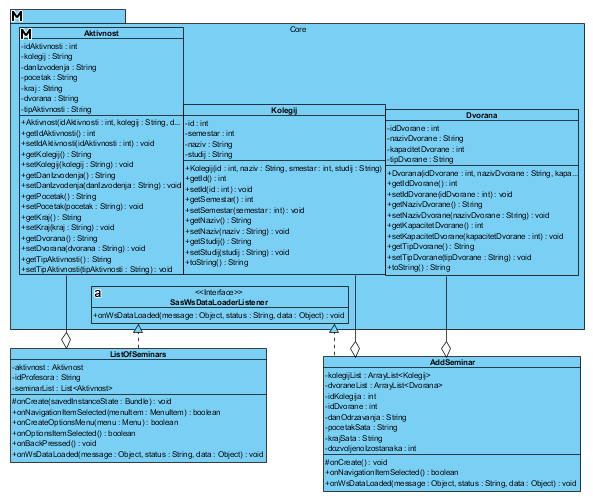
### Pretraživanje i dodavanje seminara



Odabirom funkcionalnosti, profesoru se prikažu svi seminari koji su mu dodjeljeni. Odabirom „tri točkice“ na izborniku, profesor može dodati novi seminar. Dijagram klasa je opisan u sljedećem poglavlju.



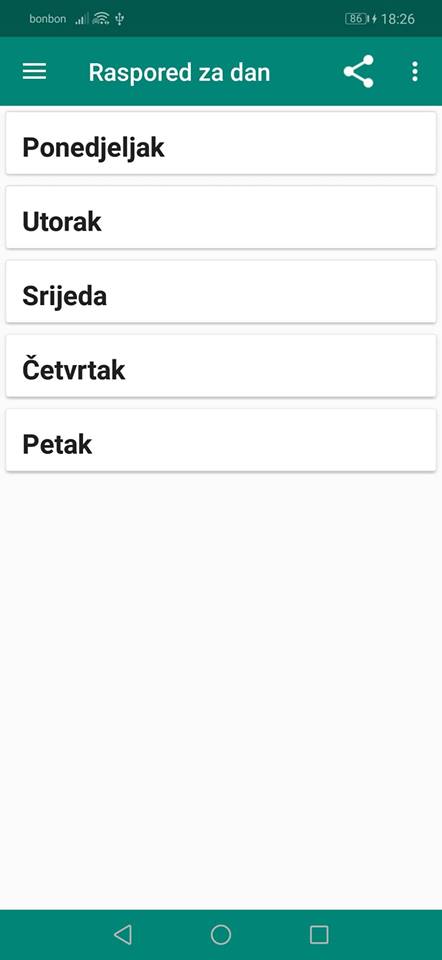
#### Dijagram klasa



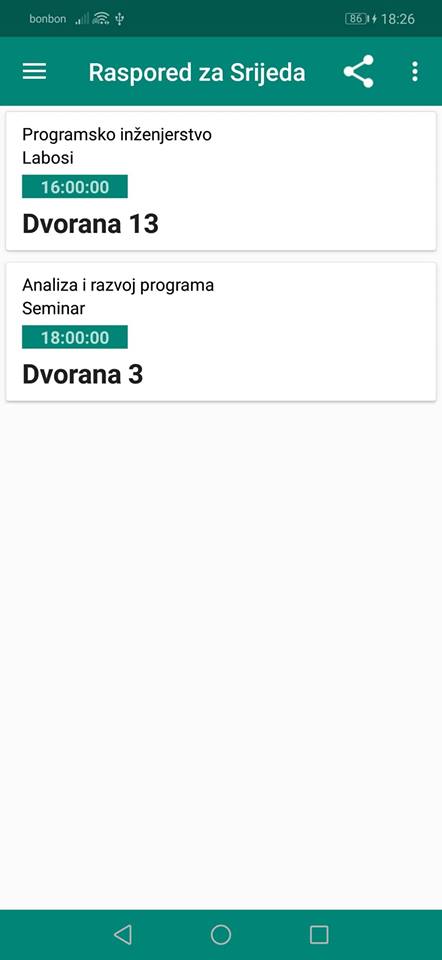
Slika 5. Dijagram klasa za seminare

Funkcionalnost se sastoji od dvije aktivnosti, tri entitetne klase i jednog sučelja. Core je modul u kojem se nalaze entitetne klase i sučelja. „ListOfSeminars“ je klasa za prikaz svih seminara od profesora i ona agregira entitetnu klasu „Aktivnost“. „AddSeminar“ je klasa za dodavanje novog seminara profesora i ona agregira entitetnu klasu „Kolegij“ i „Dvorana“. Kroz sučelja se dobivaju podaci sa WebServis-a.

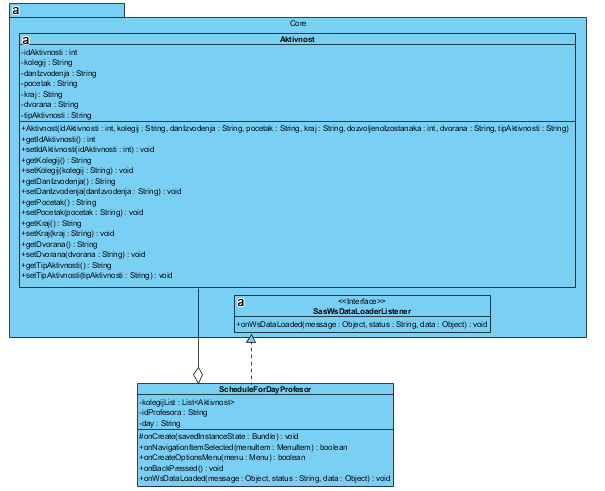
### Raspored predavanja



Odabirom ove funkcionalnosti, profesor ima uvid na raspored na temelju odabranog dana. Klikom na pojedini dan, otvara se nova aktivnost za prikaz rasporeda za odabrani dan.



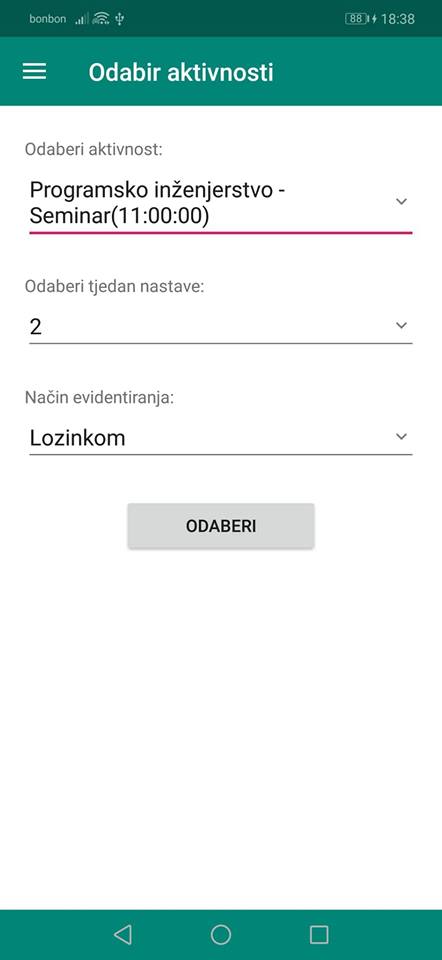
#### Dijagram klasa

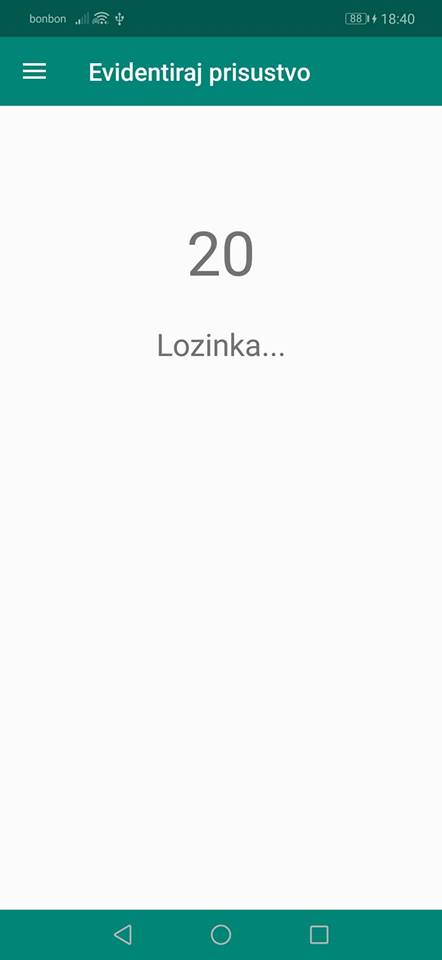


Slika 6. Dijagram klasa za raspored predavanja

Funkcionalnost se sastoji od jedne aktivnosti, jedne entitetne klase i jednog sučelja. „ScheduleForDayProfesor“ je klasa za prikaz dnevnog rasporeda za odabrani dan i ona agregira entitetnu klasu „Aktivnost“ iz Core modula. Sučelje „SasWsDataLoadedListener“ nam pruža metodu onWsDataLoaded() koja se implementira u aktivnosti i pomoću nje se dohvaćaju podati sa WebServis-a.

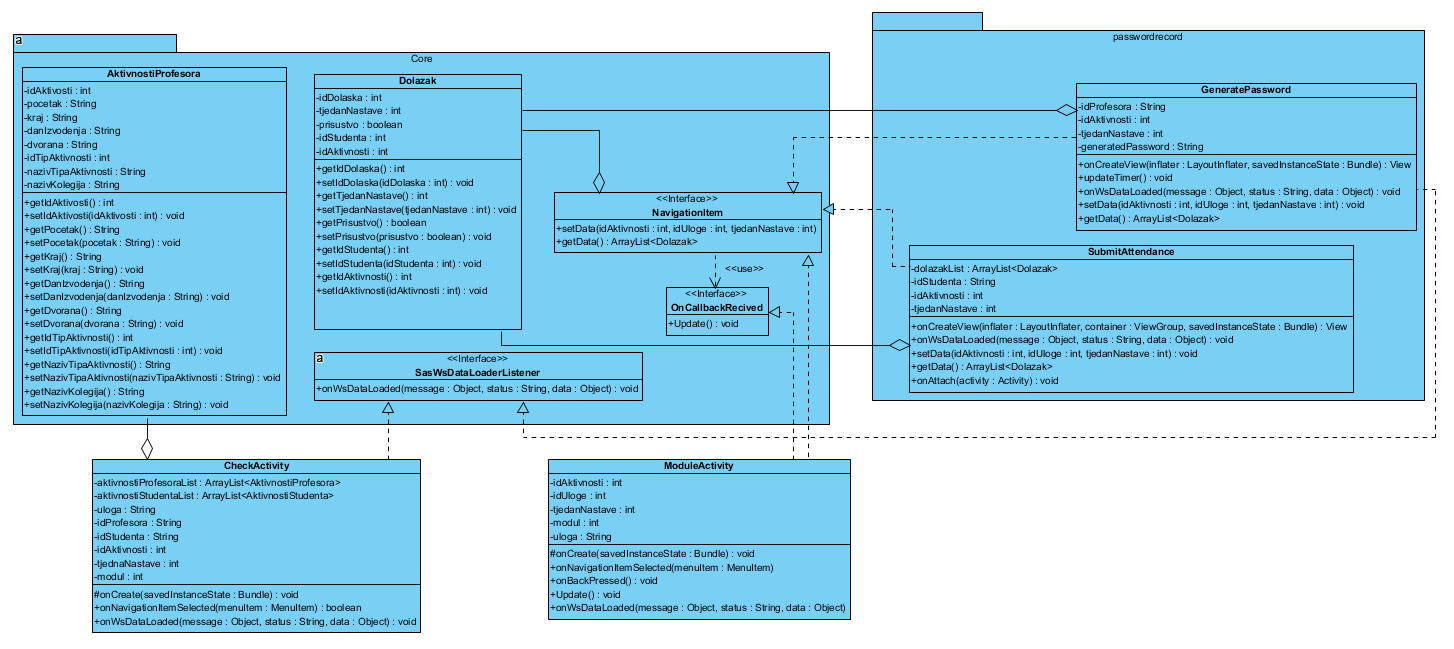
### Prisustvo pomoću lozinke





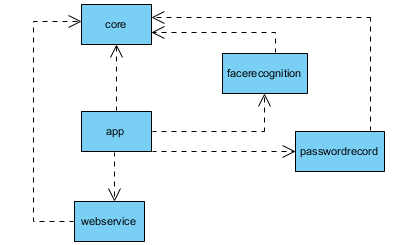
Odabirom ove funkcionalnosti, profesor može generirati lozinku za određenu aktivnost i određeni tjedan nastave. Funkcionalnost je realizirana modularno, odn. podaci o aktivnosti se kroz sučelje šalju iz „app“ modula u „passwordrecord“ modul i u „passwordrecord“ modulu se generira lozinka koja se na posljetku prikazuje korisniku. Valja napomenuti da generirana lozinka za odabranu aktivnost vrijedi samo 20 sec, odn. 20 sec studenti imaju vremena za unos iste kako bi potvrdili svoje prisustvo na toj aktivnosti.

#### Dijagram klasa



Slika 7.Dijagram klasa za generiranje lozinki

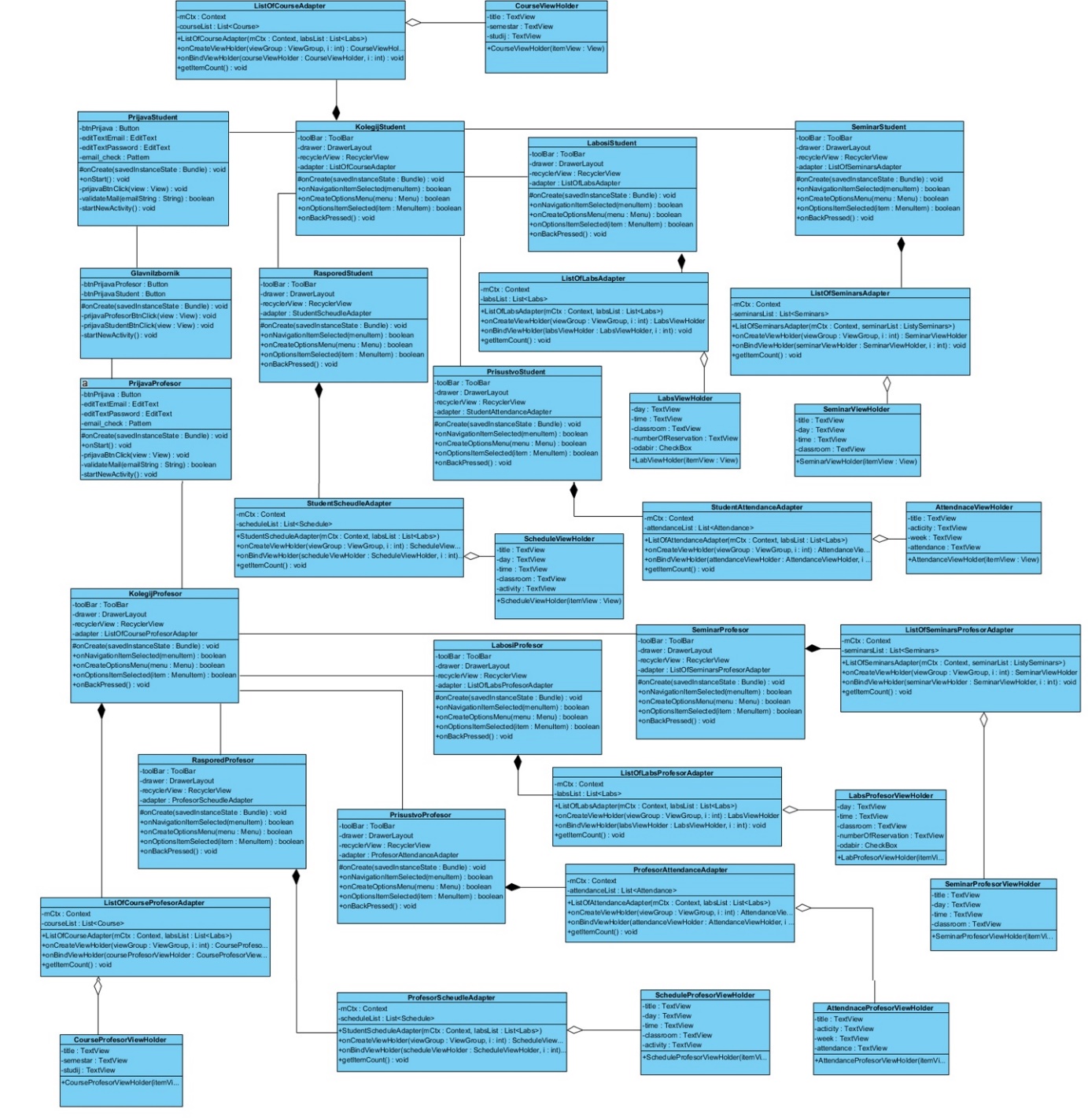
# Dijagrami modula



Slika 8. Dijagram modula

U aplikaciji se koriste tri modula – app, core, webservice. Početna aktivnost nalazi se u „*app*“ modulu, zajedno sa svim ostalim aktivnostima i adapterima za listu. Modul „*core*“ sadrži klase koje opisuju entitete poput profesora, studenta, kolegije, seminare, te podataka za padajuće izbornike, te zato za ovaj modul nismo radili dijagram klasa. Modul „*webservice*“ brine se o slanju poslanih podataka iz aplikacije, te slanju dobivenih podataka iz web servisa natrag u aplikaciju. Dijagram klasa za modul „*passwordrecord*“ će biti dodan kasnije dok se razradi plan izrade.

## Modul app



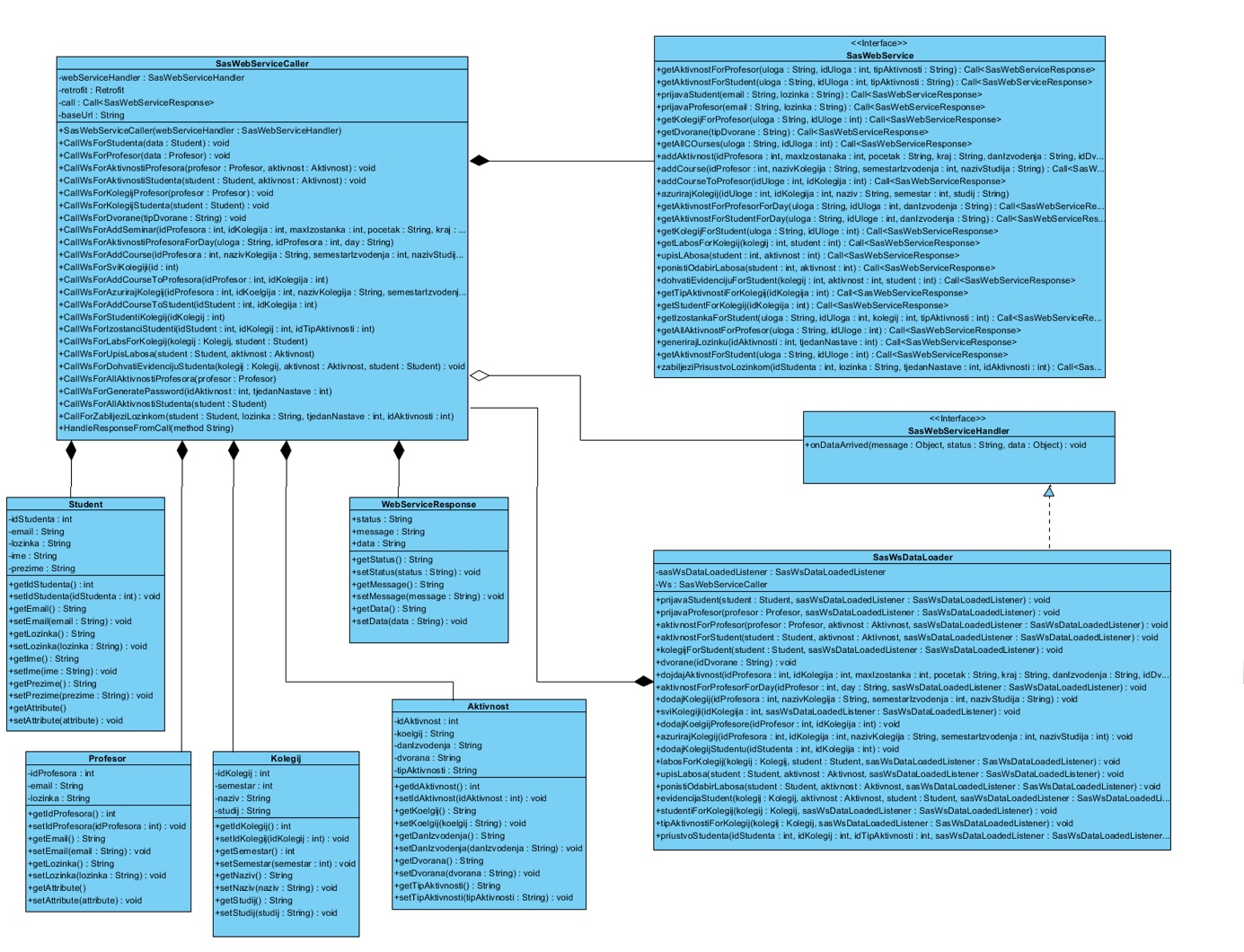
Slika 9. Dijagram klasa app modula

Na slici 2. nalazi se dijagram klasa za app modul, a opis klasa nalazi u nastavku:

* GlavniIzbornik – aktivnost na kojoj se odabire uloga korisnika (student/profesor)
* PrijavaStudent – aktivnost za prijavu studenta u sustav. Ukoliko je prijava uspješna, podaci se upisuju u shared preferences
* PrijavaProfesor – identična aktivnost kao i prethodna uz ulogu profesora
* KolegijiStudent – aktivnost sa popisom kolegija prijavljenog studenta
* KolegijProfesor - aktivnost sa popisom kolegija prijavljenog profesora
* SeminariStudent – aktivnost sa popisom seminara prijavljenog studenta
* SeminariProfesor - aktivnost sa popisom seminara prijavljenog profesora
* LabosiStudent – aktivnost sa popisom labosa prijavljenog studenta
* LabosiProfesor – aktivnost sa popisom labosa prijavljenog profesora
* ListOfSeminarAdapter – adapter klasa za punjenje liste podacima o seminarima
* SeminarViewHolder – klasa unutar adapter klase ListOfSeminarsAdapter
* ListOfLabsAdapter – adapter klasa za punjenje liste podacima o labosima
* LabViewHolder – klasa unutar adapter klase ListOfLabsAdapter
* RasporedStudent – aktivnost sa rasporedom sati za predavanja, labose i seminare za studenta
* ListOfCourseAdapter – adapter klasa za punjenje liste podacima o predavanjima
* CourseViewHolder – klasa unutar adapter klase ListOfCourseAdapter
* PrisustvoStudent – aktivnost sa evidencijom prsisustva studenta na pojedinom kolegiju za labose, predavanja i seminare (prikaz po tjednu nastave)

**Napomena**! Za profesora su klase identične kao i za studenta, pa ih zato nismo dodavali da dijagram bude pregledniji.

## Modul webservice

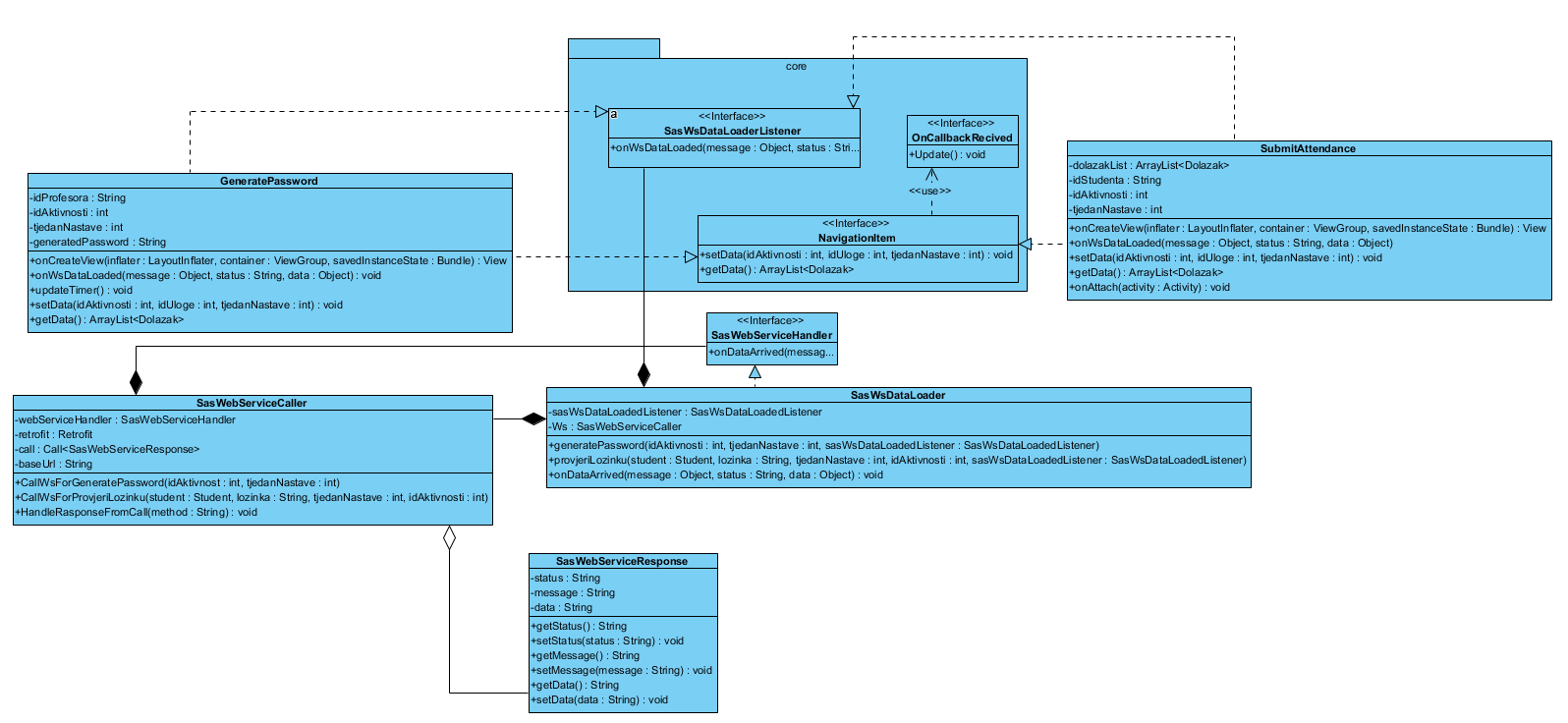


Slika 10. Dijagram klasa webservice modula

Dijagram klasa za modul webservice prikazan je na slici 3, a detalji pojedine klase su navedeni u nastavku:

* WebServiceResponse: POJO klasa koja definira izgled odgovora od strane web servisa
* WebServiceHandler: sučelje koje javlja svim klasama koje je implementiraju da je web servis vratio odgovor. Ovu klasu implementira jedino WsDataLoader, koji je ujedno i zadužen za komunikaciju app modula i webservice modula
* WebService: sučelje koje koristi retrofit. U njemu su definirani svi API pozivi prema web servisu.
* WebServiceCaller: klasa koja šalje podatke dobivene od WsDataLoadera, te dobiva podatke direktno od web servisa (te ih po potrebi filtrira i šalje natrag WsDataLoader-u)

## Modul passwordrecord



Slika 11. Dijagram klasa passwordrecord modula

# Web servis

Kako aplikaciju istovremeno može koristiti više korisnika koji trebaju podatke iz baze napravljen je PHP RESTfull web servis te se za dohvaćanje ili manipulaciju podataka treba organizirati određena struktura URL-a. GitHub link webservisa je <https://github.com/matokc95/SAS-WebService>. Kao nepromjenjivi dio, tj. baseUrl je korišten url: <https://studentattendancesystem-sas.000webhostapp.com/webservice/>.

|  |  |  |  |
| --- | --- | --- | --- |
| WS METODA | HTTP METODA | URI | TIJELO |
| Prijava profesora | GET | prijava/profesor/{email}/{lozinka}/ |  |
| Prijava studenta | GET | prijava/student/{email}/{lozinka}/ |  |
| Registracija profesora | POST | registracija/profesor/ | {ime},{prezime},{email},{titula} |
| Registracija studenata | POST | registracija/student/ | {ime},{prezime},{email},{slika} |
| Dodavanje kolegija | POST | kolegij/novi/profesor/{idUloge}/ | {naziv}, {semestar},  {studij} |
| Dohvaćanje dvorana | GET | dvorane/dohvati/{tipDvorane}/ |  |
| Dohvaćanje svih kolegija | GET | kolegij/svi/{uloga}/{idUloge}/ |  |
| Dohvaćanje kolegija profesora | GET | kolegij/dohvati/{uloga}/{idUloge}/ |  |
| Dohvaćanje kolegija studente | GET | kolegij/dohvati/{uloga}/{idUloge}/ |  |
| Dodavanje aktivnosti | POST | aktivnost/nova/profesor/ | {profesor},  {dozvoljenoIzostanaka},{pocetak},{kraj},  {danIzvodjenja},  {dvorana},{kolegij},{tipAktivnosti} |
| Dohvaćanje aktivnosti profesora | GET | aktivnost/dohvati/{uloga}/{idUloge}/{tipAktivnosti}/ |  |
| Dohvaćanje aktivnosti studenta | GET | aktivnost/dohvati/{uloga}/{idUloge}/{tipAktivnosti}/ |  |
| Ažuriranje kolegija | POST | kolegij/azuriraj/profesor/{idUloge}/ | {idKolegija},{naziv}, {semestar},  {studij} |
| Brisanje kolegija | GET | kolegij/obrisi/{kolegij} |  |
| Dohvaćanje aktivnosti profesora po tipu aktivnosti | GET | profesor/{id}/aktivnost/{tip} |  |
| Dohvaćanje aktivnosti profesora po kolegiju | GET | profesor/{id}/kolegij/{id}/aktivnost |  |
| Ažuriranje aktivnosti | GET | aktivnost/azuriraj/{id}/{tipAktivnosti}/{pocetak}/{kraj}/{danIzvodjenja}/{dozvoljenoIzostanaka}/{dvorana} |  |
| Brisanje aktivnosti | GET | aktivnost/obrisi/{id} |  |
| Dohvaćanje evidencije | GET | kolegij/{id}/aktivnost/{id}/dolasci/all |  |
| Dohvaćanje evidencije po terminu | GET | kolegij/{id}/aktivnost/{id}/dolasci /{tjedanNastave} |  |
| Dohvaćanje evidencije za studenta | GET | kolegij/{id}/aktivnost/{id}/dolasci/all/student/{id} |  |
| Dohvaćanje evidencije za studenta po terminu | GET | kolegij/{id}/aktivnost/{id}/dolasci /{tjedanNastave}/student/{id} |  |
| Generiranje lozinke za aktivnost | GET | lozinkaAktivnosti/nova/{aktivnost} |  |
| Evidentiraj dolazak | GET | evidentirajDolazak/{aktivnost}/{student}/{lozinka} |  |
| Upis aktivnosti | GET | aktivnost/upis/{aktivnost}/{student} |  |
| Dohvaćanje aktivnosti po danu za profesora | GET | aktivnost/dohvatiPoDanu/{uloga}/{idUloge}/{danIzvodenja}/ |  |
| Dohvaćanje aktivnosti po danu za studenta | GET | aktivnost/dohvatiPoDanu/{uloga}/{idUloge}/{danIzvodenja}/ |  |
| Upisivanje kolegija profesoru | POST | kolegij/dodaj/profesor/{idUloge}/ | {kolegij} |
| Upisivanje kolegija studentu | POST | kolegij/dodaj/student/{idUloge}/ | {kolegij} |

Tablica 1 metode web servisa

Svaka funkcija vraća odgovor JSON formata. Glavni atributi su status, message i dana. Pri čemu status označava uspješno ili neuspješno dohvaćanje podataka sa web servisa, a prikazuju i ispravnost ili neispravnost pri izmjeni ili dodavanju podataka u bazu.

|  |  |  |  |
| --- | --- | --- | --- |
| RESPONSE | Uspješna prijava | Neuspješna prijava (krivi podaci uneseni) | Neuspješna prijava (nedostupna baza podataka) |
| Status | OK | NOT OK | NOT OK |
| Message | „Uspješna prijava“ | Tekst greški | „Došlo je do problema na web servisu.“ |
| data | Niz sa atributom id\_profesora ili id\_studenta | Niz kojem je prijava = „error“ | Niz kojem je prijava = „error“ |

Tablica 2 Struktura odgovora pri metodi za prijavu

|  |  |  |
| --- | --- | --- |
| RESPONSE | Uspješna registracija | Neuspješna registracija |
| Status | OK | NOT OK |
| Message | „Uspješna registacija“ | Tekst greški |
| data | Niz sa atributima email i lozinka | Niz kojem je reg = „error“ |

Tablica 3 Struktura odgovora pri metodi za registraciju

|  |  |  |
| --- | --- | --- |
| RESPONSE | Nema kolegija u bazi | Dohvaćeni su kolegiji |
| Status | OK | OK |
| Message | „Nema zapisa u bazi!“ | „Pronađeni kolegiji!“ |
| data | Atributi elemenata u kolegij su id = -1, naziv = ““, semestar = -1 i studij = ““ | Niz svih kolegija u JSON formatu. |

Tablica 4 Struktura odgovora pri metodi za dohvaćanje kolegija profesora

|  |  |  |
| --- | --- | --- |
| RESPONSE | Kolegij nije dodan. | Kolegij uspješno dodan. |
| Status | NOT OK | OK |
| Message | Tekst greški | „Kolegij je dodan u bazu.“ |
| data | Niz s atributom dodavanje i njegovom vrijednosti error | Niz s nazivom atributa dodavanje i njegovom vrijednosti OK |

Tablica 5 Struktura odgovora pri metodi za dodavanje odgovora

|  |  |  |
| --- | --- | --- |
| RESPONSE | Nisu dohvaćene aktivnosti. | Dohvaćene su aktivnosti |
| Status | NOT OK | OK |
| Message | „Nema aktivnosti za traženog korisnika“ | „Uspješno dohvaćanje“ |
| data | Niz s atributom aktivnost (vrijednost mu je „error“) | Niz aktivnosti s atributima: idKolegija, nazivKolegija, (idAktivnosti, dozvoljenoIzostanaka, pocetak, kraj, danIzvodjenja, dvorana, tipAktivnosti) |

Tablica 6 Struktura odgovora pri metodi za dohvaćanje aktivnosti

|  |  |  |
| --- | --- | --- |
| RESPONSE | Aktivnost nije dodana | Aktivnost uspješno dodana |
| Status | NOT OK | OK |
| Message | Tekst greški | „Uspješno dodano!“ |
| data | Niz s atributom dodavanje i njegovom vrijednosti error | Niz s nazivom atributa dodavanje i njegovom vrijednosti OK |

Tablica 7 Struktura odgovora pri metodi za dodavanje nove aktivnosti

|  |  |  |
| --- | --- | --- |
| RESPONSE | Aktivnost nije ažurirana | Aktivnost uspješno ažurirana |
| Status | NOT OK | OK |
| Message | Tekst greški | „Uspješno ažurirano!“ |
| data | Niz s atributom ažuriranje i njegovom vrijednosti error | Niz s nazivom atributa ažuriranje i njegovom vrijednosti OK |

Tablica 8 Struktura odgovora pri metodi za ažuriranje aktivnosti

|  |  |  |
| --- | --- | --- |
| RESPONSE | Aktivnost nije obrisana | Aktivnost uspješno obrisana |
| Status | NOT OK | OK |
| Message | Tekst greški | „Uspješno obrisano!“ |
| data | Niz s atributom brisanje i njegovom vrijednosti error | Niz s atributom brisanje i njegovom vrijednosti OK |

Tablica 9 Struktura odgovora pri metodi za brisanje aktivnosti

|  |  |
| --- | --- |
| RESPONSE | Generirana lozinka |
| Status | OK |
| nbResults | 0 |
| Message | „Uspješno generirano!“ |
| data | Niz s atributom lozinka |

Tablica 10 Struktura odgovora pri metodi za generiranje lozinke za prisutnost na aktivnosti

|  |  |  |
| --- | --- | --- |
| RESPONSE | Dolazak nije evidentiran | Dolazak je evidentiran |
| Status | NOT OK | OK |
| Message | Tekst greški | „Dolazak je uspješno evidentiran“ |
| data | 'prisustvo'=>'error' | 'prisustvo'=>'prisutan' |

Tablica 11 Struktura odgovora pri metodi za evidentiranje dolazaka

|  |  |  |
| --- | --- | --- |
| RESPONSE | Upis nije uspio | Upis je uspio |
| Status | NOT OK | OK |
| Message | „Kapacitet aktivnosti je popunjen“ | „Uspješno upisana aktivnost“ |
| data | 'upisan'=>'error' | 'upisan'=>'upisan' |

Tablica 12 Struktura odgovora pri metodi za upis na aktivnost

|  |  |  |
| --- | --- | --- |
| RESPONSE | Nema kolegija u bazi | Dohvaćeni su kolegiji |
| Status | OK | OK |
| Message | „Nema zapisa u bazi!“ | „Pronađeni kolegiji!“ |
| data | Atributi elemenata u kolegij su id = -1, naziv = ““, semestar = -1 i studij = ““ | Niz svih kolegija u JSON formatu. |

Tablica 13. Struktura odgovora pri metodi za dohvaćanje kolegija student

# Popis slika

[Slika 1. Arhitektura aplikacije 8](#_Toc536635910)

[Slika 2. ERA model 14](#_Toc536635911)

[Slika 3. Dijagram slučajeva korištenja 19](#_Toc536635912)

[Slika 4. Dijagram klasa prijave 21](#_Toc536635913)

[Slika 5. Dijagram klasa za seminare 22](#_Toc536635914)

[Slika 6. Dijagram klasa za raspored predavanja 24](#_Toc536635915)

[Slika 7.Dijagram klasa za generiranje lozinki 25](#_Toc536635916)

[Slika 8. Dijagram modula 26](#_Toc536635917)

[Slika 9. Dijagram klasa app modula 27](#_Toc536635918)

[Slika 10. Dijagram klasa webservice modula 29](#_Toc536635919)

[Slika 11. Dijagram klasa passwordrecord modula 30](#_Toc536635920)