SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Tim AIR1916**

FOI Knjižnica

**PROJEKT IZ KOLEGIJA „Analiza i razvoj programa“**

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Antonio Kudelić**

**Marin Mačinković**

**Matej Lipovača**

**Matija Ivanić**

**Stiven Drvoderić**

**GitHub repozitorij:** [**https://github.com/AIR-FOI-HR/AIR1916**](https://github.com/AIR-FOI-HR/AIR1916)

ZenHub: <https://app.zenhub.com/workspaces/air1916-5db63d487e2344000155eec5/board?repos=217938608>

FOI Knjižnica – Tehnička dokumentacija

**PROJEKT IZ KOLEGIJA „Analiza i razvoj programa“**

Mentor:

Doc. dr. sc. Boris Tomaš

Varaždin, studeni 2019.

**Sadržaj**

[1. Uvod 1](#_Toc31273973)

[2. Popis funkcionalnosti 2](#_Toc31273974)

[2.1 Karakteristike korisnika 8](#_Toc31273975)

[3. WireFrame 9](#_Toc31273976)

[4. UseCase dijagram 14](#_Toc31273977)

[5. Arhitektura aplikacije 15](#_Toc31273978)

[5.1 FOI CAS server 15](#_Toc31273979)

[5.2 Azure – API 15](#_Toc31273980)

[5.3 Azure – Baza podataka 15](#_Toc31273981)

[5.4 Azure – Function App 16](#_Toc31273982)

[5.5 Azure – Notification Hub 16](#_Toc31273983)

[5.6 Aplikacija 16](#_Toc31273984)

[5.7 Ograničenja aplikacije 16](#_Toc31273985)

[6. Baza podataka 17](#_Toc31273986)

[7. UML dijagram klasa 18](#_Toc31273987)

[8. Xamarin 19](#_Toc31273988)

[8.1 Instaliranje Xamarina 19](#_Toc31273989)

[8.2 Rad u Visual Studio Xamarin 22](#_Toc31273990)

[8.3 Način izrade programskog rješenja u Xamarinu 22](#_Toc31273991)

[8.4 Karakteristika pokretanja aplikacije 23](#_Toc31273992)

[9. BurnDown grafovi 26](#_Toc31273993)

[9.1. Radni satovi za izradu projekta 26](#_Toc31273994)

[9.2. Prvi sprint 26](#_Toc31273995)

1. Uvod

Kroz ovaj uvod ujedno će biti opisana sama domena aplikacije kako bi pobliže objasnili projekt koji smo zamislili. Aplikacija je namijenjena svima onima koji imaju pravo autentifikacije putem AAI@EduHr sustava te bi željeli posuditi knjigu iz FOI knjižnice. Vrsta aplikacije će biti Android te će se sastojati od nekoliko ključnih funkcionalnosti koje bi olakšale svakodnevno rezerviranje i posudbe unutar knjižnice. Valja naglasiti da je ovo neovisan projekt te nemamo pristup stvarnim podacima iz FOI knjižnice te će podaci unutar aplikacije biti testne naravi. Preko korisničkog sučelja bila bi omogućena inicijalna prijava putem AAI@EduHr sustava nakon čega bi se korisnicima ponudila opcija da se ubuduće prijavljuju koristeći pin, uzorak ili otisak prsta. Nakon prijave korisnici bi bili u mogućnosti pregledavati katalog knjižnice (sortiranje, filtriranje, pretraga i sl.) te prema želji rezervirati knjigu ako je ona slobodna. Rezervacija bi se slala putem e-maila knjižničarima te bi ista vrijedila određeni vremenski period. Ukoliko bi publikacija bila podignuta u tom periodu novo stanje dostupnosti bi bilo vidljivo svim korisnicima na korisničkom sučelju, u suprotnom rezervacija bi bila otkazana. Ukoliko bi korisnici prikazali određeni interes za neku publikaciju te ju stavili u vlastite favorite, dobili bi notifikaciju o raspoloživosti iste kako bi ju mogli rezervirati i posuditi. Razlog radi kojeg bi izradili ovakvu vrstu aplikacije je upravo taj što ovakva mobilna aplikacija još ne postoji za FOI knjižnicu (osim web stranice koja ne posjeduje sve mogućnosti koje bi bile implementirane unutar ove aplikacije) te smatramo da bi uvelike mogla olakšati proces posudbe knjižničarima, ali i korisnicima.

1. Popis funkcionalnosti

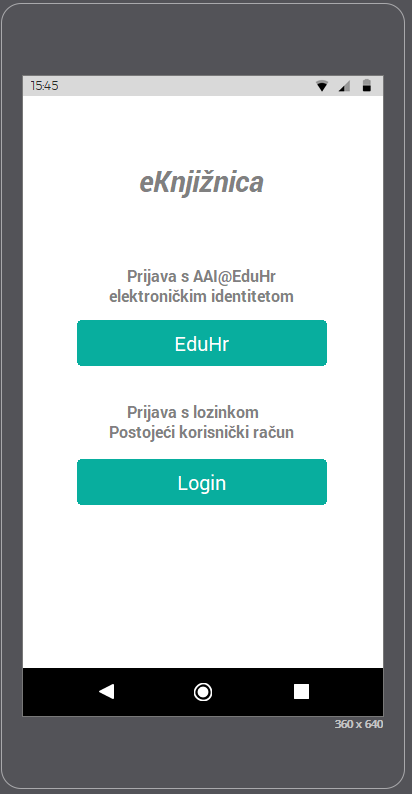
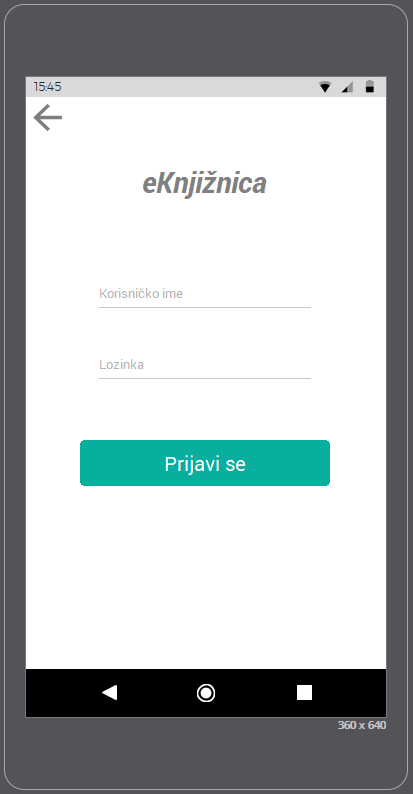
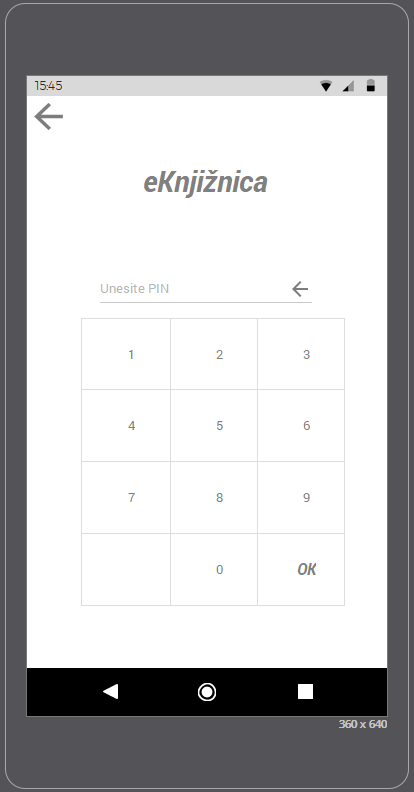
|  |  |  |  |
| --- | --- | --- | --- |
| **Broj funkc.** | **Naziv**  **funkcionalnosti** | **Opis funkcionalnosti** | **Slika ekrana** |
| 1. | Prijava (modularno) | Mogućnost prijave korisnika u sustav sa svojim podacima za prijavu. Prijava na svoj račun putem tuđeg mobilnog uređaja/uređaja na kojim nisu napravljeni brzi načini prijave bit će moguća samo uz korisničko ime i lozinku, a prijava brzim načinom samo na svojem mobilnom uređaju/uređaju na kojem su napravljeni brzi načini prijave. |  |
| 1.1 | Prijava (korisničko ime i lozinka) | Prijava putem korisničkog imena i lozinke je osnovna vrsta prijave koju korisnik uvijek može koristiti u slučaju da ostale vrste ne rade ili ako korisnik to preferira. Kada se korisnik prvi puta prijavi sa svojim [AAI@edu.hr](mailto:AAI@edu.hr) podacima u našu bazu će se povući njegovi podaci i kreirati njegov korisnički profil u našoj bazi te će se korisniku u budućnosti omogućiti prijava sa tim podacima unutar same aplikacije. |  |
| 1.2 | Prijava (PIN)(modularno) | Prijava putem PIN koda bit će jedna od opcija za bržu prijavu u sustav. Kako bi korisnik postavio PIN za prijavu mora se najprije prijaviti u aplikaciju sa svojim korisničkim imenom i lozinkom te jedanput kada je prijavljen može unutar svojeg profila aktivirati mogućnost prijave putem PIN-a . |  |
| 1.3 | Prijava (Otisak prsta)(modularno) | Prijava putem otiska prsta bit će opcija brze prijave za korisnike koji imaju čitač otiska prsta na svojem mobilnom uređaju. Kada se korisnik prijavi u sustav može u unutar svojeg profila aktivirati mogućnost korištenja otiska prsta te će mu se u budućnosti nuditi ta opcija za prijavu. |  |
| 1.4 | Prijava (Uzorak)(modularno) | Prijava putem uzorka je opcija brze prijave korisnika u sustav na način da u polju točaka 3x3 napravi svoj uzorak i ako je on točan korisnik se prijavljuje u sustav. Korisnik svoj uzorak aktivira u svojem računu i specificira izgled uzorka kojeg će koristiti. |  |
| 2. | Pregled literature | Osnovni pregled literature u obliku popisa sa slikom naslovnice knjige, informacijama o autorima, godini izdanja, izdavačkoj kući te mogućnošću otvaranja ekrana sa detaljima o knjizi. |  |
| 2.1 | Filtriranje literature | Mogućnost filtriranja literature prema prvom slovu naslova, autoru, razdoblju izdavanja, izdavaču itd. Za opcije filtriranja otvara se novi ekran gdje se postavljaju željeni parametri te se nakon toga otvara opet pregled literature. |  |
| 2.2 | Sortiranje literature | Mogućnost sortiranja literature po abecednom redu, godini izdavanja, autoru itd. |  |
| 2.3 | Pretraga | Mogućnost pretrage baze podataka literature prema unesenoj ključnoj riječi bio to dio naziva knjige, autoru, izdavačkoj kući itd. |  |
| 2.4 | Detaljne informacije | Poseban ekran koji se otvara pritiskom na željenu literaturu kod pregleda literature. Pregled detaljnih informacija sastoji se od kratkog opisa sadržaja knjige, osnovnih informacija o knjizi ( autori, broj stranica, godina izdavanja…), informacija o dostupnosti pojedine instance knjige itd. |  |
| 2.5 | Pregled sadržaja | Pritiskom na gumb korisniku se otvara novi ekran koji sadrži listu sadržaja kako bi se korisniku bolje predočila knjiga. |  |
| 2.6 | Rezerviranje knjige | Aplikacije daje korisniku mogućnost rezerviranja dostupne knjige pritiskom na gumb te se stanje knjige mijenja u rezervirano, nakon čega korisnik podiže knjigu u knjižnici u zadanome roku te se time mijenja stanje knjige u zauzeto. |  |
| 3. | Korisnički račun | Ekran korisničkog računa sadrži informacije o trenutno posuđenim, rezerviranim i prethodno posuđenim knjigama. Također daje mogućnost otvaranja detaljnih informacija. |  |
| 3.1 | Prekidanje rezervacije | Mogućnost prekidanja trenutno rezervirane knjige. |  |
| 3.2 | Favoriti | Mogućnost da korisnik doda neku knjigu kao favorita te primi notifikacije o dostupnosti knjige |  |
| 4. | Postavke/Notifikacije | Ekran koji nudi mogućnost odabira notifikacija koje korisnik želi primati. |  |
| 4.1 | Podsjetnik o slobodnoj knjizi | Notifikacija koja korisniku daje obavijest o tome da se oslobodila neka knjiga koju je on odredio da ga zanima. |  |
| 4.2 | Notifikacija o isteku rezervacije ili roka vraćanja knjige | Notifikacija koja daje korisniku informaciju da će za neko vrijeme isteći njegova rezervacija ili da će isteći rok posudbe. |  |
| 5. | Pomoć u sustavu | Pomoć korisniku pritiskom na upitnik u gornjem kutu ekrana. |  |
| 6. | Slanje poruke | Slanje poruke na e-mail adresu knjižnice. |  |
| 7. | Osnovne informacije o knjižnici | Ekran sa radnim vremenom te kontakt informacijama knjižnice. |  |

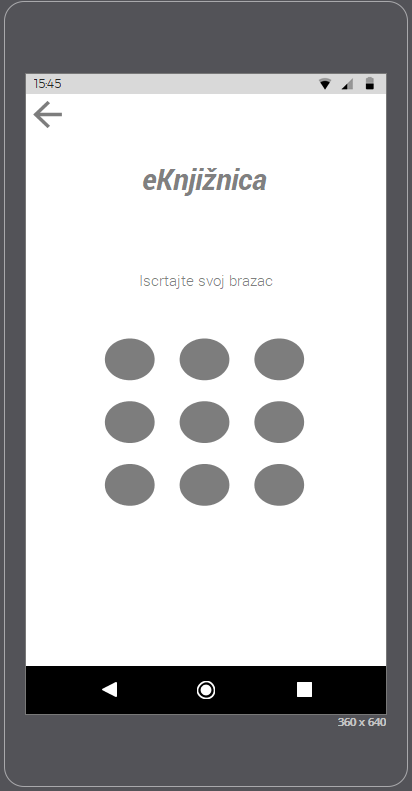
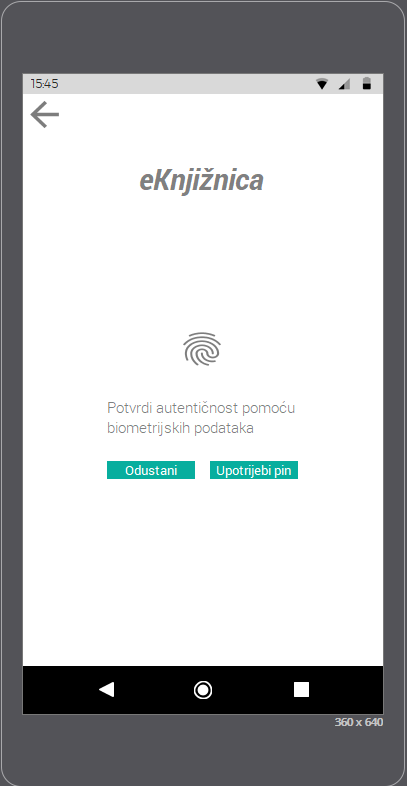
* 1. Karakteristike korisnika

Aplikacija je namijenjena svima koji imaju pristup FOI autentikaciji i željeli bi rezervirati/posuditi neku publikaciju iz knjižnice. Bitno je naglasiti da je korisnik samo onaj koji posuđuje, ali ne i knjižničar/knjižničarka.

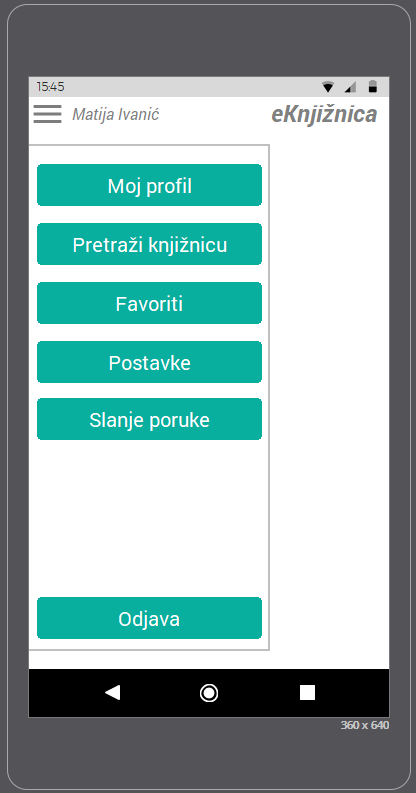
1. WireFrame

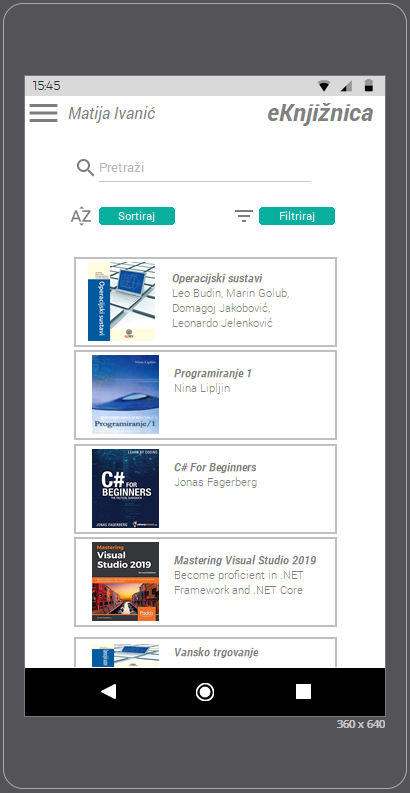
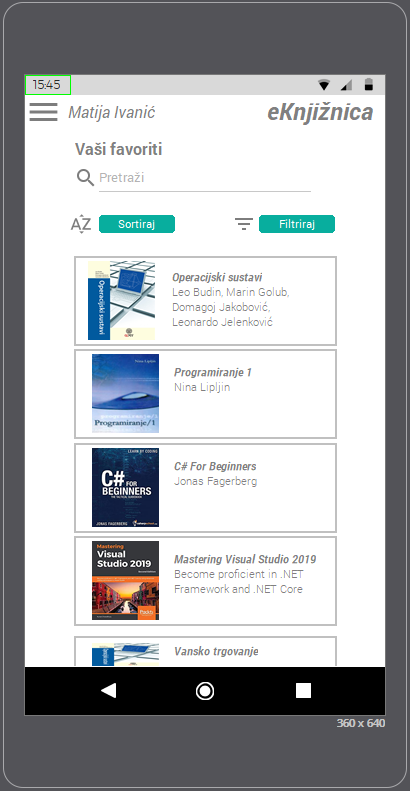
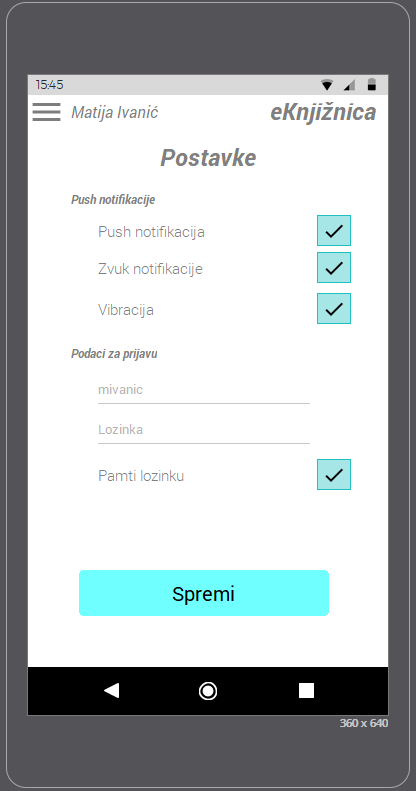
Početni prikaz omogućava nam da odaberemo mogućnost prijave. Ukoliko se prijavljujemo prvi puta, onda ćemo za odabir odabrati gumb „EduHr“ s kojim ćemo se prijaviti na eduHr, te potvrditi naš status kao studenta FOI, te nam omogućiti prijavu u aplikaciju. Ukoliko smo se već prijavili, te je aplikacija spremila našu prijavu i naš korisnički račun, onda za lakšu i bržu prijavu možemo odabrati drugi gumb, „Login“. S ovim gumbom odlazimo do ekrana s kojim se prijavljujemo u aplikaciju. Ovisno o našem izboru, to može biti preko korisničkog imena i lozinke, otiska prsta, pin zaporke ili crtanje uzorka, odnosno obrazac.

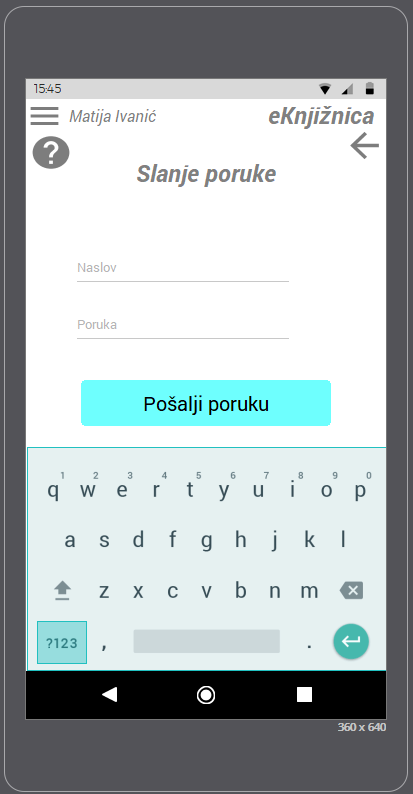
  

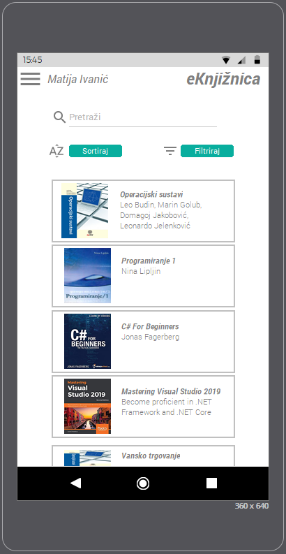
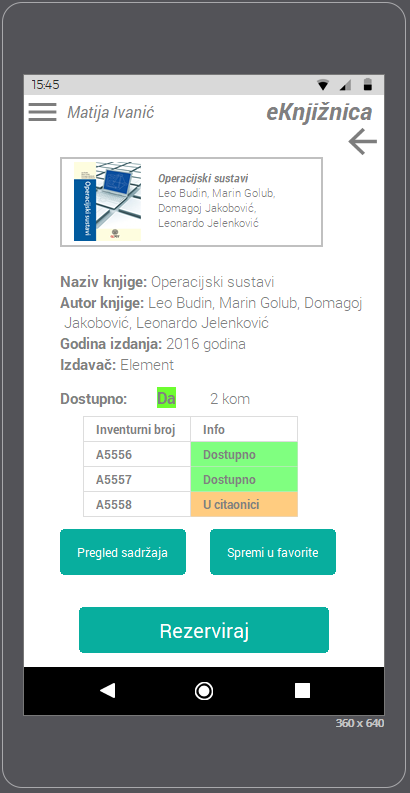
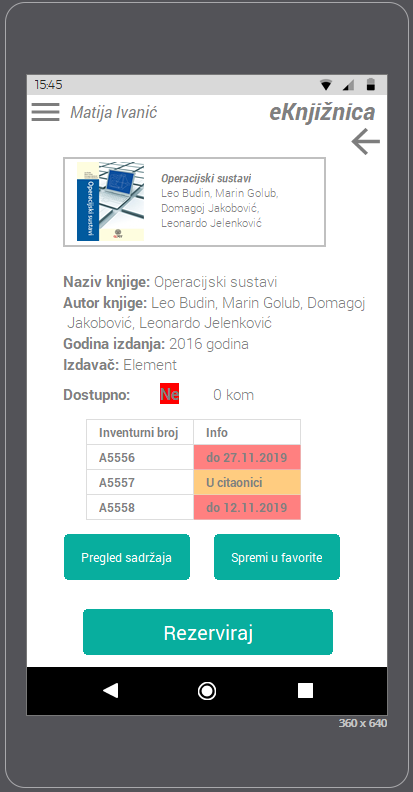
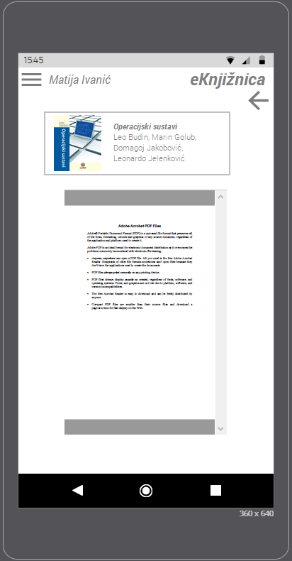
Ukoliko smo se uspješno prijavili, otvara nam se početni zaslon, odnosno naš korisnički račun, u kojem možemo vidjeti detaljnije informacije o našim posuđenim knjigama. Isto tako u listi prijavljenih rezervacija, možemo otkazati našu rezervaciju, odabirom gumba „Otkaži rezervaciju“. U gornjem lijevu kutu nalazi se ikonica izbornika. Pritiskom na ikonicu otvara nam se izbornik. Odabirom određenog gumba, odlazimo do sljedećeg ekrana.



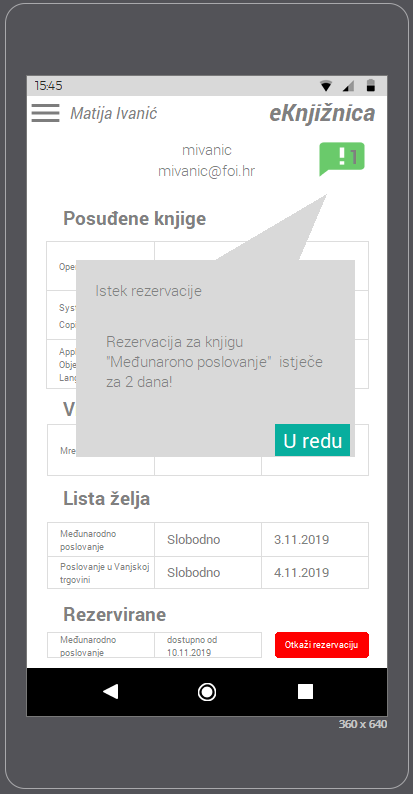
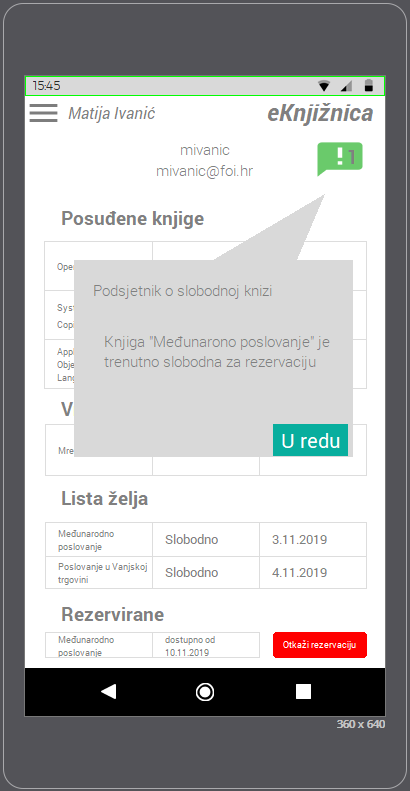
  

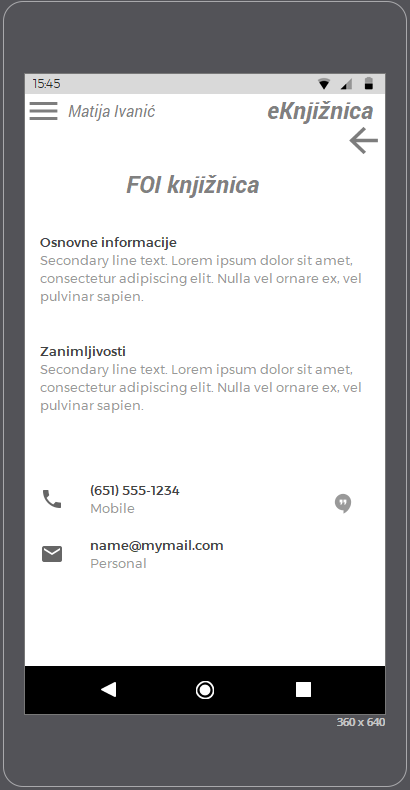
U ekranu pretraživanje, možemo prelistati cijeli katalog knjižnice, ali isto tako unosom određenog pojma možemo pretražiti, sortirati, ali i filtrirati. Klikom na odabranu knjigu, otvara nam se detaljni prikaz knjige, gdje možemo vidjeti sve pojedinosti o toj knjizi, te odabirom gumba „Prikaz sadržaja“, možemo vidjeti nekoliko stranica iz knjige. Ovdje imamo mogućnost rezervacije knjige ukoliko je slobodna, te spremiti u favorite. Pomoć u sustavu osmišljena je na način da nam se u svakom prozoru pojavi znak „?“, upitnika. Klikom na upitnik otvara nam se prozor sa pomoć u sustavu za određeni problem. Tako da korisniku olakšamo snalaženje, te mu omogućimo jednostavnost. Primjer ovoga možemo vidjeti na slici prozora „Slanje poruke“, gdje imamo upitnik, te klikom na njega otvaramo pomoć. Za vraćanje nazad, samo pritisnemo znak strelice unatrag.

Ukoliko dobijemo notifikaciju, nju možemo vidjeti u notifikacijama, ali isto tako možemo vidjeti novu nepročitanu notifikaciju na našem profilu klikom na ikonicu notifikacije

Ukoliko želimo vidjeti osnovne informacije o knjižnici, to možemo vidjeti klikom na logo „eKnjižnica“, koja nam otvara poseban ekran sa svim detaljnim informacijama o FOI knjižnici, te kontakt. To možemo učiniti na bilo kojem ekranu, gdje je prikazan logo „eKnjižnica“.

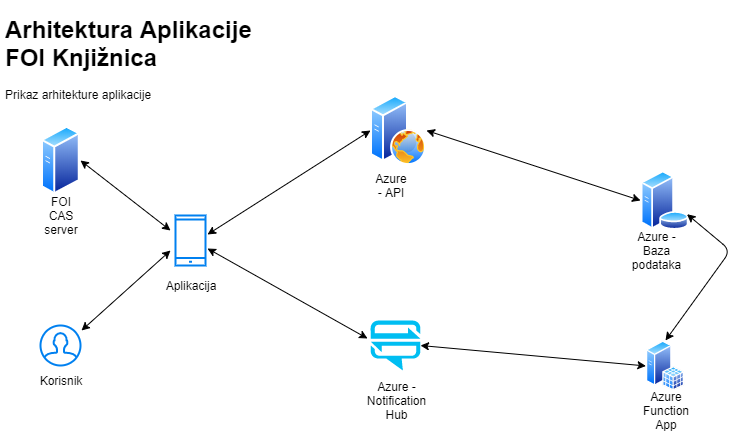
1. UseCase dijagram

Slika na kojoj se prikazuje tekst, karta

Opis je automatski generiran

Na slici gore prikazan je UseCase dijagram za aplikaciju Foi knjižnica na kojoj su prikazani svi mogući slučajevi korištenja aplikacije . Potrebno je napomenuti da je slučaj prijave u aplikaciju potreban za izvađanje ostalih slučajeva, ali nije prikazano na slici zbog preglednosti.

1. Arhitektura aplikacije

Na slici ispod vidljiva je arhitektura „FOI Knjižnica“ aplikacije. Vidljivo je da se aplikacija temelji na funkcionalnosti nekoliko različitih funkcionalnih cjelina koji će biti objašnjeni u nastavku.

* 1. FOI CAS server

Riječ je o FOI serveru koji nam omogućuje našu inicijalnu prijavu u aplikaciju. Temelji se na svojoj vezi sa *AAI@EduHr* sustavom koji predstavlja autentikacijsku i autorizacijsku infrastrukturu znanosti i visokog obrazovanja u Republici Hrvatskoj. Nakon uspješne prijave našeg korisnika u sustav, odnosno inicijalne registracije, aplikacija od FOI CAS servera, koristeći **DotNetCasClient** dobije osnovni podatak o prijavljenom korisniku koji se potom zapisuje u bazu podataka na našoj strani.

* 1. Azure – API

Sva komunikacija između baze podataka i aplikacije, koja je vezana uz dohvaćanje i ažuriranje zapisa (u našem slučaju – Publikacija) obavlja se preko Web API-a implementiranog u ASP.NET-u koristeći „*Api Controller*“ klasu.

* 1. Azure – Baza podataka

Naša baza podataka na Azure servisu koja nam služi za pohranu svih potrebnih podataka.

* 1. Azure – Function App

Azure Function App u projektu nam služi, zajedno s *Notification Hubom*, za slanje obavijesti korisniku o isteku rezervacije/posudbe određene publikacije. Također, koristi se još i za automatsko ukidanje isteklih rezervacija.

* 1. Azure – Notification Hub

Azure Notification Hub nam omogućuje slanje obavijesti na mobitel korisnika, a temelji se na izvršenju *time trigger-a* sa *Function App-a*.

* 1. Aplikacija

Sama aplikacija se sastoji od nekoliko projekata: **FOIKnjiznica**, **FOIKnjiznica.Android**, **FOIKnjiznicaWebServis** i modula za prijavu.

* *FOIKnjiznica* – Sadrži sve ekrane i potrebne klase za rad aplikacije
* *FOIKnjiznica.Android* – Sadrži *activitye* koji se pokreću prilikom pokretanja aplikacije
* *FOIKnjiznicaWebServis* – Sadrži kontrolere preko kojih komuniciramo s bazom podataka te nam omogućuje prijavu koristeći *FOI login*.
  1. Ograničenja aplikacije

Aplikacije je ograničena na korisnike mobilnih uređaja s Android operativnim sustavom koji moraju imati nekakav oblik konekcije na Internet kako bi pristupili našoj aplikaciji – potrebno se spojiti na bazu podataka, tj. WebAPI-e. Također, kako je navedeno i u potpoglavlju 2.1., aplikacija je namijenjena isključivo osobama s pristupom FOI autentikaciji.

1. Baza podataka

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

Na slici iznad nalazi se osnovni ERA model naše aplikacije. U nastavku će biti opisane uloge svake od tablica kako bi model bio jasniji.

1. **Publikacije** – Tablica koja se odnosi na sve publikacije koje se u knjižnici nalaze te sadrži sve osnovne informacije o njima.
2. **Kategorije** – Kategorije predstavljaju vrstu publikacije jer publikacija može biti knjiga ili časopis npr.
3. **Autori** – Tablica sadrži imena svih autora
4. **Je\_Autor** – Tablica koja sadrži podatke o tome tko je autor koje publikacije (međutablica)
5. **Izdavači** – Tablica sadrži popis izdavača
6. **Članovi** - Tablica koja sadrži osnovne podatke o članovima koji imaju pristup posuđivanju knjiga. Tablica je napravljena na temelju imeničke sheme AAI@EduHr sustava koja se može pronaći na: <https://www.aaiedu.hr/o-sustavu/imenicke-sheme/shema?fbclid=IwAR01PQ5iwTHi1ALza5MqguzVi1hE9_t5dOCQUkLBNZHGQBiZbBBMoskmwWo>. Riječ je o osnovnim podacima koji bi se pohranili prilikom inicijalne prijave u sustav koje ćemo onda koristiti kako bi omogućili nesmetan rad korisnika u aplikaciji.
7. **Je\_Favorit** – Omogućuje članovima odabir publikacija koje favorizira
8. **Kopija\_Publikacije** – Tablica koja služi za evidenciju svih kopija neke publikacije jer svaka ima određeni broj kopija koji se nalazi u knjižnici.
9. **Stanje\_Publikacije** – Tablica koja nam služi za dohvaćanje podatka o određenoj kopiji publikacije. Odnosno, omogućuje nam evidenciju rezervacija, posudbi, vraćenih publikacija i sl. Također, stupac te tablice „datum\_do“ bi se popunjavao isključivo prilikom posudbe publikacije kako bi se mogla voditi evidencija do kad je neka publikacija posuđena.
10. **Vrsta\_Statusa** – Tablica koja služi za pohranu mogućih statusa neke publikacije (slobodna, rezervirana, posuđena).
11. **Auth\_Protocol** – Tablica koja je vezana uz modularni dio aplikacije te koristi za evidenciju mogućih načina prijave korisnika u aplikaciju (pin, otisak prsta, uzorak).
12. **Članovi\_Auth\_Protocol** – Međutablica između Auth\_Protocol i Članovi tablica putem koje vodimo evidenciju o tome što je pojedini član odabrao za način prijave u sustav.
13. UML dijagram klasa

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

Na slici iznad može se vidjeti UML dijagram klasa za aplikaciju FOI Knjižnica. Slika predstavlja trenutni izgled dijagrama klasa te će se dijagram mijenjati kroz vrijeme dodavanjem i mijenjanjem klasa u samoj aplikaciji. Na trenutnome dijagramu mogu se vidjeti klase potrebne za prvi sprint, a to su klase vezane uz same publikacije. Klase vezane uz publikacije su Kategorija, Autor, Izdavac, Publikacija, KopijaPublikacije, StanjePublikacije te vrsta statusa. Te klase poslužit će za izradu glavnog ekrana na kojem će se prikazivati rezultati nekog filtriranja, pretrage i sortiranja. Osim tih klasa za prvi sprint također je prikazana klasa BazaPodataka koja će služiti za povezivanje aplikacije sa bazom podataka. Klasa će sadržavati neke preddefinirane SQL upite kao funkcije klase te će korisniku omogućiti i unos vlastitog SQL upita. Zadnja stvar koja je prikazana na UML dijagramu je modularni dio aplikacije, a to je sučelje kojeg implementiraju prethodno navedeni moduli.

1. Xamarin

**Xamarin** se integrira s Visual Studio, Microsoftovim IDE-om za .NET Framework, a kasnije je dostupan za upotrebu macOS korisnicima putem Visual Studio za Mac. Koristi C# kod, te razne alate i biblioteke za kreiranje aplikacija na platformama kao što su iOS, Android, macOS, Windows apps, te za razne druge.

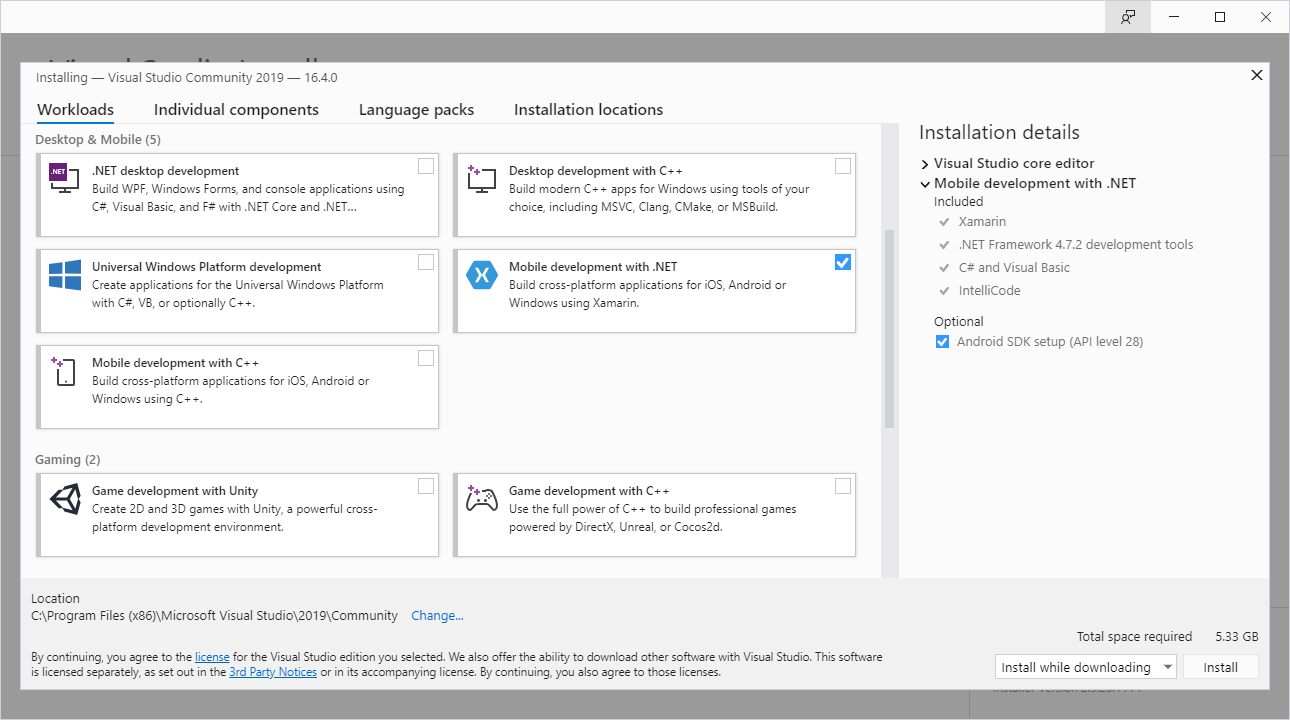
Microsoft **Visual Studio** je integrirano razvojno okruženje (IDE) koje je razvio Microsoft. Koristi se za razvoj računalnih programa, kao što su web-stranice, web aplikacije, servisi, te mobilne aplikacije.

Pošto je Xamarin dio Visual Studio paketa, zasniva se na .NET platformi. .NET je razvojna platforma napravljena od raznih alata, programskog jezika i biblioteka za kreiranje raznih različitih vrsta aplikacija. Baza platforme pruža komponente svim različitim vrstama aplikacija. .NET je besplatna cross-platform platforma, otvorenog koda. Podržava različite programske jezike, editore i biblioteke za kreiranje web, mobilnih, desktop, gaming, i IoT aplikacija.

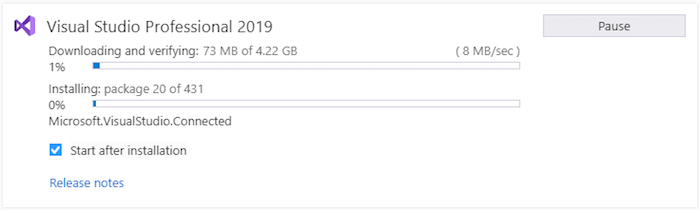
* 1. Instaliranje Xamarina

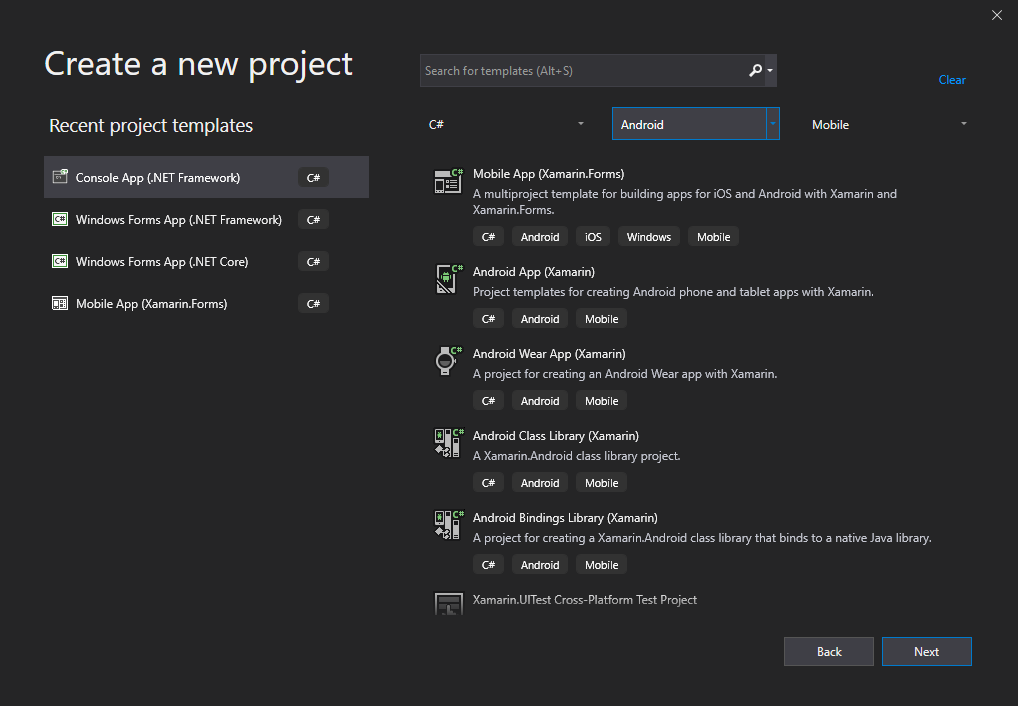
Prema uputama na moodle-u u nastavku će biti opisana instalacija i neke od osnova rada u Xamarinu. Pošto smo mi koristili Xamarin za Windows, slijedi opis instalacije istoga na računalu s operacijskim sustavom Windows.

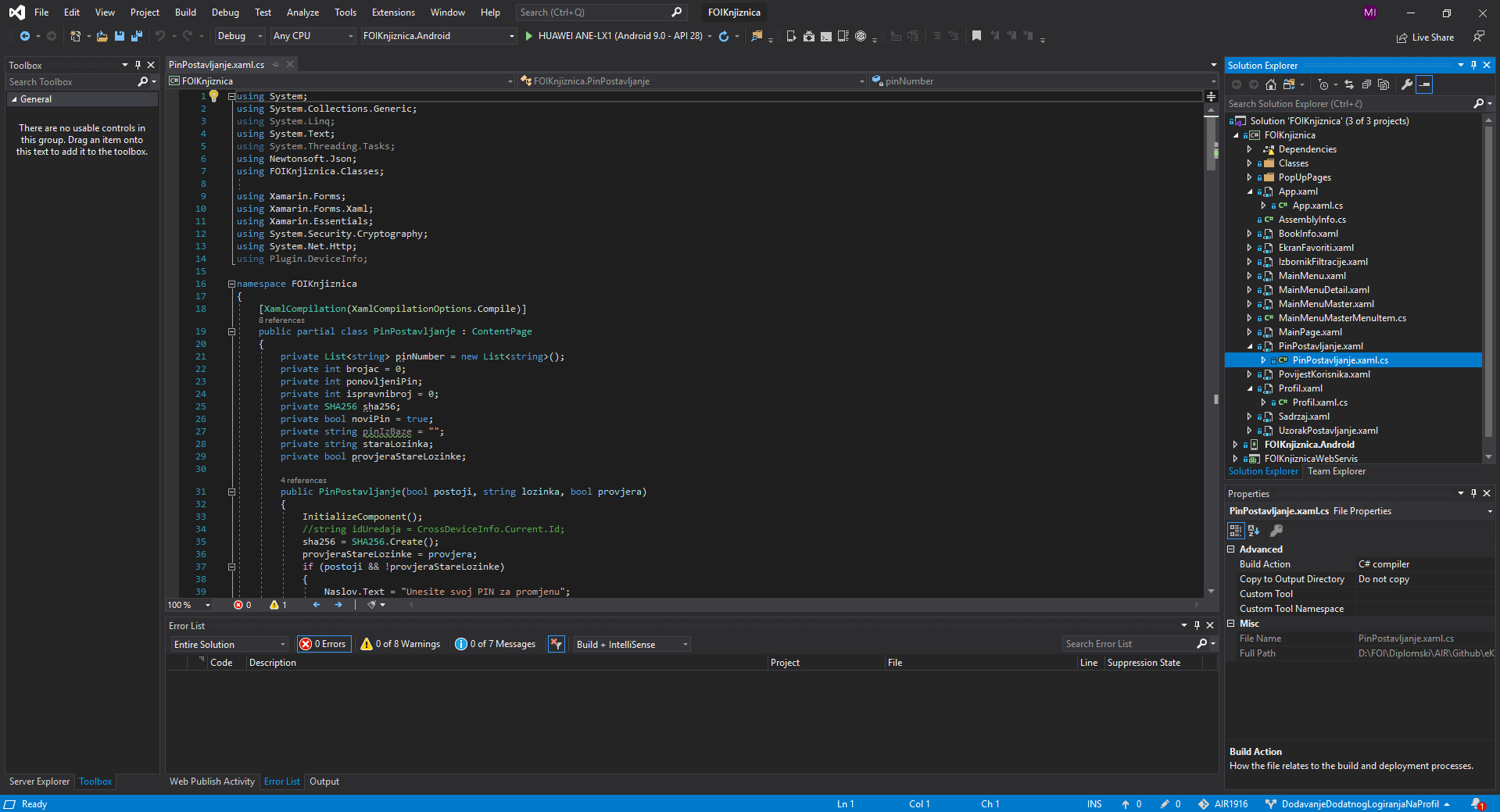
Prvo što trebamo je skinuti Visual Studio, najbolje zadnju verziju (2019). Kako mi koristimo besplatnu verziju, onda ćemo svakako odabrati Community verziju. Verzija koju mi koristimo je Visual Studio 2019 Community.

Posjetimo VisualStudio stranicu , te odaberemo navedenu verziju, te nakon završenog skidanja, dvaput kliknemo na start instalacija. Otvara nam se ekran instalacije (slika) , gdje odabiremo željene instalacijske pakete. Paket koji mi trebamo je „Mobile development with .NET“.

Nakon što smo odabrali, kliknemo na „Install“ u donjem desnom kutu. Za ovu instalaciju potrebno je oko 19 GB slobodno prostora na disku.



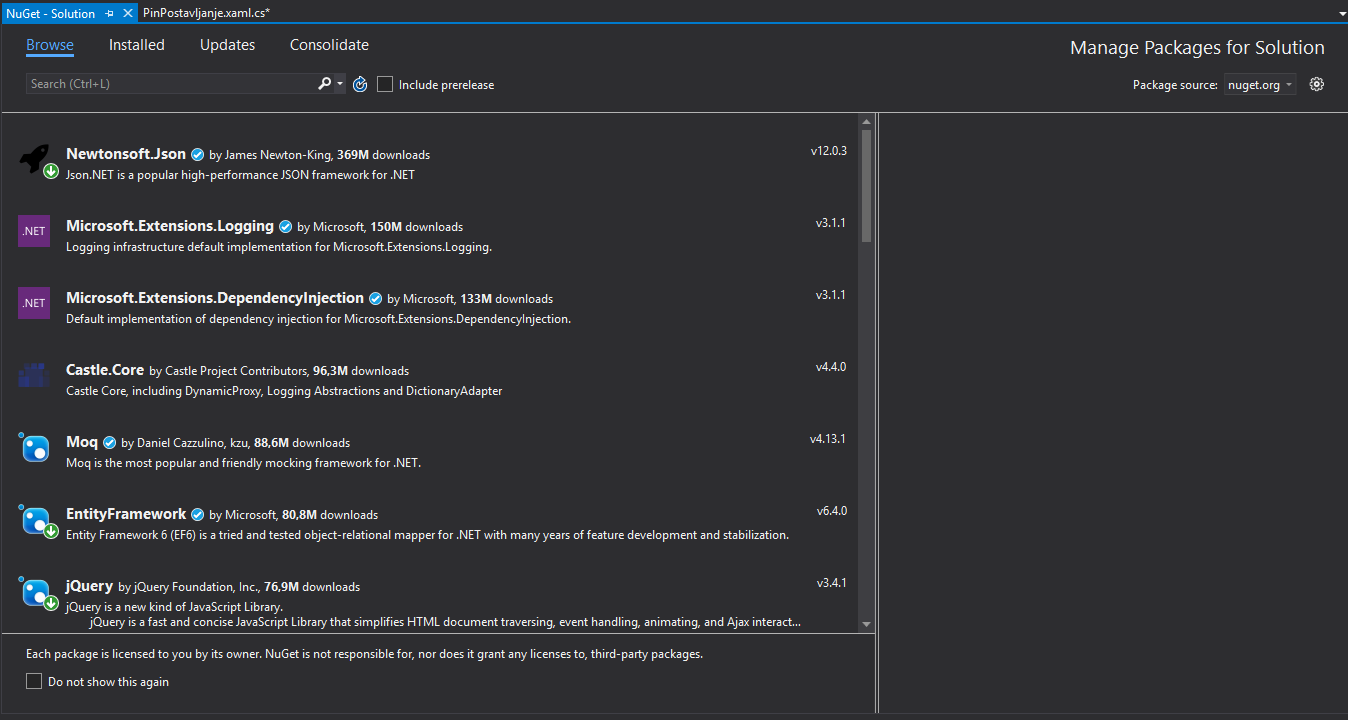
Nakon što je instalacija gotova, kliknemo na „*Launch button*“ kako bi pokrenuli Visual studio. Nakon što smo pokrenuli Visual studio možemo odabrati kreiranje novog projekta za mobline aplikacije (slika ispod).

Odabirom na projekt, u našem slučaju to je **Android App (Xamarin)**, dolazimo do sučelja (slika ispod) na kojem radimo dalje naš projekt.

* 1. Rad u Visual Studio Xamarin

Vrlo važna stavka u radu unutar Xamarin-a su svakako „Nuget „ paketi. Nuget paketi su moduli, odnosno gotova rješenja za neku vrstu funkcionalnosti. Time se postiže jednostavnost, te se značajno ubrzava rješenje, odnosno izrada aplikacije.

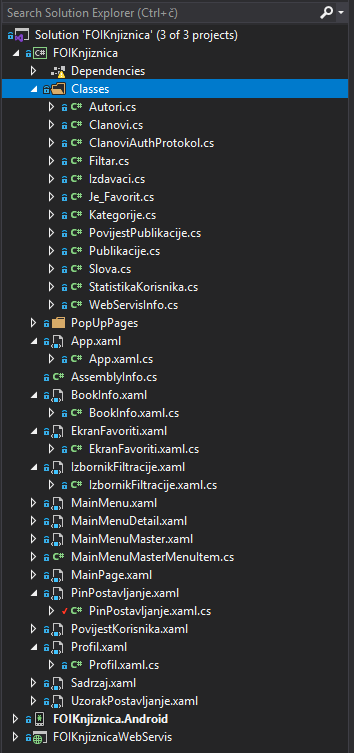
Kako bismo dodali Nuget paket, potrebno je otići *Tools -> Nuget packete Manager -> Manage Nuget Packets for solutions*. Nakon odabira, otvara nam se Nuget prozor (slika). Nuget browser se sastoji od 4 izbornika. *Browse* u kojem tražimo, odnosno odabiremo paket kojeg želimo uključiti u naše rješenje. Nakon odabira, potrebno je odabrati u koje projekte našeg rješenja želimo uključiti paket. Neki paketi trebaju se uključiti u cijelo rješenje, dok drugi u samo određene projekte. I to je vrlo važno da odaberemo pravi projekt, kako bi paket mogao normalno raditi.



Sljedeći izbornik je *Installed*, gdje možemo vidjeti sve Nuget pakete koje smo instalirali u naše rješenje, te možemo vidjeti točno u koji dio projekta smo ga uključili. Treći dio izbornika je *Updates*, koji nam javlja potrebna ažuriranja. Zadnji izbornik je *Consolidate*.

* 1. Način izrade programskog rješenja u Xamarinu

Rad u Xamarinu je veoma sličan kao i radi u ''običnom'' Visual Studio-u. Solution explorer sastoji je od projekata. U našem slučaju to je *FOIKnjiznica*, *FOIKnjiznica.Android* i *FOIKnjiznicaWebServis*. FOIKnjižnica.Android je projekt koji služi kao komunikaciju s android uređajima, te služi za komunikaciju androida i našeg projekta. FOIKnjiznica služi za izgradnju našeg projekta. U njemu se nalaze sve potrebne klase i stranice koje koristimo u aplikaciji. Stranice u xamarinu su *''Content Page''* koje su oblika XAML koda s c# kodom u pozadini (slika). Treći dio našeg rješenja je projekt koji se zove FOIKnjiznicaWebServis te služi za komunikaciju sa bazom podataka, točnije, služi nam kao WebAPI putem kojeg se ostvaruje komunikacija aplikacije i baze.



* 1. Karakteristika pokretanja aplikacije

Trenutačno, aplikacija radi na **localhostu** jer nemamo potpuni pristup CAS serveru jer je aplikacija, realno, i dalje u razvoju. Točnije, naša domena koja se koristi nije uvrštena u pristup CAS serveru. Stoga, u Visual Studiu koristimo jedan dodatan alat **Conveyor by Keyoti** koji nam omogućuje sigurnu vezu sa FOI CAS serverom sa mobitela preko localhosta (pošto je https konekcija).

Kako bi ga instalirali potrebno je pratiti korake navedene na <https://marketplace.visualstudio.com/items?itemName=vs-publisher-1448185.ConveyorbyKeyoti>.

Nakon instalacije potrebno ga je uključiti tako da odemo u *Tools* i samo kliknemo na njega. Tada je potrebno pokrenuti naš projekt FOIKnjiznicaWebServis u IIS exploreru te ćemo dobiti prozor u kojem je prikazan „**Remote URL**“ putem kojeg možemo pristupiti localhostu. Naravno, kako bi uspjeli pristupiti istom sa mobilnog uređaja biti će potrebno instalirati conveyor certifikat, a postupku te instalacije pristupamo kada kliknemo na ikonicu pokraj Remote URL-a.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

Nakon uspješne instalacije certifikata na mobilni uređaj, potrebno je izvršiti malu konfiguraciju *web.configa* kako bi aplikacija mogla raditi na localhostu nekog računala, a koraci su sljedeći:

1. Otvoriti *command prompt* i izvršiti naredbu ipconfig te pronaći našu IPv4 adresu.
2. U *Conveyor by Keyoti* prozoru postaviti u opcijama našu IPv4 adresu (slika gore, donji lijevi kut).
3. Pokrenuti naš FOIKnjiznicaWebServis projekt te pogledati ***Remote URL*** koji dobijemo (Port je jako bitan!)
4. Sada u web.config je potrebno promijeniti atribut **serverName** te obavezno staviti i broj porta.
5. Kako bi se aplikacija uspješno izvršila na *webviewu* unutar aplikacije potrebno je još u PokreniAplikacijuActivity.cs promijeniti **webView.LoadUrl()** u istu vrijednost kao i što je *serverName*.

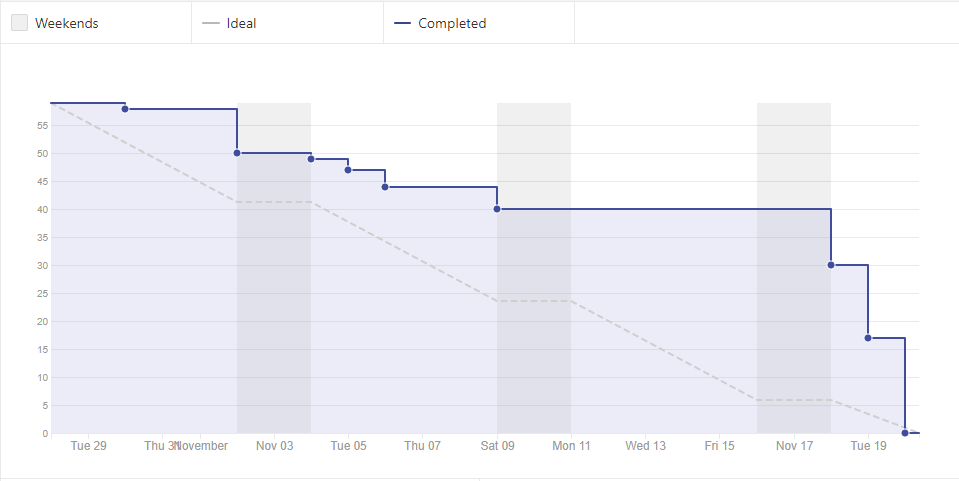
9. BurnDown grafovi

* 1. Radni satovi za izradu projekta

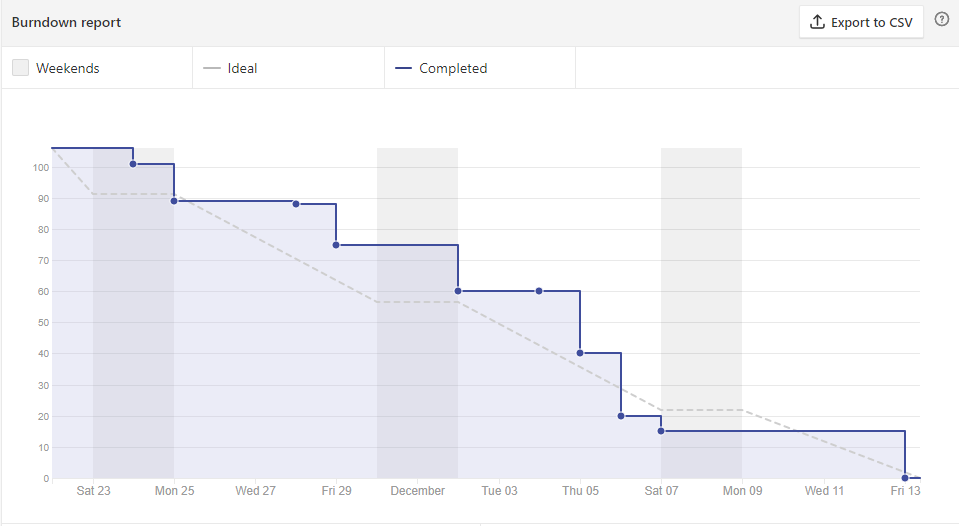
Da bismo mogli dobiti prosječan broj sati koji će nama biti potreban za uložiti, da bismo izradili aplikaciju, potrebno je odrediti ukupni potreban broj sati za kolegij. Kako ovaj kolegij nosi 6 ECTS bodova, to znači da student treba odraditi 180 sati rada. Ako stavimo da prosječan broj tjedana iznosi 15, dobijemo da student treba raditi 12 sati tjedno. Tjedno kolegij sadržava 3 sata predavanja, te još tri sata labos vježbi. Ukupno 6 sati tjedno. Kako se nastava izvodila 10 tjedana, dobijemo da student potroši 60 sati za potrebe nastave. Kako je 60 sati potrebno raspodijeliti na 15 tjedana, dobijemo da student prosječno „potroši“ 4 sata za potrebe nastave. Stoga za na aplikaciji ostalo 8 sati tjedno. Ako 8 sati podijelimo na 7 dana, jer smo radili i kroz vikende, stoga dolazimo da dnevno možemo raditi prosječno sat vremena. Kako nas je 5 u timu, dolazimo do toga da dnevno možemo raditi 5 sati. Prosječno nam sprint traje 21 dan. Stoga smo za sprintove uzeli 105 sati rada.

Ovo je subjektivan broj sati, neki prosjek koji smo uzeli da bismo odredili trajanje sprintova, te želimo napomenuti kako stvarni broj sati koji smo odradili u svakom sprintu je puno već, jer nismo uzimali vrijeme koje nam je bilo potrebno da uklonimo određene probleme, koje smo imali što zbog nove tehnologije, što zbog manja znanja., što zbog nedostatka dostupnih materijala.

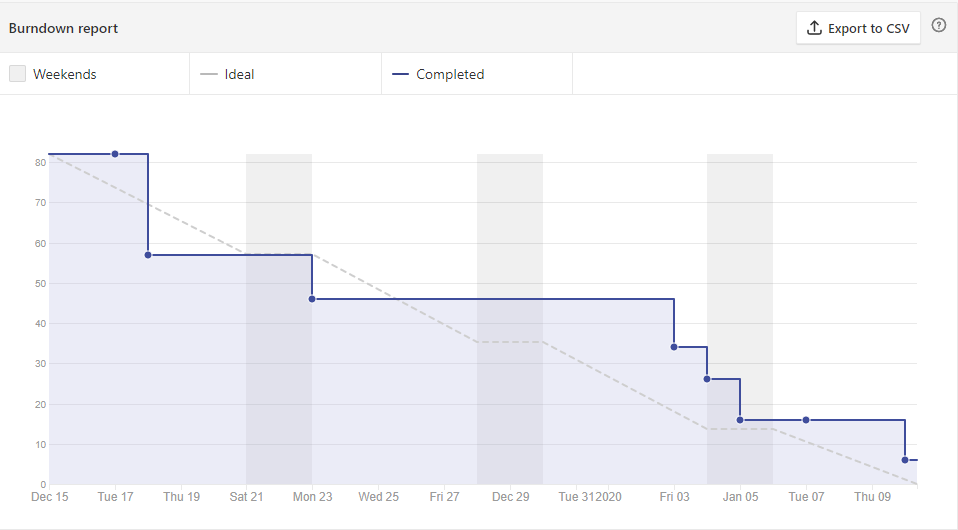
* 1. Prvi sprint



Za prvi sprint odredili smo trajanje od 21 dana., te smo ga u roku , kako smo i isplanirali, izvršili. Odradili smo ga veoma dobro. U prvi sprint stavili smo tehničku dokumentaciju, kreiranje baze podataka, spajanje sa našom aplikacijom, većinom početne funkcionalnosti projekta . Uspjeli smo još odraditi i funkcionalnost prikaza podataka. U ovom sprintu obavili smo konzultacije sa mentorom našeg projekta, timski sastanak u kojem smo napravili reviziju našeg rada, te da odredimo smjernice za sljedeći sprint. Broj sati koji smo si odredili bio je idealno za izvršavanje svih zadanih zadataka. Nije bilo potrebe za dodavanjem dodatnih satova.

* 1. Drugi sprint

Kao i prvom sprintu, tako smo i u drugom sprintu uspjeli izvršiti sve zadane zadatke. Ovaj sprint bio je puno teži i vremenski zahtjevniji od prvog. Razlog težine ovog sprinta sigurno je što su zadaci bili puno kompleksniji, te su bili zadani za izradu funkcionalnosti aplikacije. U ovom sprintu upoznali smo se sa novom tehnologijom. Naučili smo način funkcioniranja programskog okruženja Xamarin. Po prvi puta upoznali smo se s radom WebAPI-em. Sa radom fokusirali smo se dohvaćanje publikacije, te njihovo sortiranje i filtriranje. Svakako najteži dio ovog sprinta bilo je dohvaćanje potrebnih informacija o tome kako se implementira FOI login.

* 1. Treći sprint

Nažalost ovaj sprint nismo uspjeli završiti na vrijeme, iako smo stavili veći broj sati, odnosno trajao je 4 dana duže od prethodnih. Razlog što se ovaj sprint izvodio za vrijeme zimskih praznika svakako nije razlog kašnjenja, štoviše radili smo i više, već je razlog što smo naišli na neke probleme za koje nismo mogli naći pravo rješenje, te smo gubili puno sati na dobivanje odgovora za rješavanje problema. Jedan od najvećih problema u ovom sprintu svakako je bio EDU FOI prijava.