

Programsko inženjerstvo

Ak. god. 2020./2021.

Online tečajevi

Dokumentacija, Rev. 2

Grupa: JamesBondi

Voditelj: Marin Fabijanić

Datum predaje: 14.01.2021.

Nastavnik: Nikolina Frid

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	20
3.2 Ostali zahtjevi	25
4 Arhitektura i dizajn sustava	26
4.1 Baza podataka	28
4.1.1 Opis tablica	28
4.1.2 Dijagram baze podataka	31
4.2 Dijagram razreda	32
4.3 Dijagram stanja	33
4.4 Dijagram aktivnosti	35
4.5 Dijagram komponenti	37
5 Implementacija i korisničko sučelje	38
5.1 Korištene tehnologije i alati	38
5.2 Ispitivanje programskog rješenja	39
5.2.1 Ispitivanje komponenti	39
5.2.2 Ispitivanje sustava	42
5.3 Dijagram razmještaja	46
5.4 Upute za puštanje u pogon	47
6 Zaključak i budući rad	49
Popis literature	50
Indeks slika i dijagrama	52

Dodatak: Prikaz aktivnosti grupe

53

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Sokolić	16.10.2020.
0.2	Dodan opis projektnog zadatka	Gojević	20.10.2020.
0.3	Dodan prvi dio opisa obrazaca uporabe	Gojević	23.10.2020.
0.3.1	Dodani svi opisi obrazaca uporabe	Gojević	24.10.2020.
0.3.2	Nadopunjen opis projektnog zadatka	Sokolić	24.10.2020.
0.4	Dodani dionici, aktori i njihovi funkcionalni zahtjevi	Sokolić	25.10.2020.
0.4.1	Dodani nefunkcionalni zahtjevi	Gojević	31.10.2020.
0.5	Dodani dijagrami obrazaca uporabe	Gojević	03.11.2020.
0.6	Dodani sekvencijski dijagrami	Sokolić	06.11.2020.
0.7	Dodan opis baze podataka	Gojević	06.11.2020.
0.7.1	Dodan uvod u poglavlje o arhitekturi	Sokolić	12.11.2020.
0.7.2	Dopunjen opis baze podataka i dodan njen dijagram	Gojević	12.11.2020.
0.8	Dodan dijagram razreda	Sokolić	13.11.2020.
1.0	Korigiranje teksta i provjera dokumentacije	Gojević, Sokolić	13.11.2020.
1.1	Koregirana prva verzija dokumentacije	Gojević	07.01.2021.
1.2	Dodano poglavlje Zaključak i budući rad, te potpoglavlje Korištene tehnologije i alati	Gojević	08.01.2021.
1.3	Dodani dijagram stanja i dijagram razmještaja, dovršen dijagram razreda	Sokolić	12.01.2021.
1.4	Dodan dijagram komponenti i završeno poglavlje Ispitivanje programskog rješenja	Sokolić	13.01.2021.
1.5	Dodane Upute za puštanje u pogon, dijagram aktivnosti	Gojević	14.01.2021.
1.6	Dodan dijagram aktivnosti za grupu	Gojević	14.01.2021.
2.0	Konačna verzija		

2. Opis projektnog zadatka

Cilj ovog projekta jest razviti platformu pomoću koje će se moći održavati online tečajevi. U ovo doba pandemije i izolacija, mnogi žele korisnije provesti svoje vrijeme za ekranom te naučiti neku novu vještinu. Ova aplikacija nudit će upravo to - mogućnost pohađanja tečajeva u raznim kategorijama (npr. IT, kuhanje, uređenje doma i vrta...). Tečajevi će sadržavati unaprijed pripremljene materijale poput skripti za polaznike, videosnimki i prezentacija, a unutar aplikacije postojat će i opcija konzultacija uživo putem video poziva.

Pri pokretanju aplikacije, korisniku će se prikazati njen početni zaslon na kojem će moći odabrati opciju prijave s postojećim korisničkim računom ili registracije s novim korisničkim računom. Svi korisnici moraju biti registrirani kako bi mogli koristiti aplikaciju.

Neregistriranom korisniku nudi se opcija kreiranja korisničkog računa predavača ili polaznika. Jedni i drugi unose korisničko ime i lozinku za prijavu u sustav.

Ukoliko se odabere opcija polaznika, korisnik mora unijeti:

- ime
- prezime
- e-mail
- broj kartice za naplatu

Ako se korisnik želi registrirati kao predavač, potrebni podaci su:

- ime
- prezime
- e-mail
- IBAN
- *opcionalno*: fotografija i kratka biografija

Korisnik prijavom daje suglasnost da *Bond&Learn* prikuplja, pohranjuje, elektronički obrađuje osobne podatke za potrebe registracije i korištenja aplikacije uz poštivanje odredbi Uredbe EU o zaštiti pojedinaca u vezi s obradom osobnih podataka i o slobodnom kretanju takvih podataka te Zakona o zaštiti osobnih podataka.

Svi registrirani korisnici mogu pregledavati i mijenjati svoje osobne podatke.

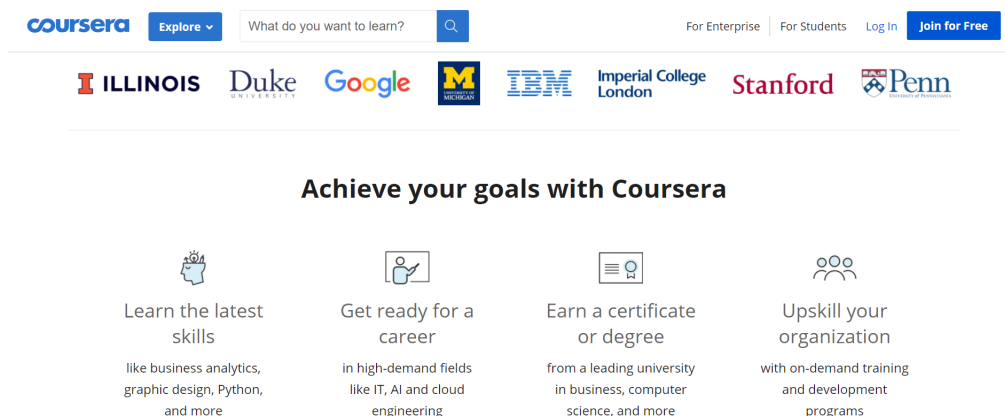
Korisnici registrirani kao **predavač** mogu stvoriti tečaj te ga nakon toga i modificirati (dodavati materijale i ukloniti ga). Tečaj se stvara tako da se najprije odabere jedna od ponuđenih kategorija:

- IT
- kuhanje
- uređenje doma i vrta
- uljepšavanje
- razno

Nakon toga, odabire se razina zahtjevnosti tečaja koja može biti početnička, srednja ili napredna. Zatim se unosi naziv tečaja, kratki opis po želji (maksimalni kapacitet 1000 znakova) i cijena tečaja te se postavljaju materijali za učenje (PDF format do ukupno 50MB) koje je kasnije moguće dodatno uređivati. Korisnici koji prilikom registracije odaberu opciju predavača također imaju mogućnost pregledavanja polaznika svojih tečajeva te praćenja financija na pojedinom tečaju.

Korisnici koji odaberu registraciju kao **polaznik** tečaja mogu pregledavati dostupne tečajeve. Tečajeve mogu pretraživati prema kategorijama i ključnim riječima u samom nazivu tečaja. Kada pronađu tečaj koji odgovara njihovim željama i potrebama, mogu ga i upisati te, nakon uspješne naplate, pristupiti svim materijalima koje je predavač postavio dostupnima za taj tečaj. Polaznicima tečajeva dana je mogućnost slanja zahtjeva za konzultacijama, koje predavač može potvrditi i započeti ili ih otkazati.

Jedno od sličnih rješenja jest interaktivna platforma za online učenje pod nazivom **Coursera** (slika 2.1), koja surađuje sa više od 200 svjetskih sveučilišta i kompanija. Ovo rješenje nudi 5,166 tečajeva u raznim kategorijama, no za sada ne nudi opciju kreiranja vlastitog tečaja, razvija ih isključivo u suradnji s suradnicima.



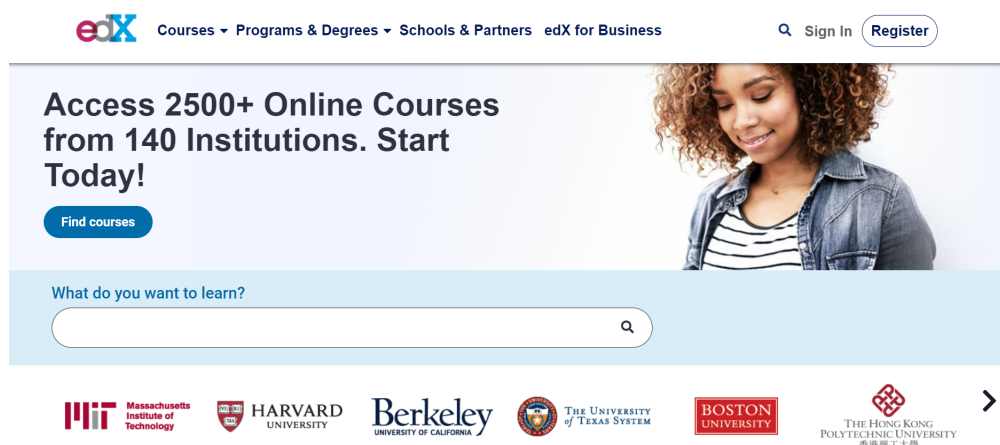
Slika 2.1: Naslovna stranica Coursere

Drugo slično postojeće rješenje jesu online tečajevi **Sveučilišnog računskog centra (SRCE)** (slika 2.2) koji nudi besplatne IT tečajeve svim korisnicima koji su registrirani na sustav za e-učenje, ali kao i prethodno rješenje ne nudi opciju stvaranja vlastitog tečaja.



Slika 2.2: Stranica online tečajeva Sveučilišnog računskog centra

Zadnje već postojeće rješenje koje će biti spomenuto u ovom opisu jest **edX** (slika 2.3), platforma koja također surađuje sa preko 120 sveučilišta diljem svijeta. Nudi preko 2,500 tečajeva u 31 području interesa te je pretežito namijenjena i pogodna studentima.



Slika 2.3: Naslovna stranica platforme edX

Platforma koja se razvija na ovom projektu namijenjena je onima koji žele razviti nove vještine. Zbog velikog broja ponuđenih kategorija, ostvareno rješenje bit će od interesa široj populaciji. Jednostavan postupak stvaranja tečaja omogućit će mnogim predavačima da podijele svoje znanje sa svima zainteresiranim. Velik izbor predavača polaznicima daje mogućnost da odaberu tečaj koji najbolje odgovara njihovim željama i potrebama.

Razvijenu platformu bit će moguće nadograditi. Neke od nadogradnji koje bi dodatno unaprijedile ostvareno rješenje su:

- sustav obavijesti
- mogućnost dijeljenja na društvenim mrežama
- dodatni načini plaćanja
- e-mail marketing

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnici
 - (a) Polaznik
 - (b) Predavač
2. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) registrirati se u sustav
 - i. stvoriti novi korisnički račun polaznika za koji su mu potrebni korisničko ime, lozinka, ime, prezime, e-mail adresa i broj kartice za naplatu **ili**
 - ii. stvoriti novi korisnički račun predavača za koji su mu potrebni korisničko ime, lozinka, ime, prezime, e-mail adresa i IBAN računa, uz opciju stavljanja svoje slike i kratke biografije
2. Predavač (inicijator) može:
 - (a) pregledavati i mijenjati osobne podatke
 - (b) pretraživati ponuđene tečajeve
 - (c) dodati, urediti i obrisati tečaj
 - (d) pregledati polaznike svojih tečajeva
 - (e) prihvatiti ili odbiti termin konzultacija

3. Polaznik (inicijator) može:

- (a) pregledavati i mijenjati osobne podatke
- (b) pretraživati ponuđene tečajeve
- (c) upisati i platiti tečaj
- (d) pristupiti plaćenom tečaju
- (e) preuzimati materijale upisanog tečaja
- (f) poslati zahtjev za konzultacijama

4. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima
- (b) pohranjuje sve podatke o tečajevima i njihove materijale

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Izbor maila i korisničkog imena
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik unosi potrebne podatke
 2. Korisnik je usmjeren na stranicu za unos dodatnih podataka (UC2)
- **Opis mogućih odstupanja:**
 - 1.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnog e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

UC2 - Unos podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješan unos e-mail adrese i korisničkog imena
- **Opis osnovnog tijeka:**
 1. Korisnik unosi potrebne podatke
 2. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 1.a Ostavljen prazan jedan od potrebnih podataka
 1. Sustav obavještava korisnika da mora unijeti podatak
 - 1.b Korisnik odustaje od registracije
 1. Sustav briše cijelu registraciju

UC3 - Prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Pristup sustavu
- **Sudionici:** Baza podataka

- **Preduvjet:** Postojeći račun u sustavu
- **Opis osnovnog tijeka:**
 1. Korisnik unosi e-mail i šifru
 2. Korisnik prima obavijest o uspješnoj prijavi
- **Opis mogućih odstupanja:**
 - 1.a Ostavljen prazan jedan od potrebnih podataka
 1. Sustav obavještava korisnika da mora unijeti podatak
 - 1.b Unesen krivi e-mail ili šifra
 1. Sustav obavještava korisnika o neispravnosti unesenih podataka

UC4 - Pregled svih tečajeva

- **Glavni sudionik:** Polaznik, Predavač
- **Cilj:** Uspješan prikaz svih kreiranih tečajeva
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava
- **Opis osnovnog tijeka:**
 1. Tečajevi su prikazani na aplikaciji

UC5 - Pristup tečaju

- **Glavni sudionik:** Polaznik
- **Cilj:** Pristup sadržaju tečaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava i plaćen tečaj kojem se pristupa
- **Opis osnovnog tijeka:**
 1. Polaznik odabire tečaj
 2. Polaznik je usmjeren na stranicu tečaja

UC6 - Upis tečaja

- **Glavni sudionik:** Polaznik
- **Cilj:** Dobivanje pristupa tečaju
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava i tečaj kojeg polaznik nije još platio
- **Opis osnovnog tijeka:**
 1. Polaznik odabire željeni tečaj
 2. Polaznik je usmjeren na stranicu za naplatu tečaja

UC7 - Naplata tečaja

- **Glavni sudionik:** Polaznik
- **Cilj:** Uspješna naplata tečaja
- **Sudionici:** Baza podataka, vanjski servis za naplatu
- **Preduvjet:** Uspješna prijava
- **Opis osnovnog tijeka:**
 1. Polaznik potvrđuje odabrani tečaj i iznos naplate
 2. Polaznik izvršava plaćanje tečaja
- **Opis mogućih odstupanja:**
 - 2.a Nedovoljno sredstava na kartici
 1. Sustav obavještava polaznika o nemogućnosti pohađanja tečaja zbog financija

UC8 - Dodavanje tečaja

- **Glavni sudionik:** Predavač
- **Cilj:** Uspješno kreiranje tečaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava
- **Opis osnovnog tijeka:**
 1. Predavač unosi potrebne podatke o tečaju
 2. Predavač dodaje materijale za tečaj

UC9 - Brisanje tečaja

- **Glavni sudionik:** Predavač
- **Cilj:** Uspješno ukloniti tečaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava i vlasništvo tečaja
- **Opis osnovnog tijeka:**
 1. Predavač izabire brisanje vlastitog tečaja
 2. Predavač potvrđuje brisanje tečaja
 3. Tečaj je uklonjen iz baze podataka
- **Opis mogućih odstupanja:**
 - 1.a Postoje zahtjevi za konzultacijama na tom tečaju
 1. Sustav otkazuje sve konzultacije

UC10 - Uređivanje tečaja

- **Glavni sudionik:** Predavač
- **Cilj:** Promjena opisa i/ili materijala tečaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava i vlasništvo tečaja
- **Opis osnovnog tijeka:**
 1. Predavač izabire uređivanje vlastitog tečaja
 2. Predavač mijenja sadržaj tečaja i/ili materijala

UC11 - Odjava

- **Glavni sudionik:** Predavač, Polaznik
- **Cilj:** Uspješno se odjaviti iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju Odjave (Log out)
 2. Sustav usmjerava korisnika na stranicu za prijavu

UC12 - Pretraživanje tečajeva

- **Glavni sudionik:** Predavač, Polaznik
- **Cilj:** Pretraživanje tečajeva prema ključnim riječima u imenima tečajeva
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojeći račun u sustavu
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju Search
 2. Korisnik unosi ključnu riječ
 3. Korisniku se prikazuju svi tečajevi koji sadrže ključnu riječ u nazivu ili opisu

UC13 - Pregled osobnih podataka

- **Glavni sudionik:** Predavač, Polaznik
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Odabir opcije "Moji podaci"
 2. Aplikacija prikazuje osobne podatke korisnika

UC14 - Promjena osobnih podataka

- **Glavni sudionik:** Predavač, Polaznik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Uredi moje podatke"
 2. Korisnik mijenja svoje osobne podatke
 3. Korisnik sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik mijenja svoje osobne podatke, ali ne odabire opciju "Spremi moje promjene"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

UC15 - Slanje zahtjeva za konzultacijama

- **Glavni sudionik:** Polaznik
- **Cilj:** Uspješno poslati zahtjev za terminom
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava i plaćen tečaj za kojeg se traže konzultacije
- **Opis osnovnog tijeka:**
 1. Odabir tečaja za kojeg se traže konzultacije
 2. Odabir datuma za termin konzultacija

UC16 - Odgovor na zahtjev za terminom konzultacija

- **Glavni sudionik:** Predavač, polaznik
- **Cilj:** Odgovoriti na zahtjev za terminom konzultacija
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava i vlasništvo tečaja
- **Opis osnovnog tijeka:**
 1. Predavač potvrđuje primljeni zahtjev za terminom konzultacija
 2. Polaznik dobiva obavijest o potvrđenom terminuili
 1. Predavač odbija zahtjev za terminom konzultacija i predlaže novi termin
 2. Polaznik prima obavijest o novom terminu
 3. Polaznik potvrđuje termin ili ga odbija i šalje novi zahtjev (UC15)

UC17 - Video poziv

- **Glavni sudionik:** Predavač, Polaznik
- **Cilj:** Održavanje konzultacija putem video poziva
- **Sudionici:** Baza podataka, vanjski servis za video pozive
- **Preduvjet:** Uspješna prijava, i polaznik i predavač su potvrdili termin konzultacija
- **Opis osnovnog tijeka:**
 1. Predavač i polaznik odabiru opciju uspostave video poziva
 2. Vanjski servis uspostavlja poziv između polaznika i predavača

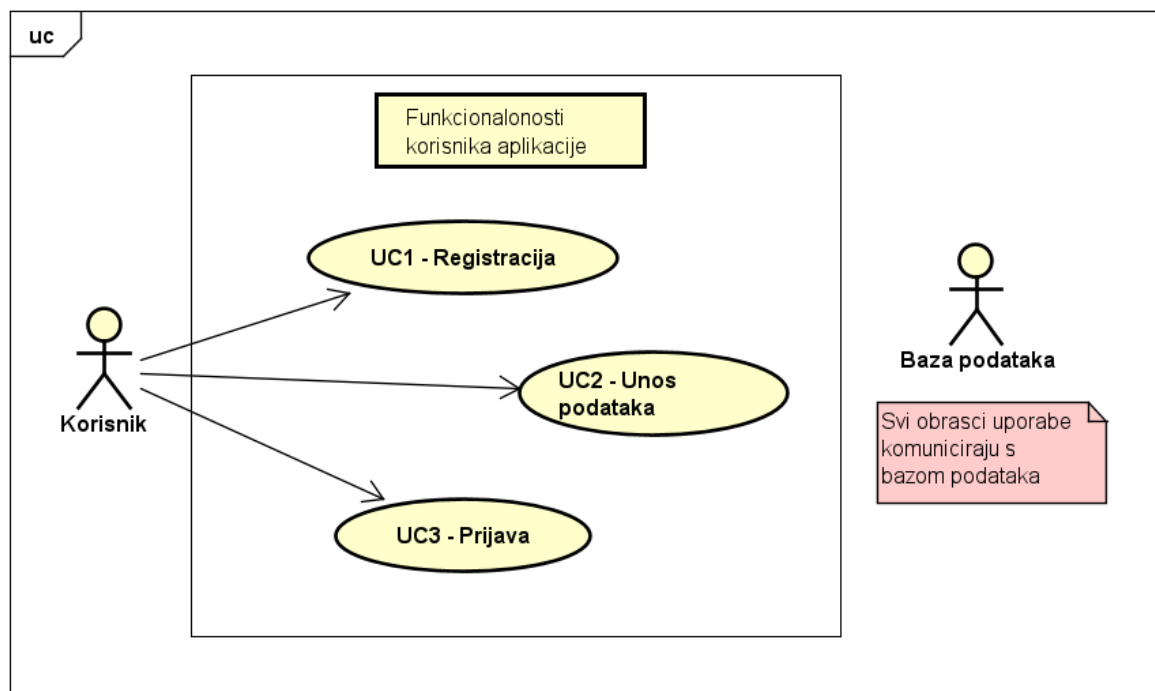
UC18 - Pregled kreiranih tečajeva

- **Glavni sudionik:** Predavač
- **Cilj:** Pregledati sve tečajeve koje je predavač kreirao
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava
- **Opis osnovnog tijeka:**
 1. Predavač odabire opciju Prikaži moje tečajeve (Show your courses)
 2. Predavač je usmjeren na stranicu sa svim njegovim tečajevima

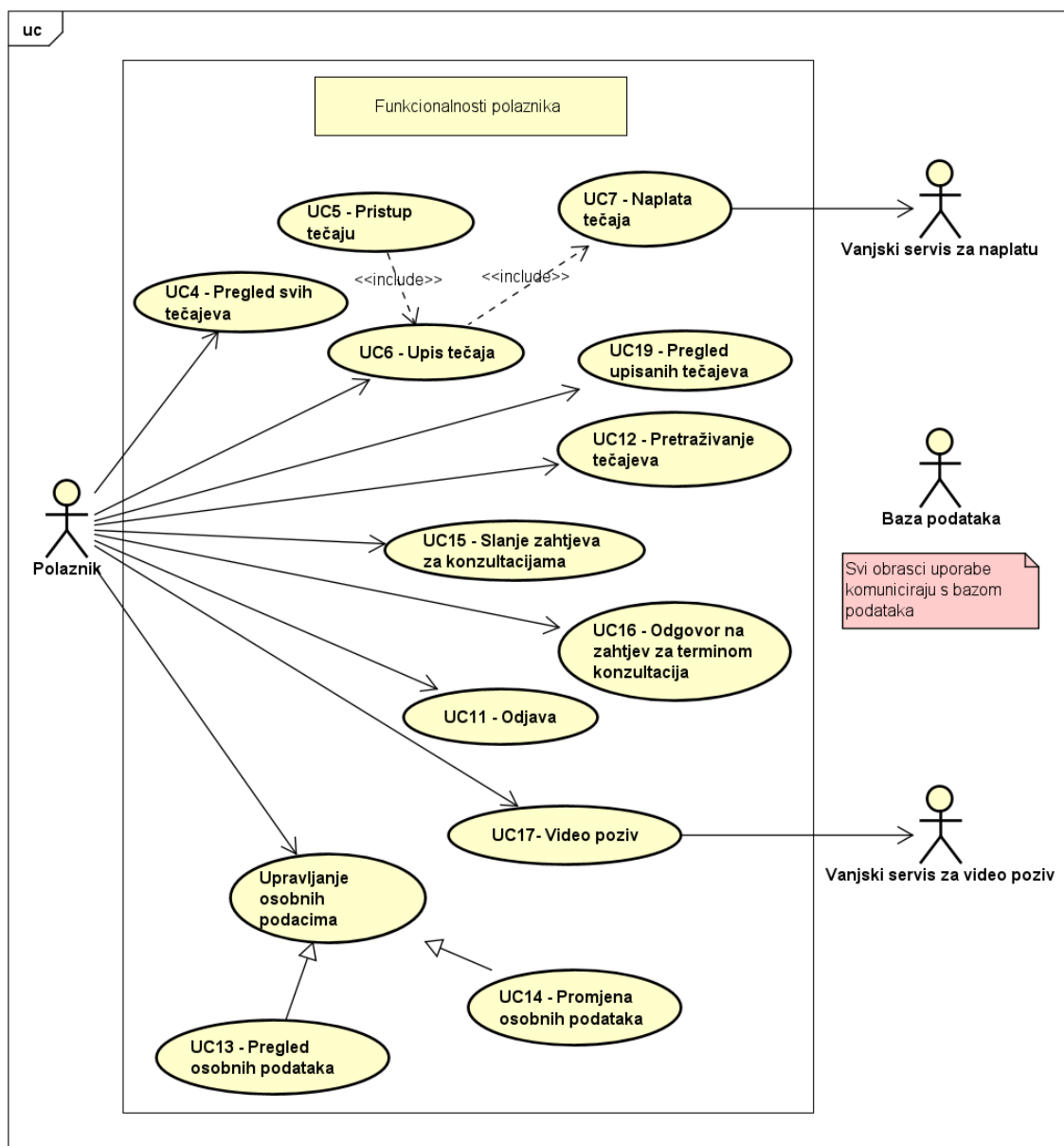
UC19 - Pregled upisanih tečajeva

- **Glavni sudionik:** Polaznik
- **Cilj:** Pregledati sve tečajeve koje je polaznik upisao
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna prijava
- **Opis osnovnog tijeka:**
 1. Polaznik odabire opciju Prikaži moje tečajeve (Show my courses)
 2. Polaznik je usmjeren na stranicu sa svim tečajevima koje je uspješno upisao

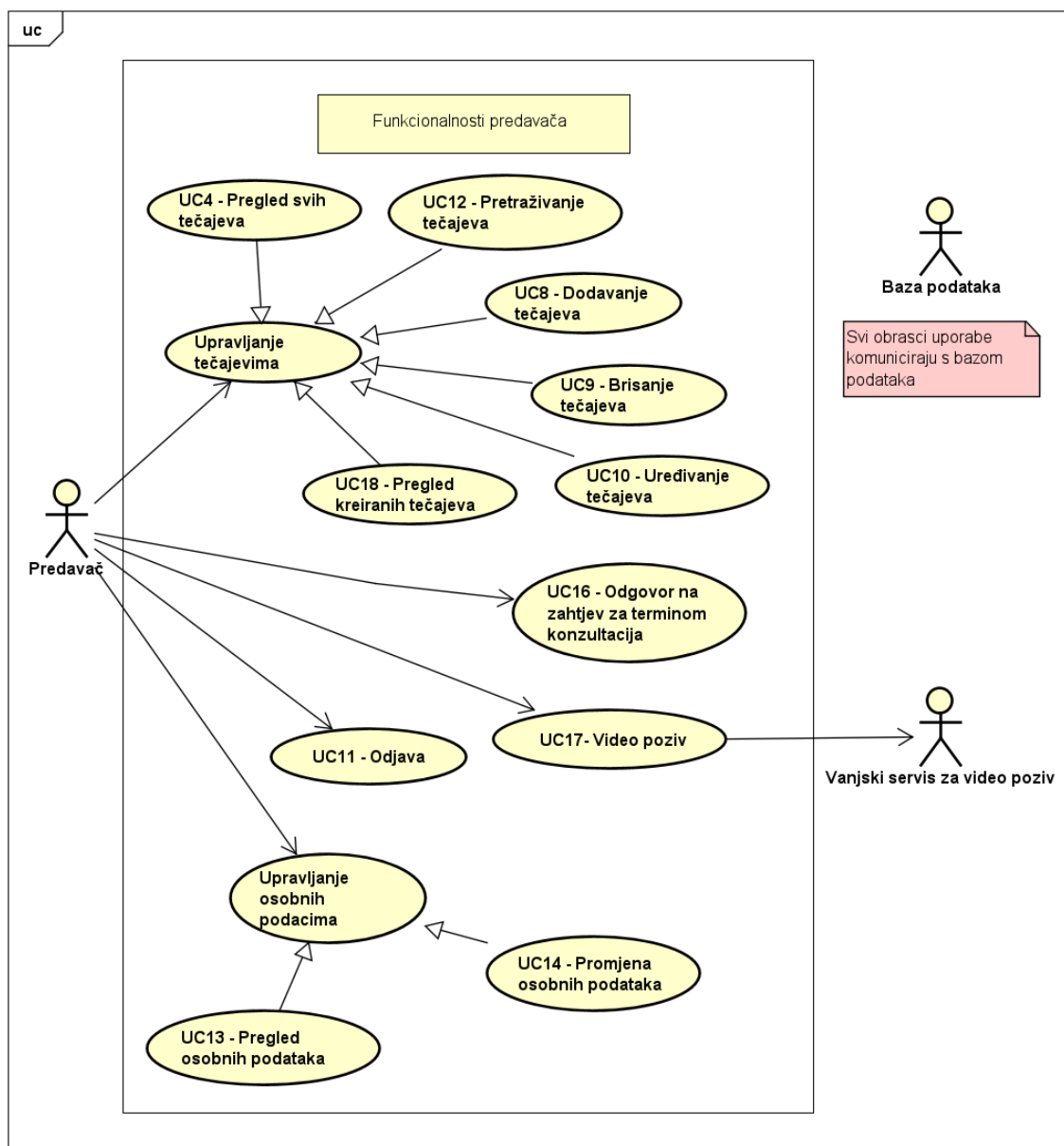
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost polaznika tečaja



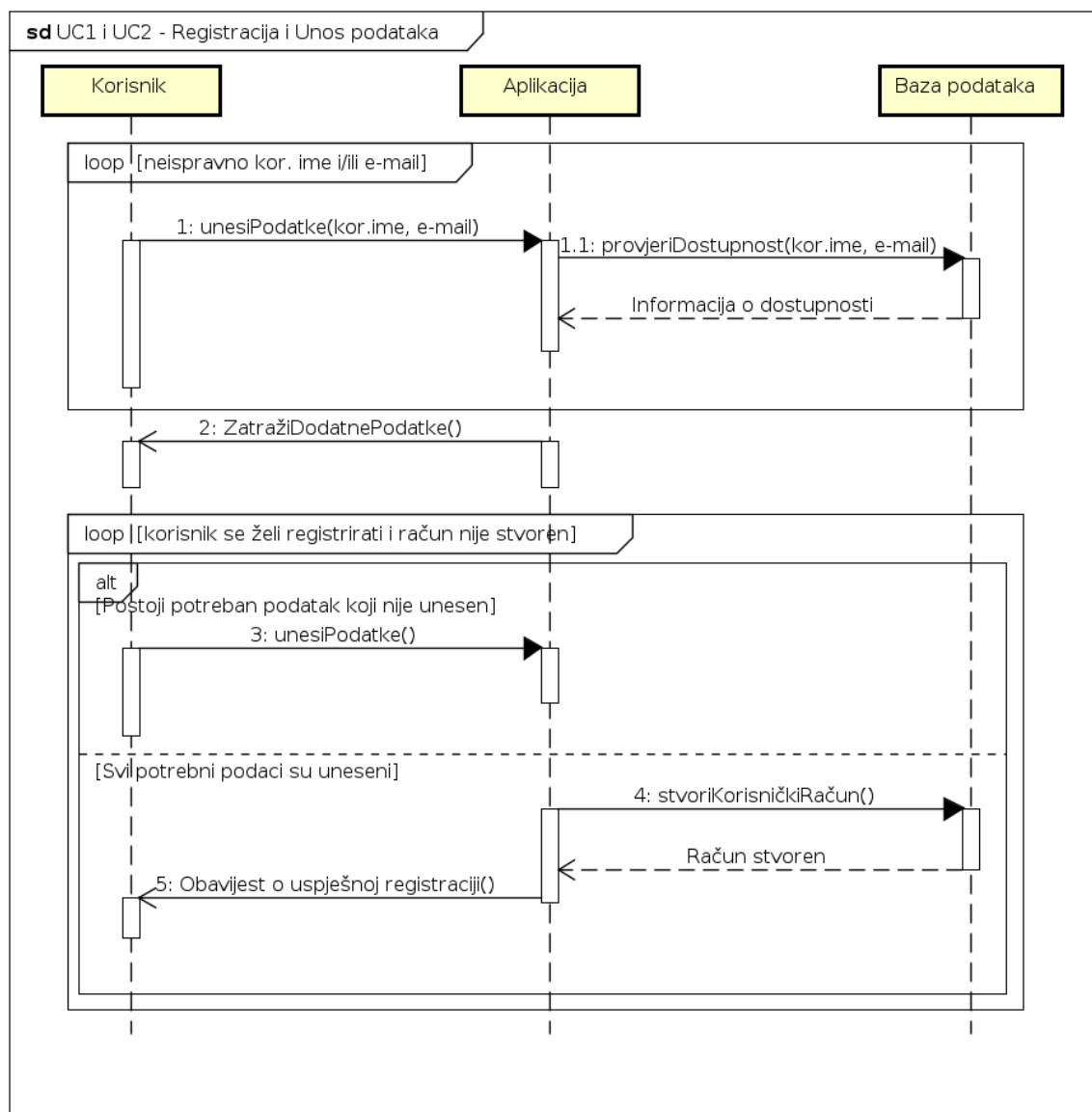
Slika 3.3: Dijagram obrasca uporabe, funkcionalnost predavača

3.1.2 Sekvencijski dijagrami

Obrasci uporabe UC1 i UC2 - Registracija i Unos podataka

Korisnik unosi svoju e-mail adresu i željeno korisničko ime. Poslužitelj s bazom podataka provjerava jesu li unesena e-mail adresa i korisničko ime dostupni (nisu zauzeti od strane drugog korisnika). Poslužitelj također provjerava jesu li podaci u dozvoljenom formatu te je li navedena e-mail adresa ispravna. Ako podaci nisu ispravni, sustav korisniku šalje obavijest o neuspjelom upisu i vraća ga na ponovni unos podataka. Ako su podaci ispravni, korisnika se usmjerava na stranicu za unos dodatnih podataka.

Zatim korisnik unosi dodatne podatke. Ako su potrebni podaci ispravno uneseni, stvara se novi korisnički račun i korisnik dobiva obavijest o uspješnoj registraciji. Ako nije unesen jedan ili više potrebnih podataka, sustav obavještava korisnika. Ukoliko korisnik odluči odustati, registracija se briše.

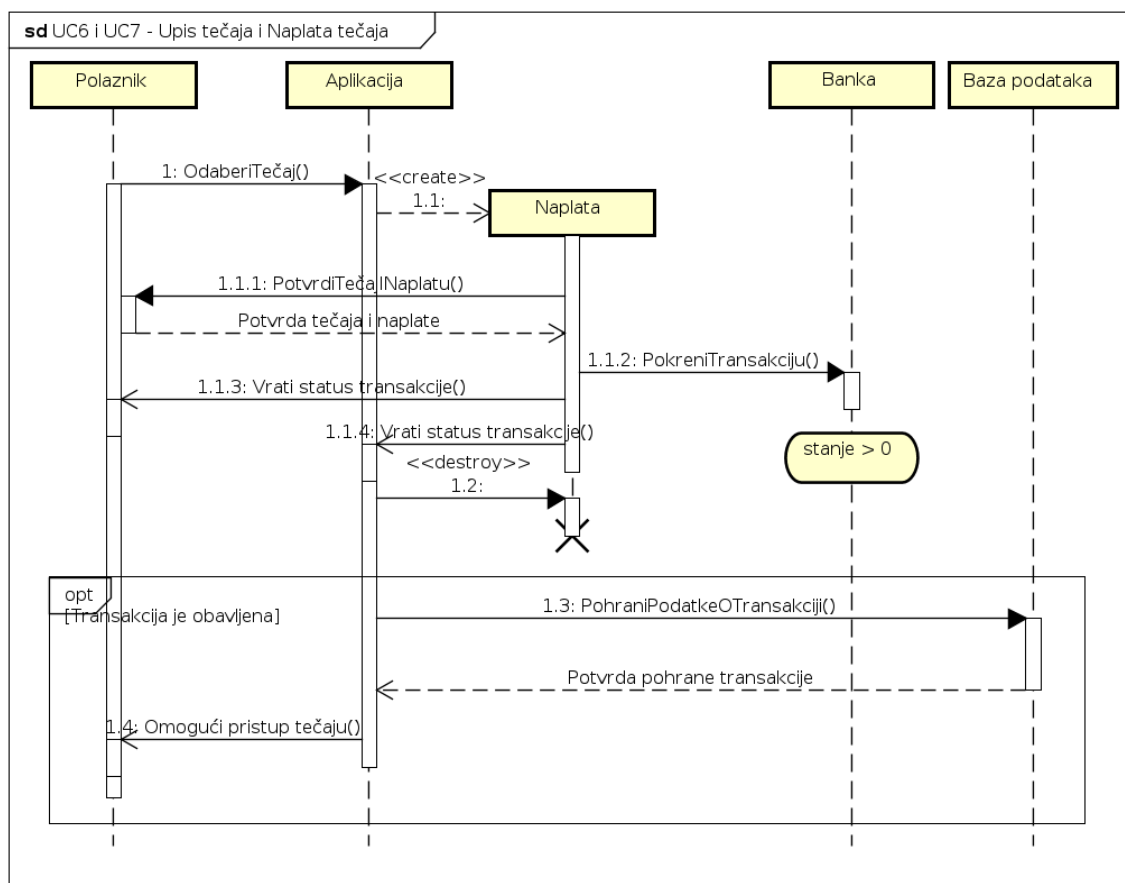


Slika 3.4: Sekvencijski dijagram za UC1 i UC2

Obrasci uporabe UC6 i UC7 - Upis tečaja i Naplata tečaja

Polaznik odabire tečaj koji želi upisati. Nakon odabira tečaja, sustav usmjerava polaznika na stranicu za naplatu.

Polaznik potvrđuje odabrani tečaj i iznos naplate, a zatim izvršava plaćanje tečaja. Ako na kartici ima dovoljno sredstava, tečaj se plaća i korisnik dobiva pristup tečaju. Ukoliko na kartici nema dovoljno sredstava, sustav obavještava korisnika o nemogućnosti pohađanja tečaja zbog financija.

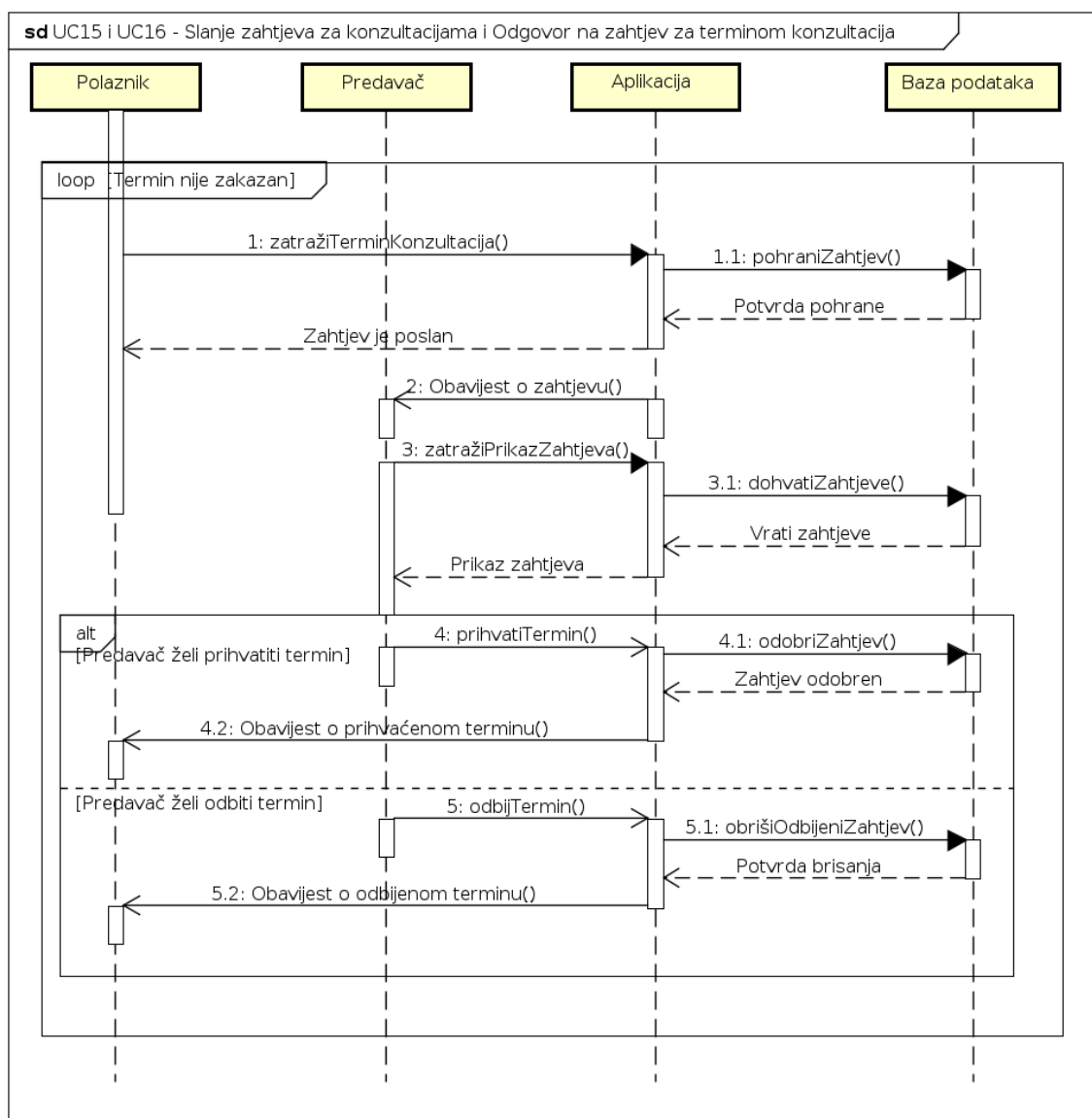


Slika 3.5: Sekvencijski dijagram za UC6 i UC7

Obrasci uporabe UC15 i UC16 - Slanje zahtjeva za konzultacijama i Odgovor na zahtjev za terminom konzultacija

Polaznik šalje zahtjev za terminom konzultacija. Podaci o zahtjevu se pohranjuju u bazu podataka.

Nakon što je zahtjev pohranjen u bazu podataka, poslužitelj predavaču šalje obavijest o dospelom zahtjevu. Predavač šalje poslužitelju zahtjev za prikaz liste zahtjeva za terminima konzultacija. Poslužitelj dohvaća listu zahtjeva iz baze podataka i prikazuje ih predavaču. Predavač može prihvatiti ili odbiti predloženi termin konzultacija. Ako predavač prihvati termin, poslužitelj izmjenjuje zahtjev u bazi podataka koja vraća potvrdu o odobrenom zahtjevu. Poslužitelj tada šalje obavijest polazniku da je zahtjev odobren. Ako predavač odbije termin, poslužitelj briše poslani zahtjev iz baze podataka i obavještava polaznika da je zahtjev odbijen. Polaznik zatim ponovno šalje zahtjev za terminom konzultacija, sve dok mu se zahtjev ne odobri.



Slika 3.6: Sekvencijski dijagram za UC15 i UC16

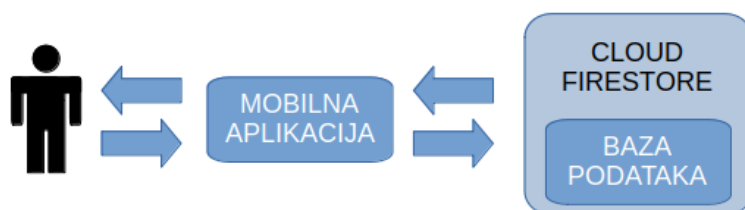
3.2 Ostali zahtjevi

- Sustav treba podržavati rad više korisnika u isto vrijeme
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Neispravno korištenje sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav kao valutu koristi HRK
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Aplikacija mora imati responzivno korisničko sučelje
- Svi osobni podaci u aplikaciji moraju se čuvati u skladu s GDPR regulativom

4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na dva podsustava:

- Mobilna aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Mobilna aplikacija je program koji se izvodi na mobilnom uređaju. Mobilne aplikacije se uglavnom preuzimaju s distribucijskih platformi, nakon čega se provodi njihova instalacija. Korisnik koristi mobilnu aplikaciju za obrađivanje željenih zahtjeva, pri čemu aplikacija po potrebi pristupa bazi podataka.

Baza podataka je organizirani skup strukturiranih podataka. Za pohranu i pristup podacima koristi se računalni sustav.

Za izradu naše mobilne aplikacije odabrali smo **Flutter**, softver za razvoj korisničkog sučelja otvorenog koda koji je stvorio Google. Razlog zbog kojeg smo odabrali Flutter je jednostavnost izrade aplikacija i mogućnost višeplatformskog razvoja korištenjem jednog koda. Još jedna velika prednost je ta što se njime može realizirati i frontend i backend. Flutter koristi programski jezik Dart. Odabrano razvojno okruženje je Visual Studio Code.

Baza podataka koju ćemo koristiti je **Cloud Firestore**. Cloud Firestore je NoSQL, dokumentno orijentirana baza podataka. Dokumenti su organizirani u kolekcije, a svaki dokument ima svoje ime (koje služi kao identifikator) i skup atributa kojima su pridružene vrijednosti. Glavne prednosti ovakve organizacije su efikasni upiti

i velika brzina rada. Cloud Firestore omogućuje iOS, Android i web aplikacijama izravan pristup putem njihovih vlastitih programskih alata.

Arhitektura sustava temeljiti će se na MVVM (Model-View-ViewModel) konceptu. Karakteristika MVVM koncepta je podjela na komponente čiji nezavisni razvoj osigurava fleksibilnost, smanjuje međuovisnost, povećava ponovnu uporabivost i pojednostavljuje ispitivanje.

MVVM koncept se sastoji od:

- **Model** - Pohranjuje podatke i informacije potrebne za rad aplikacije. Odvojen je od logičkog dijela koji određuje prikaz podataka i usluga koje njima manipuliraju.
- **View** - Predstavlja sučelje koje korisnik vidi u interakciji s aplikacijom. Identificira i reagira na korisničke akcije.
- **ViewModel** - Središnja komponenta sustava. Služi kao sučelje između Modela i Viewa. Šalje i prima podatke od Modela te osigurava podatke potrebne Viewu. Također, promatra promjene koje se događaju u Viewu. Raspolaze metodama koje održavaju stanja Viewa i upravlja podacima u Modelu.

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo NoSQL bazu podataka orijentiranu na dokumente, odnosno kolekcije dokumenata. Gradivna jedinka baze je dokument koji je definiran svojim imenom i skupom atributa. Zadaća baze podataka jest jednostavna i brza pohrana, izmjena i dohvat podataka za daljnju obranu. Baza podataka ove aplikacije sastoji se od tri entiteta, a to su:

- User
- Categories
- Meetings

4.1.1 Opis tablica

User Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: about, cardExp, creditCard, firstName, iban, image, lastName, lecturer, mail, secCode, username. Za svakog novog registriranog korisnika kreira se dokument pod šifrom u bazi podataka.

User		
Mail	string	mail korisnika
Username	string	korisničko ime
Lecturer	boolean	oznaka je li korisnik registriran kao predavač
FirstName	string	ime korisnika
LastName	string	prezime korisnika
CardExp	string	datum isteka kartice za naplatu
CreditCard	string	broj kartice za naplatu
SecCode	string	kod za verifikaciju kartice
iban	string	IBAN računa za isplatu honorara
About	string	kratka biografija o korisniku ukoliko je predavač
Image	string	poveznica na fotografiju korisnika ukoliko je predavač
Courses	array	sadrži sve tečajeve koje je korisnik upisao (za polaznika) ili koje je kreirao (za predavača)

Categories Ovaj entitet sadržava sve važne informacije o tečajevima koji su dostupni na aplikaciji. Sadrži atribut name te kolekciju za svaku razinu tečaja (beginner, intermediate, advanced) koja će sadržavati dodatne informacije o pojedinom tečaju. Za svaku kategoriju tečaja kreiran je dokument u bazi podataka, a za svaki tečaj kreira se dokument unutar kolekcije za odrabranu razinu i kategoriju tečaja.

Categories		
Name	string	naziv kategorije tečaja
Difficulty	kolekcija	sadrži tri moguće razine tečajeva: početnička, srednja i napredna

Difficulty Ova kolekcija unutar entiteta Categories sadrži kolekcije tečajeva (pod određenom kategorijom) raspoređene po razinama.

Difficulty		
Beginner	kolekcija	početnička razina tečaja
Advanced	kolekcija	napredna razina tečaja
Intermediate	kolekcija	srednja razina tečaja

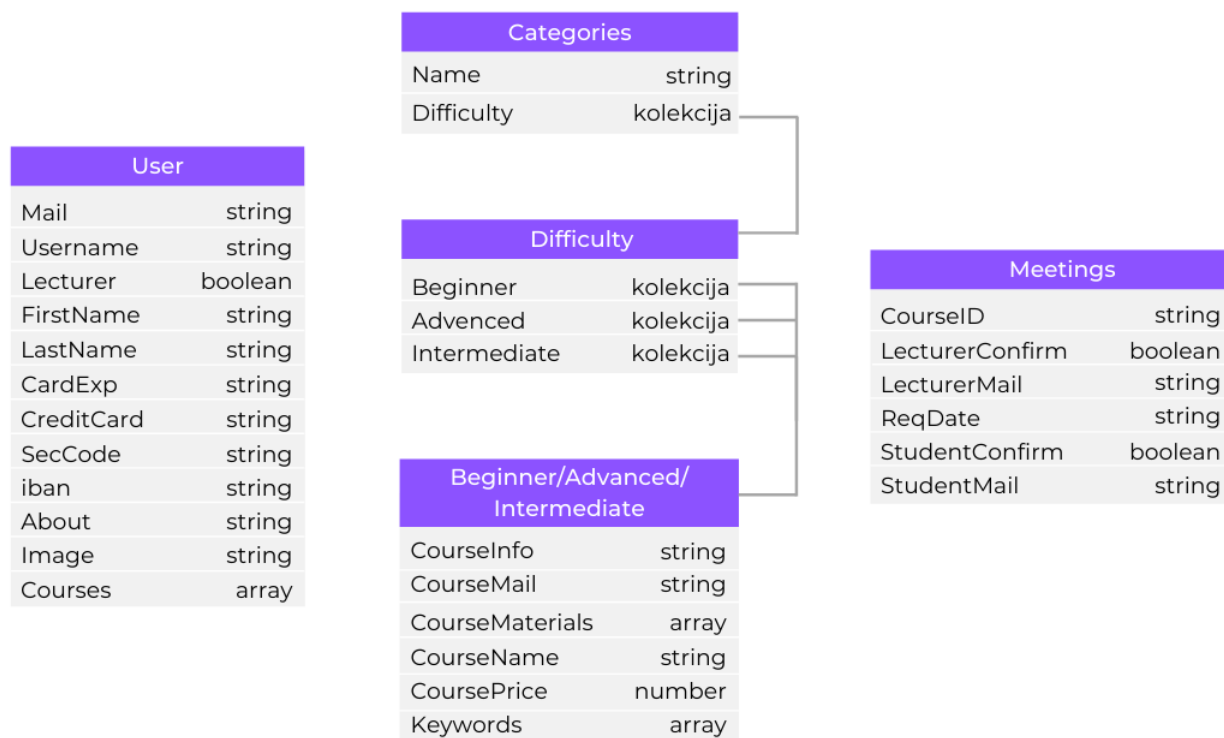
Beginner/Advanced/Intermediate Ove kolekcije sadrže informacije o tečaju koje su nužne za njegovo kreiranje.

Beginner/Advanced/Intermediate		
CourseInfo	string	kratki opis tečaja (do 1000 riječi)
CourseMail	string	e-mail adresa vlasnika tečaja
CourseMaterials	array	materijali za učenje (PDF format do 50 MB ukupno)
CourseName	string	naziv tečaja
CoursePrice	number	cijena tečaja
Keywords	array	ključne riječi po kojima je moguće pronaći tečaj

Meetings Ovaj entitet sadržava sve važne informacije o zatraženim konzultacijama.

Meetings		
CourseID	string	ID dokumenta u bazi podataka za odabrani tečaj
LecturerConfirm	boolean	oznaka da je predavač potvrdio termin konzultacija
LecturerMail	string	e-mail predavača
ReqDate	string	traženi datum konzultacija
StudentConfirm	boolean	oznaka da je polaznik potvrdio termin konzultacija
StudentMail	string	e-mail polaznika koji je zatražio konzultacije

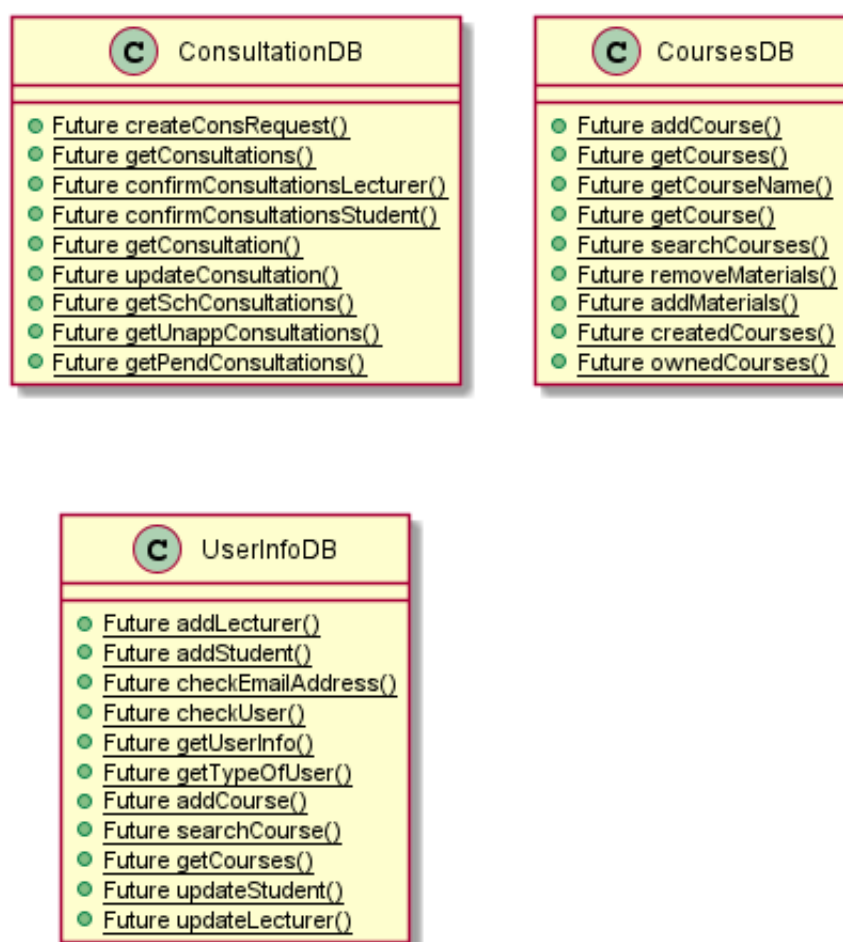
4.1.2 Dijagram baze podataka



Slika 4.2: E-R dijagram baze podataka

4.2 Dijagram razreda

Na slici 4.3 su prikazani razredi koji pripadaju *backend* dijelu MVVM arhitekture. Funkcije implementirane u razredima ConsultationDB, CoursesDB i UserInfoDB manipuliraju podacima u bazi podataka koja predstavlja Model komponentu arhitekture. Sve funkcije koje manipuliraju podacima u bazi podataka (i općenito rade s Firebase metodama) su tipa Future, što znači da su te funkcije asinkrone zbog vremenski zahtjevnih procedura.



Slika 4.3: Dijagram razreda - dio ViewModel

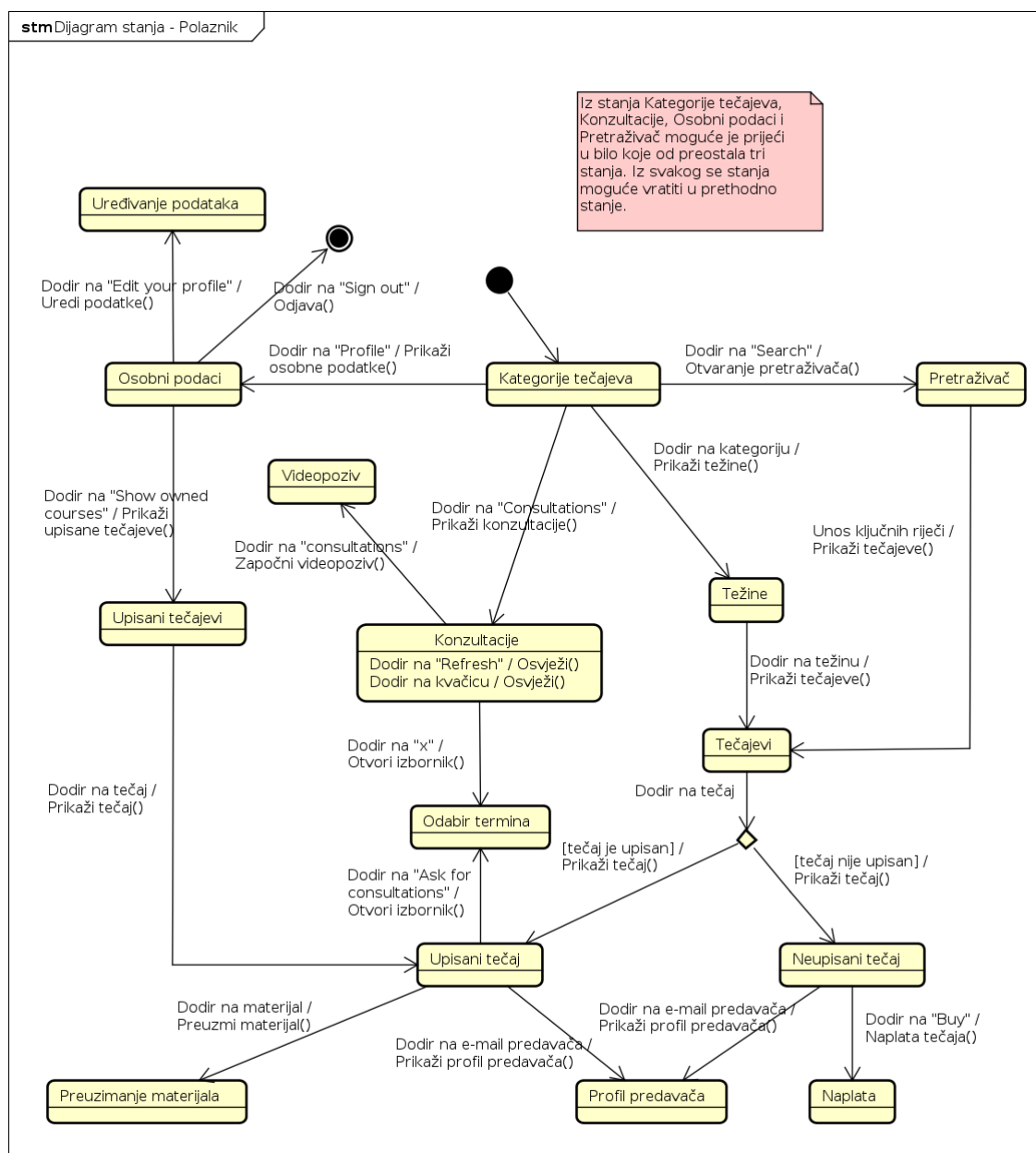
Na poveznici <https://i.ibb.co/tCS1tfL/class-Diagram.png> su prikazani razredi koji pripadaju *frontend* dijelu MVVM arhitekture. View razredi služe za prikaz podataka korisniku. Svi View razredi su implementirani kao State, što znači da se njihov prikaz osvježava sa svakom promjenom stanja razreda.

4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.4 prikazan je dijagram stanja za korisnika koji je registriran kao polaznik. Nakon prijave, polazniku se prikazuje početna stranica na kojoj su navedene kategorije tečajeva. Dodirom na željenu kategoriju polazniku se prikazuje izbornik u kojem odabire težinu tečaja. Aplikacija će zatim prikazati tečajeve koji zadovoljavaju zahtjeve polaznika. Pretraživanje tečajeva moguće je provesti i unosom ključnih riječi u pretraživač kojem se pristupa dodirom na "Search". Dodirom na tečaj prikazuju se njegove informacije. Ako polaznik odabere neupisani tečaj, ima opciju pregleda profila predavača i opciju upisa tečaja koji uključuje naplatu korištenjem vanjskog servisa. Odabirom već upisanog tečaja polazniku se također nudi opcija pregleda profila predavača, kao i opcija preuzimanja materijala. Dodirom na "Ask for consultations" otvara se izbornik koji omogućuje zatraživanje novog termina konzultacija.

Upravljanje konzultacijama polazniku je dostupno dodirom na "Consultations". Na vrhu stranice nalazi se lista termina konzultacija dogovorenih s predavačem. Listu je moguće osvježiti dodirom na "Refresh". Videopoziv se započinje dodirom na "consultations" pokraj pripadnog dogovorenog termina. Na stranici su također prikazani termini konzultacija ponuđeni od strane predavača. Ponuđeni termin moguće je prihvatiti ili odbiti. Ako polaznik prihvati termin, on se pojavljuje u listi dogovorenih termina. U slučaju odbijanja termina, polazniku se otvara izbornik u kojem je moguće predložiti novi termin konzultacija.

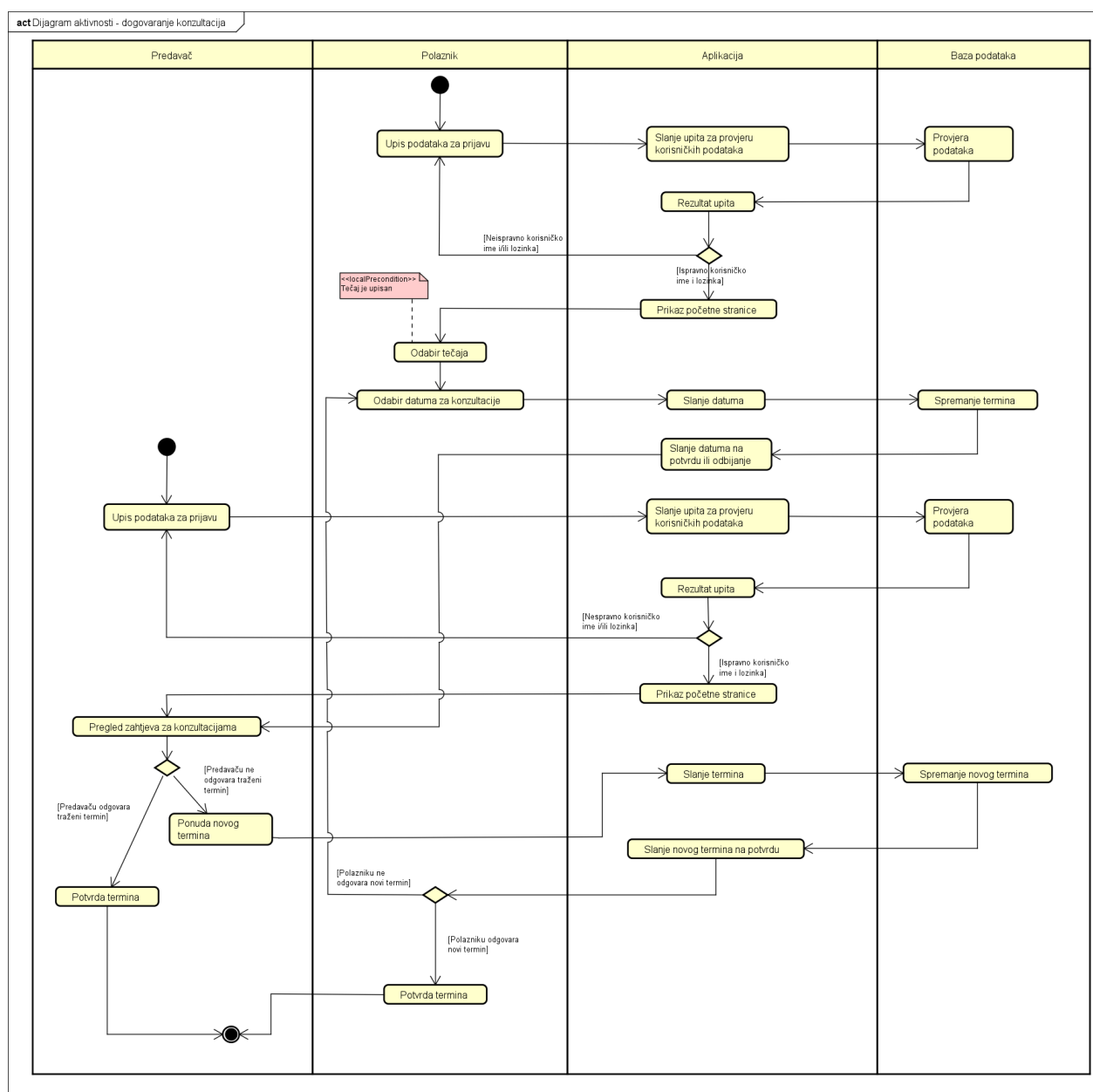
Dodirom na "Profile" polazniku se otvara prikaz osobnih podataka. Polaznik ima opciju izmijeniti osobne podatke dodirom na "Edit your profile", kao i opciju prikaza upisanih tečajeva dodirom na "Show owned courses". Odjava se provodi dodirom na "Sign out".



Slika 4.4: Dijagram stanja - Polaznik

4.4 Dijagram aktivnosti

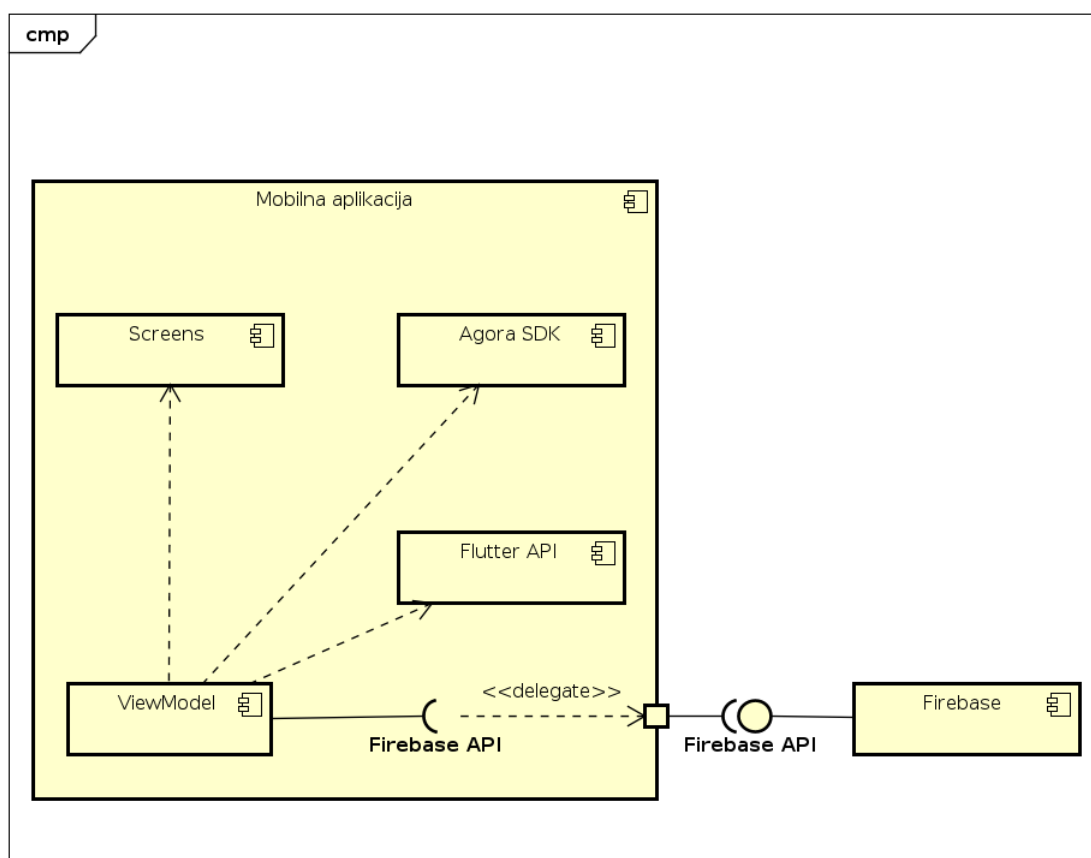
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti 4.5 prikazan je proces dogovaranja konzultacija putem video poziva. Polaznik se prijavi u sustav te na stranici tečaja odabire datum konzultacija koji bi on htio. Zatim predavač ima opciju potvrditi taj termin ili ga odbiti uz ponudu novog termina. Polaznik zatim dobiva obavijest o potvrđenom terminu ili zahtjev za novim terminom od predavača.



Slika 4.5: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazuje odnos komponenti u sustavu. Na slici 4.6 prikazana je struktura cijele aplikacije. Podaci potrebni za rad aplikacije nalaze se u NoSQL bazi podataka pohranjenoj u oblaku, kojoj se pristupa preko sučelja koje nudi Firebase. ViewModel je središnja komponenta sustava koja u sebi sadrži mapu 'components'. ViewModel koristi Flutter API, te Agora SDK za uslugu uspostavljanja videopoziva. Razredi koji služe za prikaz podataka korisniku nalaze se u mapi Screens.



Slika 4.6: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacija WhatsApp¹ i Messenger za Facebook², a sastanci su održavani putem platformi Microsoft Teams³ te Discord⁴. Za izradu UML dijagrama korišten je alat Astah UML⁵, a kao sustav za upravljanje izvornim kodom Git⁶. Udaljeni repozitorij projekta dostupan je na web platformi GitLab⁷.

Kao razvojno okruženje korišten je Visual Studio Code⁸ – uređivač koda tvrtke Microsoft. Dostupan je za Window, Linux i macOS. Uključuje podršku za otklanjanje pogrešaka, isticanje sintakse, inteligentno dovršavanje koda, refaktoriranje koda i ugrađeni Git. Inicijalno ima ugrađenu podršku za JavaScript, TypeScript i Node.js, no ima bogat sustav ekstenzija za druge jezike (na primjer C++, Java, Python, Dart) i druge radne okvire (.NET, Unity...)

Aplikacija je napisana korištenjem softera Flutter⁹ i programskog jezika Dart¹⁰. Flutter je softver za razvoj korisničkog sučelja otvorenog koda, razvijen od tvrtke Google. Omogućava višeplatformski razvoj korištenjem jednog koda te jednostavnost izrade aplikacija. Flutter koristi programski jezik Dart, kojim je realiziran i frontend i backend.

Kako bi unutar aplikacije bili omogućeni video pozivi, korišten je Agora RTC¹¹ servis koji nudi opcije video poziva, glasovnih poziva, live audio i video streaming te slanje poruka.

Baza podataka nalazi se na poslužitelju u oblaku Firestore¹².

¹<https://www.whatsapp.com/>

²<https://www.messenger.com/>

³<https://www.microsoft.com/hr-hr/microsoft-365/microsoft-teams/>

⁴<https://discord.com/>

⁵<https://astah.net/products/astah-uml/>

⁶<https://git-scm.com/>

⁷<https://gitlab.com/>

⁸<https://code.visualstudio.com/>

⁹<https://flutter.dev/>

¹⁰<https://dart.dev/>

¹¹<https://www.agora.io/en/>

¹²<https://firebase.google.com/docs/firestore>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitivanje komponenti verificira rad programskih dijelova koje je moguće neovisno zasebno ispitati. Za ispitivanje smo koristili Flutter Test i MockFirestore priključke.

Kod na slici 5.1 provodi ispitivanje funkcionalnosti stvaranja tečaja korištenjem funkcije `addCourse` unutar razreda `CoursesDB`. U testu su obavljena tri dodavanja tečaja. Nakon toga se provjerava sadrži li kolekcija tri elementa.

```
test('Add course', () async {  
  final firestoreMock = MockFirestoreInstance();  
  await addCourse(firestoreMock, 'Test 1');  
  await addCourse(firestoreMock, 'Test 2');  
  await addCourse(firestoreMock, 'Test 3');  
  final snapshot = await firestoreMock.collection('courses').get();  
  expect(snapshot.docs.length, 3);  
});
```

Slika 5.1: `addCourse`

Slika 5.2 prikazuje kod koji provjerava ispravnost dodavanja novog korisnika u bazu podataka pomoću metode `addUser`. E-mail adrese su jedinstvene unutar baze podataka, odnosno nije dozvoljeno da dva korisnika imaju istu e-mail adresu. Prva dva e-maila su različita i nakon prva dva poziva funkcije `addUser` dokument bi trebao imati dva elementa. Treći poziv metode sadrži već korištenu e-mail adresu i zbog toga se očekuje da je veličina kolekcije i dalje dva.

```
test('Add unique user to database', () async {  
  final firestoreMock = MockFirestoreInstance();  
  await addUser(firestoreMock, 'test@mail.com');  
  await addUser(firestoreMock, 'test1@mail.com');  
  final snapshot = await firestoreMock.collection('users').get();  
  expect(snapshot.docs.length, 2);  
  
  await addUser(firestoreMock, 'test@mail.com');  
  final snapshotNew = await firestoreMock.collection('users').get();  
  expect(snapshotNew.docs.length, 2);  
});
```

Slika 5.2: Dodavanje novog korisnika

Kod na slici 5.3 služi za provjeru prijavljenih korisnika. Test očekuje da nema prijavljenih korisnika, tj. da je user jednak null.

```
test('No user signed in', () async {  
  final auth = MockFirebaseAuth();  
  final user = auth.currentUser;  
  expect(user, isNull);  
});
```

Slika 5.3: Provjera prijavljenih korisnika

Prijava korisnika prikazana je na slici 5.4. Nakon unosa e-maila i lozinke očekuje se da FirebaseAuth ima prijavljenog korisnika.

```
test('Credential login', () async {  
  final auth = MockFirebaseAuth();  
  final result = await auth.signInWithEmailAndPassword(  
    email: 'test@mail.com', password: 'password');  
  final user = result.user;  
  
  expect(user.uid, isNotEmpty);  
  expect(user.displayName, isNotEmpty);  
  expect(auth.authStateChanges(), emitsInOrder([isA<User>()]));  
  expect(user.isAnonymous, isFalse);  
});
```

Slika 5.4: Prijava u sustav

Ispitni slučaj na slici 5.5 vraća korisnika koji je prijavljen.

```
test('Returns user if already signed in', () async {  
  var auth = MockFirebaseAuth(signedIn: true);  
  
  final user = auth.currentUser;  
  expect(user.uid, isNotEmpty);  
  expect(user.displayName, isNotEmpty);  
});
```

Slika 5.5: Prikaz prijavljenog korisnika

Kod na slici 5.6 provjerava je li odjava korisnika uspješno provedena.

```
test('Returns null after sign out', () async {  
  var auth = MockFirebaseAuth(signedIn: true);  
  
  final user = auth.currentUser;  
  
  await auth.signOut();  
  
  expect(auth.currentUser, isNull);  
  expect(auth.authStateChanges(), emitsInOrder([user, null]));  
});
```

Slika 5.6: Provjera odjave

Rezultati testova vidljivi su na slici 5.7.

```
✓ Add course  
✓ Add unique user to database  
✓ No user signed in  
✓ Credential login  
✓ Returns user if already signed in  
✓ Returns null after sign out  
Exited
```

Slika 5.7: Rezultati testova

Aplikacija je prošla sve testove.

5.2.2 Ispitivanje sustava

Ispitivanje sustava verificira funkcijske i nefunkcijske zahtjeve te prihvatljivost sustava. Automatizirane testove smo proveli pomoću Flutter Drivera.

Ispitni slučaj na slici 5.8 simulira pokušaj prijave s neispravnom lozinkom. Nakon unosa e-mail adrese, očekuje se da će se ona nalaziti u svom polju za unos. Jednako tako se očekuje i za lozinku.

```
test('Enters wrong credentials', () async {  
  await driver.tap(mailInput);  
  await driver.enterText('fabijanicmarin6@gmail.com');  
  await driver.waitFor(mailInput);  
  //await driver.tap(button);  
  
  expect(await driver.getText(mailInput), 'fabijanicmarin6@gmail.com');  
  
  await driver.tap(passInput);  
  await driver.enterText('test12');  
  await driver.waitFor(passInput);  
  //await driver.tap(button);  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  expect(await driver.getText(passInput), 'test12');  
  
  await driver.tap(button);  
  await driver.waitFor(wrongPassWindow);  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  await driver.tap(okButton);  
});
```

Slika 5.8: Neispravan unos korisničkih podataka

Kod na slici 5.9 prikazuje prijavu s ispravnom lozinkom. Ovaj put se nakon ponovnog unosa šifre očekuje da će se u polju za unos nalaziti ispravna lozinka, te da će se na pritisak gumba izvršiti prijava.

```
test('Enters good credentials', () async {  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  expect(await driver.getText(passInput), '');  
  await driver.tap(passInput);  
  await driver.enterText('test123');  
  await driver.waitFor(passInput);  
  expect(await driver.getText(passInput), 'test123');  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  await driver.tap(button);  
  await Future<void>.delayed(Duration(milliseconds: 750));  
});
```

Slika 5.9: Ispravan unos korisničkih podataka

Kod na slici 5.10 očekuje da će se, nakon odabira kategorije i težine tečaja, u polju za unos nalaziti naziv tečaja (ovdje je to polje ostavljeno prazno, što se i očekuje). Zatim test očekuje da će se u polju za unos opisa tečaja nalaziti opis, međutim tečaj neće biti dodan jer nije unesen njegov naziv.

```
test('Enters invalid course name', () async {  
  await driver.tap(plusbutton);  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  
  await driver.tap(category);  
  await Future<void>.delayed(Duration(milliseconds: 350));  
  await driver.tap(difficulty);  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  
  await driver.tap(courseName);  
  await driver.enterText('');  
  await driver.waitFor(courseName);  
  expect(await driver.getText(courseName), '');  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  
  await driver.tap(shortDesc);  
  await driver.enterText('Unos opisa u automatskom testu');  
  await driver.waitFor(shortDesc);  
  expect(await driver.getText(shortDesc), 'Unos opisa u automatskom testu');  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  
  await driver.tap(price);  
  await driver.enterText('100');  
  await driver.waitFor(price);  
  expect(await driver.getText(price), '100');  
  await Future<void>.delayed(Duration(milliseconds: 750));  
  await driver.scrollUntilVisible(find.byValueKey('add-scroll'), addbutton);  
  await Future<void>.delayed(Duration(milliseconds: 450));  
  await driver.tap(addbutton);  
});
```

Slika 5.10: Neispravan naziv tečaja

Kod na slici 5.11 očekuje da će se nakon ponovnog unosa naziva tečaja on nalaziti u odgovarajućem polju za unos, te da će se izvršiti dodavanje tečaja.

```
test('Enters valid course name', () async {
  await Future<void>.delayed(Duration(milliseconds: 250));
  await driver.scrollUntilVisible(
    find.byValueKey('add-scroll'), courseName);
  await Future<void>.delayed(Duration(milliseconds: 450));
  await driver.tap(courseName);
  await driver.enterText('Test Ime');
  await driver.waitFor(courseName);
  expect(await driver.getText(courseName), 'Test Ime');
  await Future<void>.delayed(Duration(milliseconds: 750));
  await driver.scrollUntilVisible(find.byValueKey('add-scroll'), addbutton);
  await Future<void>.delayed(Duration(milliseconds: 450));
  await driver.tap(addbutton);
});
```

Slika 5.11: Unos ispravnog naziva tečaja

Slika 5.12 prikazuje kod koji simulira odlazak korisnika na svoj profil i pokušaj izmjene osobnih podataka. Korisnik ispravno unosi sve potrebne podatke i uređuje svoj profil. Nakon toga, test se vraća na profilnu stranicu korisnika i provjerava izmijenjene podatke.

```
test('Enters valid profile info', () async {
  String firstNameInput = 'Marin';
  String lastNameInput = 'Fabijanac';
  String aboutInput = 'About text added through automated test';
  await driver.tap(profileButton);
  await Future<void>.delayed(Duration(milliseconds: 750));

  await driver.tap(menu);
  await Future<void>.delayed(Duration(milliseconds: 750));
  await driver.tap(editProfile);
  await Future<void>.delayed(Duration(milliseconds: 750));

  await driver.tap(firstName);
  await driver.enterText(firstNameInput);
  await driver.waitFor(firstName);
  expect(await driver.getText(firstName), firstNameInput);
  await Future<void>.delayed(Duration(milliseconds: 750));

  await driver.tap(lastName);
  await driver.enterText(lastNameInput);
  await driver.waitFor(lastName);
  expect(await driver.getText(lastName), lastNameInput);
  await Future<void>.delayed(Duration(milliseconds: 750));

  await driver.tap(about);
  await driver.enterText(aboutInput);
  await driver.waitFor(about);
  expect(await driver.getText(about), aboutInput);
  await Future<void>.delayed(Duration(milliseconds: 750));

  await Future<void>.delayed(Duration(milliseconds: 750));
  await driver.scrollUntilVisible(
    find.byValueKey('edit-scroll'), updatebutton);
  await driver.tap(updatebutton);

  await Future<void>.delayed(Duration(milliseconds: 750));
  await driver.tap(categoriesbutton);
  await driver.tap(profileButton);

  expect(await driver.getText(find.byValueKey('firstLast')),
    firstNameInput + ' ' + lastNameInput);
  expect(await driver.getText(find.byValueKey('aboutCheck')), aboutInput);
});
```

Slika 5.12: Uređivanje osobnih podataka

Rezultati testova vidljivi su na slici 5.13.

```
PS C:\Users\Marin\Desktop\jamesbondi> flutter drive --target=test_driver/app.dart
Using device Android SDK built for x86.
Starting application: test_driver/app.dart
Running Gradle task 'assembleDebug'...
Running Gradle task 'assembleDebug'... Done                    5.4s
✓ Built build/app/outputs/flutter-apk/app-debug.apk.
I/flutter (14613): Observatory listening on http://127.0.0.1:33424/jUPwJ2vyMpw=/
00:00 +0: Login Test (setUpAll)

00:00 +0: Login Test Enters wrong credentials

00:04 +1: Login Test Enters good credentials

00:06 +2: Login Test (tearDownAll)

00:06 +2: Add Course Test (setUpAll)

00:06 +2: Add Course Test Enters invalid course name

00:13 +3: Add Course Test Enters valid course name

00:16 +4: Add Course Test (tearDownAll)

00:16 +4: Edit Profile Test (setUpAll)

00:16 +4: Edit Profile Test Enters valid profile info

00:23 +5: Edit Profile Test (tearDownAll)

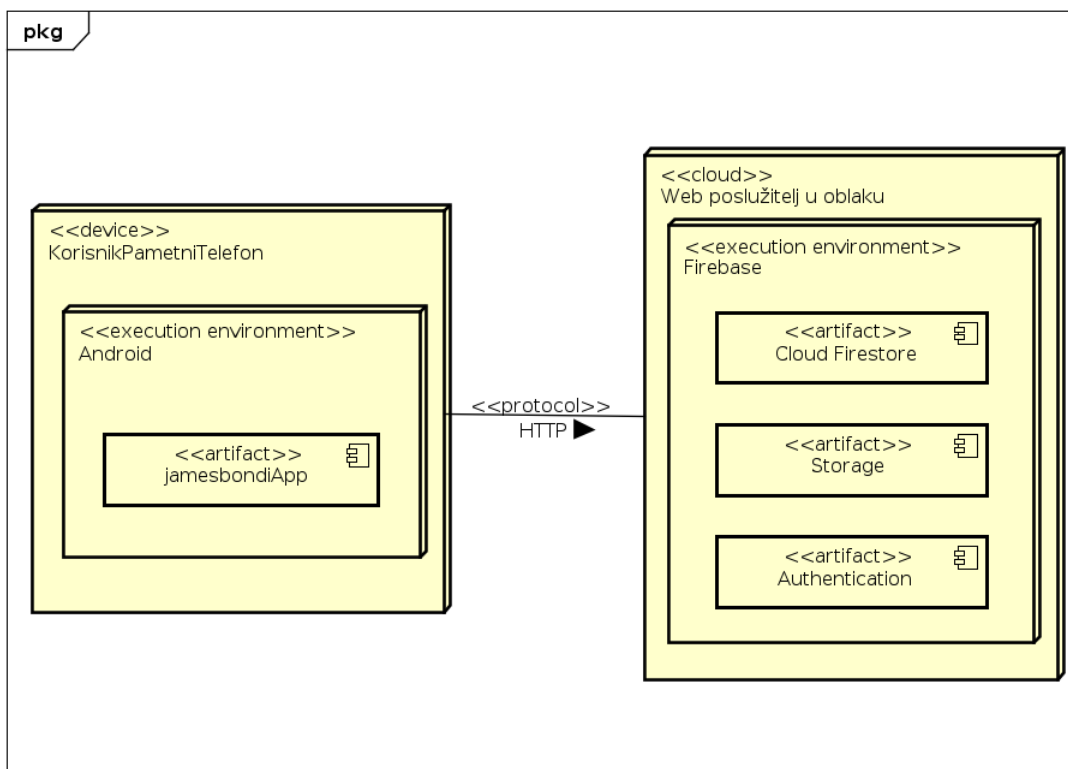
00:23 +5: All tests passed!
```

Slika 5.13: Rezultati integracijskih testova

Aplikacija je prošla sve testove.

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Na slici 5.14 prikazan je specifikacijski dijagram razmještaja koji prikazuje komunikaciju mobilnog uređaja s operacijskim sustavom Android na kojem se nalazi mobilna aplikacija s bazom podataka koja se nalazi u oblaku. Mobilna aplikacija spaja se HTTP protokolom na oblak. U oblaku se unutar Firebase platforme nalazi baza podataka Cloud Firestore, usluga za pohranu podataka Storage i Firebase Authentication za određivanje identiteta korisnika.



Slika 5.14: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Postavljanje baze podataka

Potrebno je kreirati Firebase račun preko kojega je moguće upravljati svime što je povezano uz bazu podataka. Nakon toga, treba se kreirati Firebase projekt. Kada je stvoren, potrebno je unutar aplikacije spojiti Firebase API preko API ključa i unijeti ID projekta, nakon čega je aplikacija povezana s bazom podataka.

Nakon toga, potrebno je postaviti Firebase authentication, odnosno omogućiti i postaviti metode za prijavu korisnika. U našem slučaju, jedini mogući način prijave jest putem maila. Idući korak u postavljanju naše baze podataka jest inicijalizacija baze podataka na Firestore servisu. Nakon toga, stvara se hijerarhija koja se može vidjeti prethodno u dokumentaciji (slika 4.2).

Zadnji korak je prilagođavanje pravila privatnosti svih Firebase servisa, kako bismo osigurali osjetljive podatke koji će se nalaziti u bazi podataka.

Postavljanje Agora RTC servisa

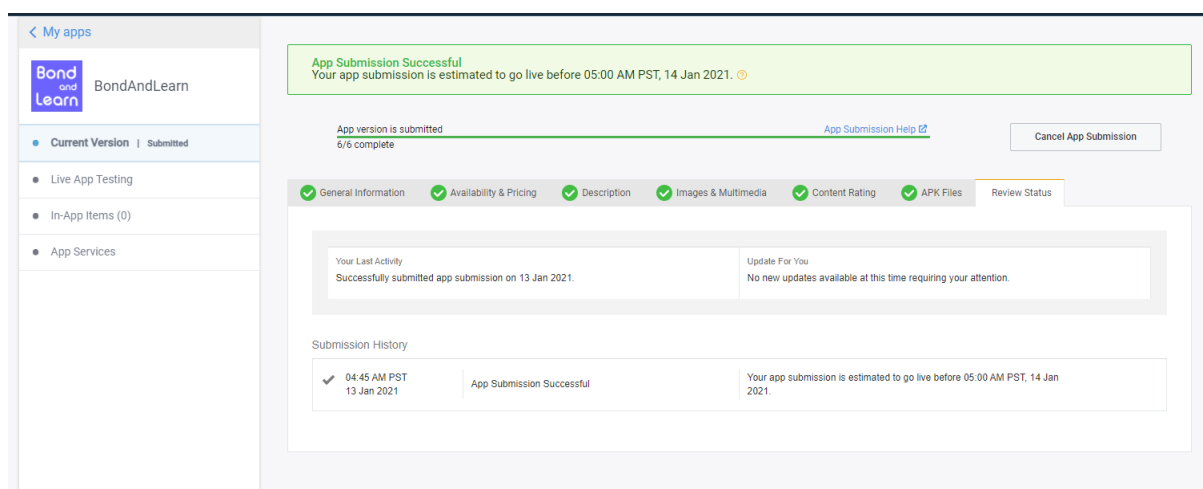
Kako bi ostvarili mogućnost uspostavljanja video poziva unutar aplikacije, potrebno je bilo kreirati račun na servisu Agora. Nakon stvaranja računa, stvoren je novi projekt te je tako dobiven poseban AppID koji smo povezali s našom aplikacijom.

Izgradnja Android aplikacije

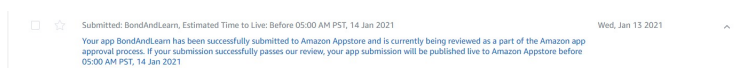
Za izgradnju Android aplikacije u Flutteru, prvo je potrebno promijeniti ikonu. Nakon toga, potrebno je digitalno potpisati aplikaciju, odnosno generirati ključ kojeg zatim treba referencirati u posebnoj datoteka. Idući korak jest uređivanje build.gradle datoteke kako bi konfigurirali digitalni potpis aplikacije. Nakon toga, potrebno je App Manifest datoteku i build.gradle datoteku prilagoditi sukladno uputama na *poveznici* te zatim kreirati APK datoteku.

Puštanje Android aplikacije u pogon

Za puštanje naše aplikacije u pogon, odabrali smo servis Getjar. Kao i za prethodne servise, i za ovaj je bilo potrebno stvoriti račun. Nakon toga, bilo je potrebno uz APK datoteku priložiti i ime aplikacije, detaljan opis, snimke zaslona i tome slično kako bi aplikacija prošla pregled od strane servisa. Aplikacija je također submitana na Amazon App trgovinu.



Slika 5.15: Status aplikacije na Getjar servisu



Slika 5.16: Status aplikacije na Amazon App trgovini

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj mobilne aplikacije putem koje se mogu održavati online tečajevi koji će obuhvaćati unaprijed pripremljene materijale (videosnimke, skripte i prezentacije) uz mogućnost održavanja konzultacija uživo putem video poziva unutar aplikacije. Nakon 17 tjedana rada u timu i razvoja, ostvarili smo zadani cilj. Provedba projekta tekla je kroz dvije faze.

Prva faza projekta uključivala je okupljanje tima za razvoj aplikacije, dodjelu projektnog zadatka, podjelu odgovornosti i dužnosti, rad na dokumentiranju zah- tjeva te razvoj osnovnih funkcionalnosti aplikacije (kreiranje korisničkog računa, prijava u aplikaciju i slično). Paralelno smo izrađivali aplikaciju i dokumentaciju, što je rezultiralo manjim pritiskom na članove tima približavanjem zadanih ro- kova. Podjela posla uvelike je olakšala rad u grupi te smanjila potrebu za sastan- cima.

Druga faza projekta bila je nešto kraća od prve. Tijekom ove faze projekta, nas- tojali smo realizirati napredne funkcionalnosti aplikacije – kreiranje tečaja, dogo- varanje konzultacija, uspostava video poziva unutar aplikacije te sam upis tečaja. Također, u ovoj fazi potrebno je bilo ispitati naše programsko rješenje i pustiti apli- kaciju u pogon.

Komunikacija među članovima tima odvijala se putem WhatsApp-a i Messen- gera za Facebook, kako bismo postigli informiranost svih članova tima o napretku projekta. Moguće proširenje postojeće inačice aplikacije jest sustav obavijesti, koji bi podsjećao korisnike na dogovorene konzultacije, nove dostupne materijale i slično.

Sudjelovanje na ovakvom projektu bilo je vrijedno iskustvo svim članovima, budući da je većini ovo bio prvi susret s radom u timu na istom projektu. Kroz nekoliko intenzivnih tjedana rada iskusili smo važnost dobre organizacije i ko- munikacije. Zadovoljni smo postignutim rješenjem, iako postoji veliki prostor za napredak aplikacije.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Flutter, <https://flutter.dev/docs>
3. Dart, <https://dart.dev/guides>
4. Astah Community, <http://astah.net/editions/uml-new>
5. Cloud Firebase, <https://firebase.google.com/docs/firestore/data-model>

Indeks slika i dijagrama

2.1	Naslovna stranica Coursere	6
2.2	Stranica online tečaja Sveučilišnog računskog centra	6
2.3	Naslovna stranica platforme edX	7
3.1	Dijagram obrasca uporabe, funkcionalnost korisnika	17
3.2	Dijagram obrasca uporabe, funkcionalnost polaznika tečaja	18
3.3	Dijagram obrasca uporabe, funkcionalnost predavača	19
3.4	Sekvencijski dijagram za UC1 i UC2	21
3.5	Sekvencijski dijagram za UC6 i UC7	22
3.6	Sekvencijski dijagram za UC15 i UC16	24
4.1	Arhitektura sustava	26
4.2	E-R dijagram baze podataka	31
4.3	Dijagram razreda - dio ViewModel	32
4.4	Dijagram stanja - Polaznik	34
4.5	Dijagram aktivnosti	36
4.6	Dijagram komponenti	37
5.1	addCourse	39
5.2	Dodavanje novog korisnika	39
5.3	Provjera prijavljenih korisnika	40
5.4	Prijava u sustav	40
5.5	Prikaz prijavljenog korisnika	40
5.6	Provjera odjave	41
5.7	Rezultati testova	41
5.8	Neispravan unos korisničkih podataka	42
5.9	Ispravan unos korisničkih podataka	42
5.10	Neispravan naziv tečaja	43
5.11	Unos ispravnog naziva tečaja	44
5.12	Uređivanje osobnih podataka	44
5.13	Rezultati integracijskih testova	45

5.14 Dijagram razmještaja	46
5.15 Status aplikacije na Getjar servisu	48
5.16 Status aplikacije na Amazon App trgovini	48
6.1 Prikaz aktivnosti na repozitoriju	57

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 05. listopada 2020.
- Prisustvovali: D.Begić, G.Brkić, M.Dodik, M.Fabijanić, S.Gojević, A.Sabljić, V.Sokolić
- Teme sastanka:
 - sastanak s asistenticom Nikolinom Frid
 - analiza zadatka
 - raščišćavanje dilema funkcionalnosti

2. sastanak

- Datum: 10. listopada 2020.
- Prisustvovali: D. Begić, G. Brkić, M. Dodik, M. Fabijanić, S. Gojević, A. Sabljić, V. Sokolić
- Teme sastanka:
 - odabir alata i tehnologija
 - podjela dužnosti i poslova

3. sastanak

- Datum: 18. listopada 2020.
- Prisustvovali: D. Begić, G. Brkić, M. Dodik, M. Fabijanić, S. Gojević, A. Sabljić, V. Sokolić
- Teme sastanka:
 - definiranje funkcionalnih zahtjeva
 - definiranje obrazaca uporabe

4. sastanak

- Datum: 02. studeni 2020.
- Prisustvovali: D.Begić, G.Brkić, M.Dodik, M.Fabijanić, S.Gojević, A.Sabljić, V.Sokolić
- Teme sastanka:

- sastanak s asistenticom Nikolinom Frid
- raščišćavanje dilema opisa obrazaca uporabe
- demonstracija postignute funkcionalnosti

5. sastanak

- Datum: 09. studeni 2020.
- Prisustvovali: D.Begić, G.Brkić, M.Dodik, M.Fabijanić, S.Gojević, A.Sabljić, V.Sokolić
- Teme sastanka:
 - sastanak s asistenticom Nikolinom Frid
 - evaluacija dosadašnjeg rada
 - raščišćavanje dilema baze podataka

6. sastanak

- Datum: 14. prosinac 2020.
- Prisustvovali: D.Begić, G.Brkić, M.Dodik, M.Fabijanić, S.Gojević, A.Sabljić, V.Sokolić
- Teme sastanka:
 - podjela dužnosti za nastavak rada na aplikaciji
 - provjera do tada ostvarenih funkcionalnosti

7. sastanak

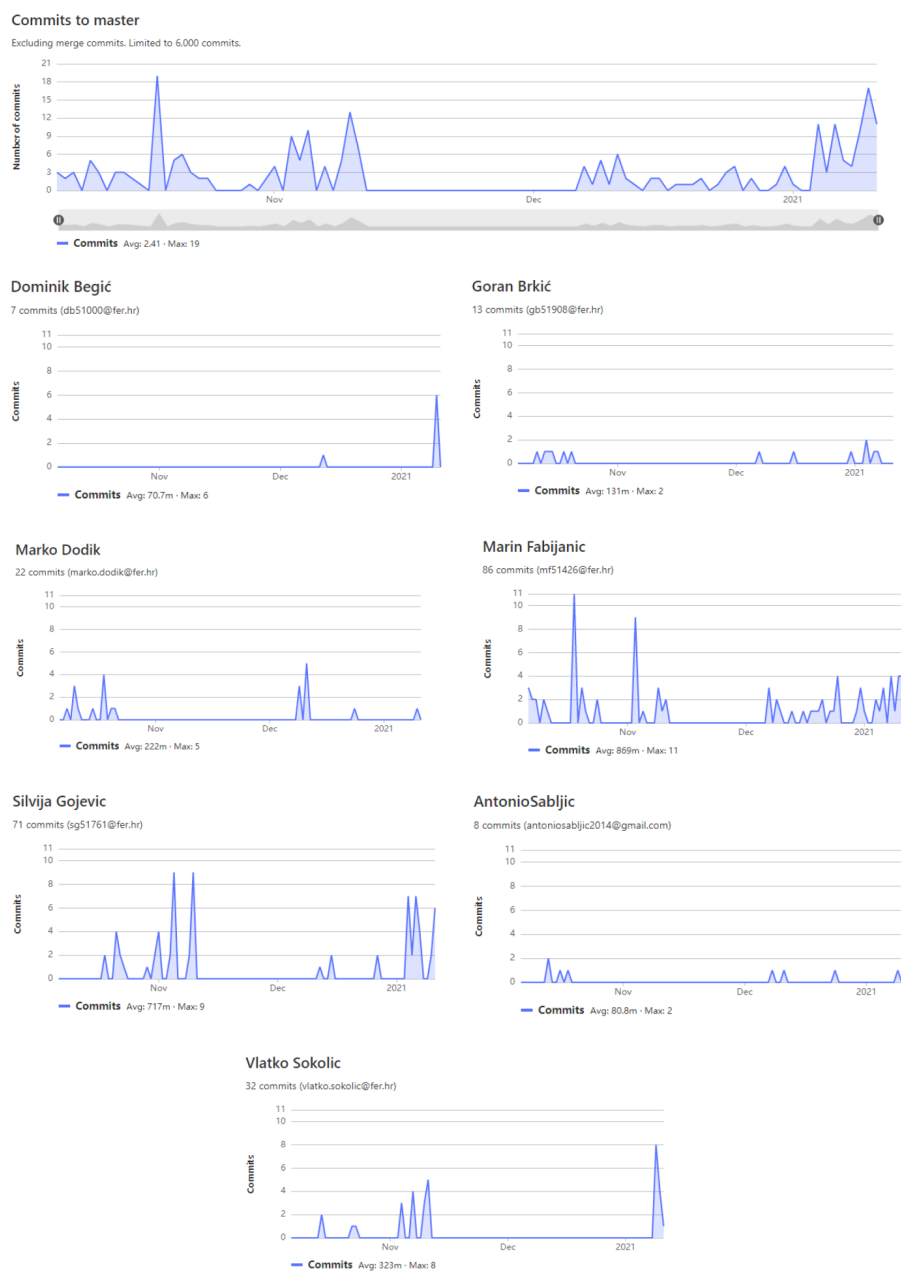
- Datum: 08. siječanj 2021.
- Prisustvovali: D.Begić, G.Brkić, M.Dodik, M.Fabijanić, S.Gojević, A.Sabljić, V.Sokolić
- Teme sastanka:
 - sastanak s asistenticom Nikolinom Frid - demonstracija alfa inačice

Tablica aktivnosti

	Marin Fabijanić	Dominik Begić	Goran Brkić	Marko Dodik	Silvija Gojević	Antonio Sabljic	Vlatko Sokolić
Upravljanje projektom	32	-	-	-	-	-	-
Opis projektnog zadatka	-	-	-	-	5	-	2
Funkcionalni zahtjevi	2	1	1	1	1	1	3
Opis pojedinih obrazaca	4	-	-	-	5	-	
Dijagram obrazaca	-	-	-	-	5	-	-
Sekvencijski dijagrami	1	-	-	-	-	-	6
Opis ostalih zahtjeva	-	-	-	-	1	-	-
Arhitektura i dizajn sustava	10	-	10	7	-	1	5
Baza podataka	18	-	-	-	4	-	-
Dijagram razreda	8	-	-	-	-	-	2
Dijagram stanja	-	-	-	-	-	-	3
Dijagram aktivnosti	-	-	-	-	-	4	-
Dijagram komponenti	2	-	-	-	-	-	2
Korištene tehnologije i alati	-	-	-	-	3	-	-
Ispitivanje programskog rješenja	18	3	6	5	-	7	2
Dijagram razmještaja	2	-	-	-	-	-	2
Upute za puštanje u pogon	2	1	-	-	2	-	-
Dnevnik sastajanja	-	-	-	-	2	-	-
Zaključak i budući rad	-	-	-	-	2	-	-
Popis literature	-	-	-	-	1	-	-
Izrada baze podataka	8	-	-	-	-	-	-
Izrada početne stranice	3	-	10	2	-	4	-
Spajanje s bazom podataka	16	-	-	-	-	-	-
Back-end	65	10	-	-	-	-	-
Front-end	21	-	48	42	-	46	-

	Marin Fabijanić	Dominik Begić	Goran Brkić	Marko Dodik	Silvija Gojević	Antonio Sabljic	Vlatko Sokolić
Priprema za rad (tutoriali, literatura...)	6	15	5	4	6	5	3

Dijagrami pregleda promjena



Slika 6.1: Prikaz aktivnosti na repozitoriju