



Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko

VISOKOŠOLSKI STROKOVNI ŠTUDIJ
Delovna praksa
ZAKLJUČNO POROČILO

Hepra d.o.o.

Priimek in ime študenta: Habjan Jernej

Vpisna številka: 63150106

Čas izvajanja delovne prakse: 19.2.2018 – 23.04.2018

Mentor na fakulteti: doc. dr. Mira Trebar

Mentor v podjetju/ustanovi: Marko Kamin, dipl. inž. rač. in inf.

Datum: _____

Podpis študenta:

Podpis mentorja v podjetju/ustanovi:

Kazalo vsebine

1. Uvod.....	1
2. Glavni del.....	2
2.1 Opis podjetja oziroma ustanove.....	2
2.2 Opis delovne prakse – dela študenta.....	2
Delovno mesto.....	2
Zahteve.....	2
Delo ki sem ga opravljal.....	2
Rezultati dela.....	8
Orodja in programski paketi pri delu	9
3. Sklep.....	10
4. Literatura in viri	11

Kazalo slik

Slika 1: Shema podatkovne baze.	4
Slika 2: Glavne tri aplikacije na projektu in njihove povezave.	5
Slika 3: Prikaz dogodka na koledarju.	6
Slika 4: Prikaz dogodka na orodni vrstici.....	7
Slika 5: Struktura aplikacije, generirana s programom Compodoc.....	8
Slika 6: Nadzorna plošča aplikacije.	8

1. Uvod

Hepra d.o.o. je podjetje, ki ima svoje prostore v Kranju, v sodelovanju s podjetji Jakša Group d.o.o. ter MarineXchange Software GmbH iz Avstrije, izdeluje programsko opremo na področju potniških ladij. Delovno prakso sem opravljal v delovni skupini, ki raziskuje nove tehnologije ter pristope za razvoj programske opreme v prihodnosti.

Tako sem bil preusmerjen na nov projekt, ki je vključeval izdelavo spletne aplikacije za prikaz uporabnikovih podatkov in obvestil. Aplikacijo podpirata podatkovna baza SQL in identitetni strežnik za prijavo uporabnikov.

Aplikacijo je bilo potrebno ustrezno strukturirati, saj se bo uporabila na različne načine v večih projektih. Prav tako je bilo pa poleg strukture tudi potrebno urediti dokumentacijo in poročilo o napakah, da lahko zaposleni sprejeli aplikacijo nemoteno.

Med delom sem se veliko naučil povezovanja zalednega dela spletnega programiranja s prikazovalnim, razhroščevanja spletnih aplikacij in zahtev. Prav tako sem se naučil strukturiranega dela z uporabo repozitorija GitHub in drugih aplikacij za pisanje dokumentacije.

2. Glavni del

2.1 Opis podjetja oziroma ustanove

Družba HEPRA d.o.o. [3] je usmerjena v razvoj inovativnih informacijskih sistemov s področja gostinstva in turizma (potniške ladje), javne uprave (prostor, kmetijstvo) in gospodarstva (odločanje in nadzor). Ker je podjetje povezano v mednarodno strukturo, je njihova organizacija ploščate narave in poleg direktorja vključuje še štiri programerje ter systemskega inženirja ter več pogodbenih partnerjev.

Ključna dejavnost je izdelava spletnih in mobilnih aplikacij za moderno podporo poslovnih procesov. Največji del predstavljajo mobilne aplikacije, ki obsegajo celoten sklop poslovanja križarke, torej plavajočega hotela. To zajema tako finančni del, od ponudb in dobave kot tudi logistični v okviru skladiščenja, inventur, vzdrževanja, ter hotelski del s sprejemom in upravljanjem z gosti, rezervacijski sistem, podpore in naročila za restavracije, kuhinje, sobarice, ekskurzije ipd. v več kot 20 ločenih mobilnih aplikacijah.

Njihovi glavni uporabniki so ladijske družbe, kot so Viking Cruises, več ladijskih družb v okviru Carnival Corporation, Aida, Costa, TUI Cruises, kot tudi Kmetijsko Gozdarska Zbornica Slovenije ter agencija za nepremičnine (AKN) v Makedoniji.

2.2 Opis delovne prakse – dela študenta

Delovno mesto

Na delovni praksi sem deloval na delovnem mestu programerja. Moja naloga je bila razvijanje programske opreme s pomočjo integriranega razvojnega okolja kot na primer Microsoft Visual Studio 2017, kjer sem aplikacije tudi razhroščeval. Spletne aplikacije sem testiral v brskalniku Google Chrome, kjer je prav tako nameščen spletni razhroščevalnik za jezik JavaScript.

Ob pričetku delovanja v podjetju, smo skupaj z zaposlenimi pripravili primerno delovno mesto. Sestavili smo mizo, pripravili monitor, podložili prenosni računalnik, da je bil na pravi višini, za tem pa je bil čas pripraviti delovno okolje na računalniku. Nanj sem namestil Microsoft Visual Studio 2017 Community Edition [13], pripravil okolje Node.js in pakete za pričetek za delo s projektom.

Zahteve

Zahteve za delo so bile predznanje spletnega programiranja, obvladovanje arhitekture Model-Pogled-Kontroler, upravljanje s podatkovnimi bazami tako na lokalnem strežniku, kot na spletni platformi, znanje programskih jezikov C#, JavaScript, TypeScript [12], SQL [10], urejanje projekta z ukaznimi pozivi na operacijskem sistemu Windows in poznanje integriranega razvojnega okolja Visual Studio. Programskega jezika TypeScript ob pričetku dela nisem znal, zato sem se ga pred delom na aplikaciji moral naučiti.

Delo ki sem ga opravljal

Sprva sem se naučil osnov programskega jezika Angular [2] na poskusnem projektu, katerega sem pripravil in izgradil v okolju Visual Studio. Ko sem se s strukturo aplikacije in komunikacijo pogleda z zalednim delom bolj spoznal, sem k projektu dodal podatkovno bazo. Sprva je bila podatkovna baza dostopna na lokalni napravi, pozneje pa sem jo premestil na spletni platformi Microsoft Azure [5], za katerega smo v sklopu študija dobili licenco za uporabo storitev. Prav tako sem na njej objavil začasno aplikacijo.

Glavni projekt, ki je vključeval predlogo PrimeNG Atlantis [8], sem moral povezati z zalednim delom, kar je bilo moje glavno delo čez celotno prakso. Predloga pa je predstavljala izgled strani s html in css kodo brez funkcionalnosti.

Namen projekta je bil razbitje aplikacije s predlogo PrimeNG Atlantis na komponente ali module, ki se lahko ponovno uporabijo v drugi aplikaciji z nekaj preuredbami in hitro povezljivostjo in neodvisnostjo komponent samih. Tu je bil podan velik poudarek na arhitekturi Model-Pogled-Kontroler, ki je osnova pri zgradbi take aplikacije. To arhitekturo smo prav tako srečali pri predmetu Spletne tehnologije [11], kjer smo si pogledali komunikacijo med predstavitev in zalednim delom na bolj direkten način, vendar Angular to zelo poenostavi z vgrajenimi asinhronimi klici.

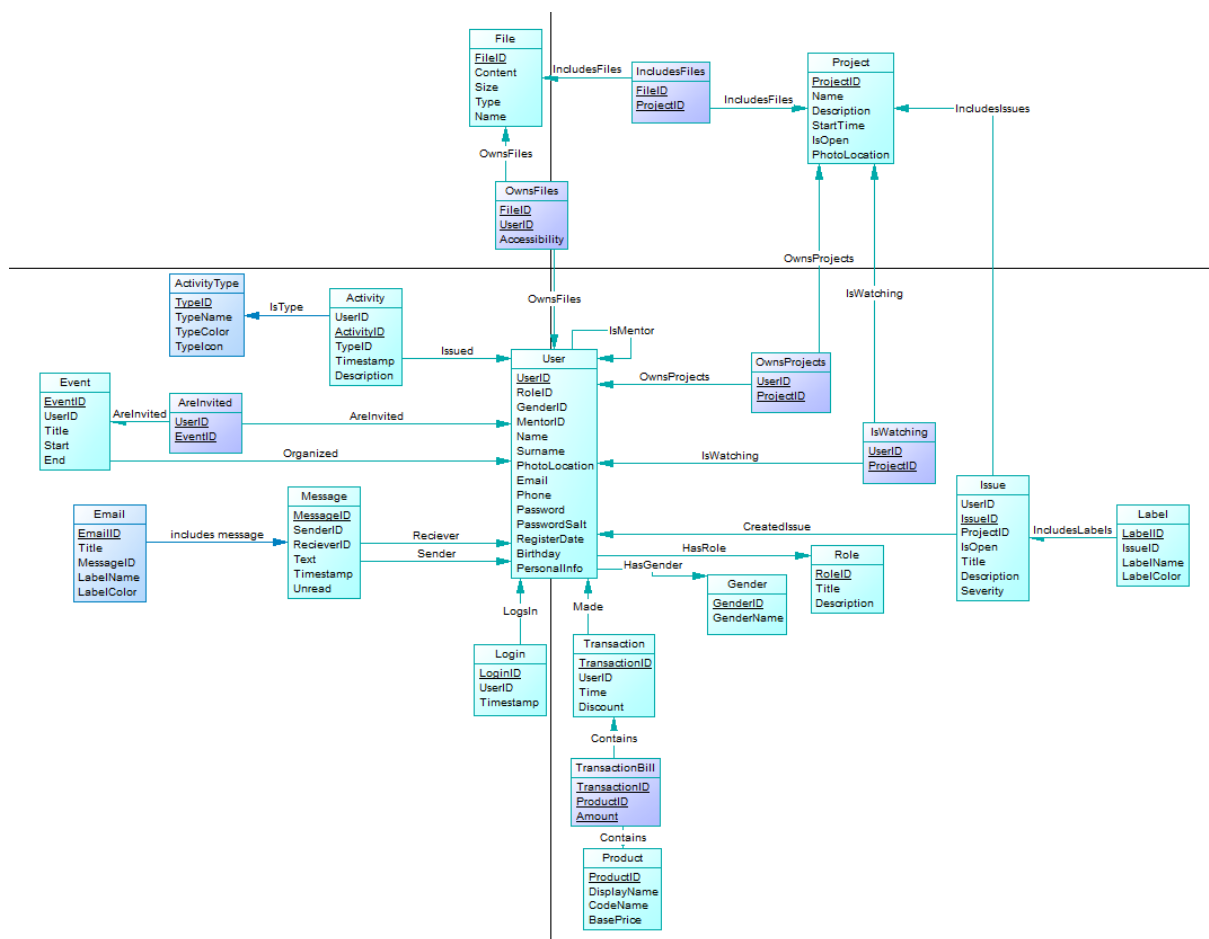
Prav tako mi je bila pri tej arhitekturi v pomoč vsebina predmeta Informacijski sistemi [4], kjer smo imeli opravka s spletnim strežnikom ISS in platformo Azure in upravljanjem podatkovne baze. Pridobljeno znanje mi je bilo v veliko pomoč pri gradnji te aplikacije, saj smo pri predmetu pokrili lep del teh tehnologij, celo do te mere, da sem si včasih tudi pomagal s kodo, ki smo jo sprogramirali za seminarsko nalogo.

Vendar je bilo potrebno spletno aplikacijo še posodobiti na novejšo različico paketov za lažje nadaljnje razvijanje. Aplikacijo sem predelal na logični sloj .NET Core 2.0 [1] in posodobil pakete z NPM (Node Package Manager). Posodobitev nekaterih paketov je povzročalo težave, ki še nisem znal razrešiti, zato teh nisem posodobil. Prav tako sem posodobil Angular 4 na 5.

Za nadaljnje delo mi je mentor razložil delovanje in strukturo aplikacije in podal razne ideje, kaj vse bi bilo lahko povezano z aplikacijo. Ena izmed teh idej je bil identitetni strežnik, ki je način potrditve uporabnikov. Prednost tega strežnika je v tem, da se ob uspešni uporabnikovi prijavi strežnik pogovori z drugim programom, ki ima dostop do podatkov in podatkovne baze, o uporabnikovih pravicah. Podatkovni strežnik za tem ve, do katerih podatkov lahko uporabnik dostopa in do katerih ne sme. Prav tako lahko ta strežnik predstavlja enoten sistem za prijavo in registracijo uporabnikov skozi več aplikacij.

Tako sem implementiral identitetni strežnik z uporabo raznih projektov in predlog na spletu, vendar dostop do njega iz spletne aplikacije še ni bil možen.

Za tem sem se posvetil samim podatkom, zato sem podatkovno bazo objavil na platformo v oblaku Microsoft Azure. Za zasnovo in izdelavo podatkovne baze [Slika 1] sem uporabil program PowerDesigner [9], ki smo ga uporabljali pri predmetu Podatkovne baze 2 [7].



Slika 1: Shema podatkovne baze.

Aplikacije PowerDesigner smo se pri tem predmetu dobro naučili, kar je zelo koristilo pri hitri zasnovi podatkovne baze od konceptualnega do logičnega in fizičnega modela. Model sem pretvoril v skripto, ki ustvari podatkovno bazo, tabele in relacije v Microsoft SQL podatkovni bazi.

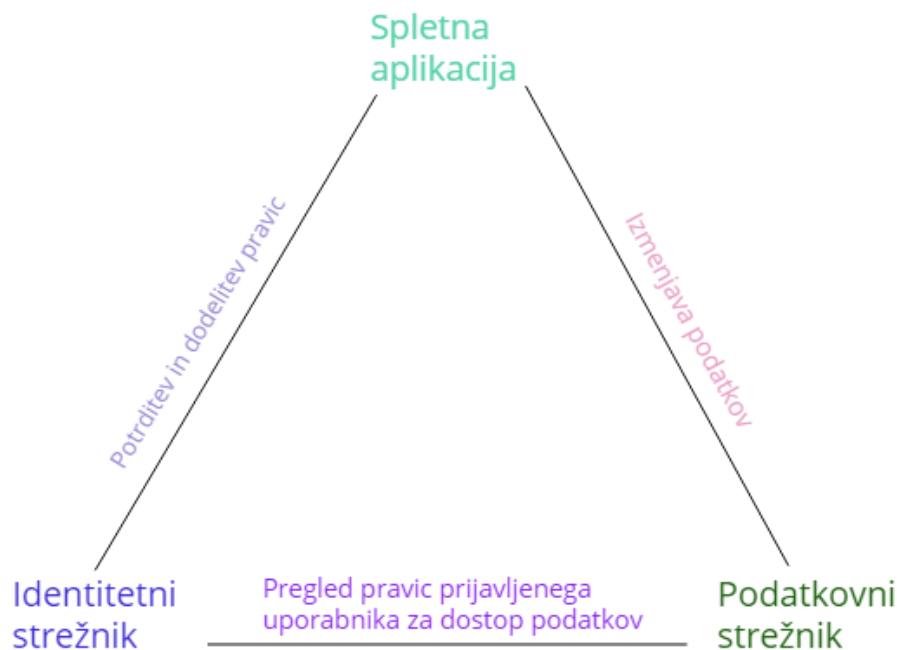
Potrebna je bila izdelava shranjenih procedur, ki se izvajajo ob klicu programa za pridobivanje podatkov. S procedurami sem pokrival celotno zbirko ukazov za manipulacijo s podatki v bazi (pridobitev, vnos, posodobitev in brisanje).

Tako sem lahko pridobil podatke iz podatkovne baze v oblaku, kot tudi te podatke vanjo vnašal. Da pa bi bili vneseni podatki pravi, sem moral narediti kontrolo podatkov, tako da ta uporabniku sporoči napako, če podatki pri vnosu manjkajo, je geslo prekratko ipd.

Ob pričetku uporabe več različnih virov, sem moral pričeti s pisanjem dokumentacije za projekt. Dokumentacijo sem pisal v preproste tekstovne datoteke, ki sem jih priložil zraven projekta. Dokumentacija je ob tem trenutku pokrila pridobivanje podatkov iz podatkovne baze, uporabo procedur ipd.

Prav tako sem shranjeval procedure v datoteke zraven projekta, da jih v primeru izbrisa podatkovne baze ne bi izgubil.

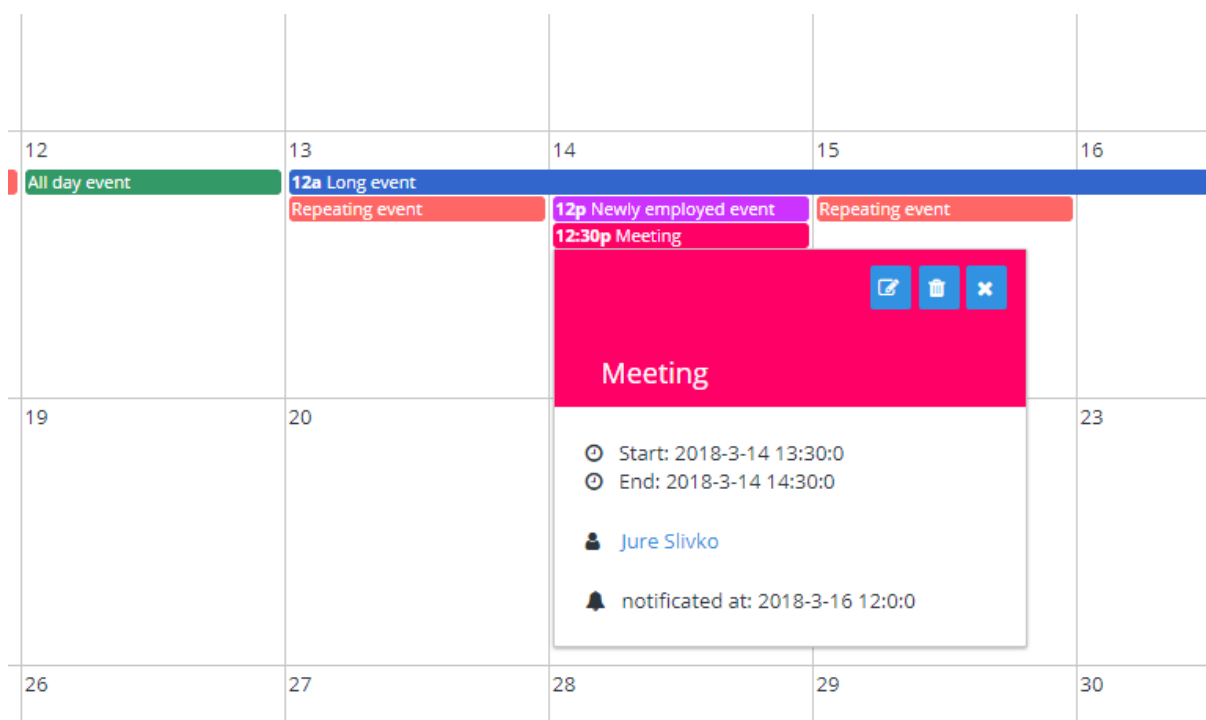
Tako sem imel pripravljene vse tri večje aplikacije za pričetek povezovanja [Slika 2].



Slika 2: Glavne tri aplikacije na projektu in njihove povezave.

Sledilo je povezovanje podatkov shranjenih v podatkovni bazi s spletno aplikacijo. Zato sem si moral natančneje ogledati PrimeNG komponente, ki sem jih uporabljal za prikaz podatkov, kot na primer tabele, spustni meniji ipd.

Sprva sem povezal naslovno stran, kjer je bilo potrebno prilagoditi tudi procedure v podatkovni bazi, da so podpirale željene podatke. Sem so spadale komponente kot so prikaz osnovnih informacij, uporabniške informacije, seznam kontaktov, obrazec za kontakt, pogovorno okno (ki sem ga pozneje še nadgradil), seznam nalog (za zgled sem vzel obliko seznamov v iz spletne aplikacije Google Keep), prikaz itinerarja ki trenutno prikazuje aktivnosti ki so se pripetile, tabela, graf, koledar (za zgled sem vzel pogled Google koledar aplikacije) [Slika 3].

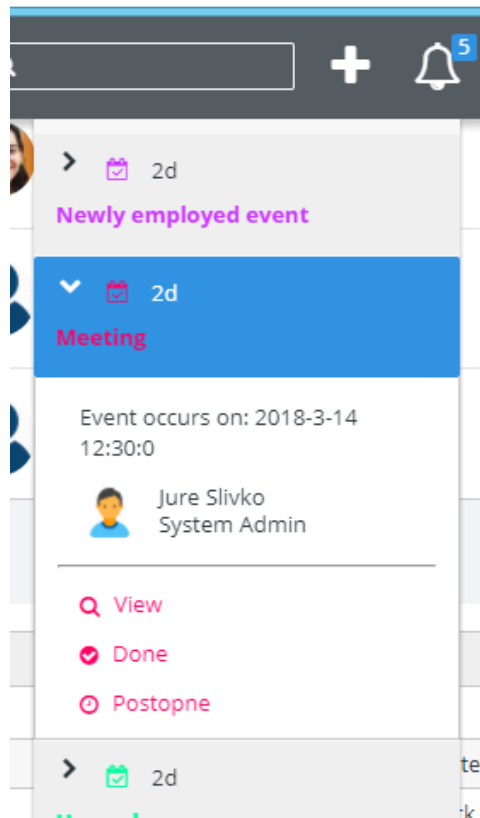


Slika 3: Prikaz dogodka na koledarju.

Ko so bile te komponente povezane, sem izdelal nove strani za pregled in ustvarjanje projekta, aktivnosti, elektronskega sporočila in profila. Preusmerjanje poteka drugače, saj je aplikacija enostranska spletna aplikacija, kar pomeni da se stran naloži enkrat, za tem pa se samo pomikamo po raznih delih strani brez večjega nalaganja, ampak se vsebina dinamično posodablja.

Nadaljnje delo je bilo predelava komponent na namizni plošči tako, da vsaka sodi v svojo datoteko, ki jo potem lahko vključimo v drugo komponento, da se ta tam izriše.

Prav tako je bilo potrebno prikazati podatke na orodni in stranski vrstici, kjer sem prikazal elektronsko pošto, projekte in sodelavce na projektih, kot tudi aktivnosti ki so jih sodelavci naredili. Za prikaz elektronske pošte in aktivnosti na orodni vrstici [Slika 4] sem videz povzel po Androidovih sporočilih, kjer lahko sporočilo hitro skrijemo, lahko pa s klikom nanj pogledamo vsebino.



Slika 4: Prikaz dogodka na orodni vrstici.

Ko je kar naenkrat nastalo veliko različnih strani in komponent, ki pa so imele vpisane specifične parametre za prevzem podatkov, sem moral pričeti pisati dokumentacijo za celotni projekt. Na platformi GitHub, kjer sem objavljaj napake in novosti in shranjeval verzije projekta, sem dodal dokumentacijo, ki popelje bralca skozi predstavitev projekta, nalaganje okolja na nov računalnik, predstavitev posameznih aplikacij in njihovo medsebojno komunikacijo, do posameznih večjih modulov, specifičnih datotek in objavljanje projekta na platformi Azure.

Za dokumentacijo funkcij, razredov in storitev pa sem dodal komentarje k kodi v projektu. S programom Compodoc [Slika 5] sem potem iz teh komentarjev in kode avtomatsko generiral dokumentacijo, ki je vidna v prikazovalniku.

Rezultat dela je delujoča modularna spletna aplikacija, ki se povezuje s podatkovnim strežnikom na varen način z uporabo verifikacije na identitetnem strežniku. Spletna aplikacija ima tudi nadgrajene komponente, kot so bile podane v PrimeNG Atlantis predlogi, z različnimi dodatki [Slika 6]. Prav tako ima posodobljene pakete za moderno programiranje in hitrejše razvijanje. Spletna aplikacija je splošna aplikacija, ki je namenjena temu, da se iz nje izgradi več različnih aplikacij, ki bodo specifične za svoje področje. Zaenkrat je aplikacija na voljo vsem uporabnikom, da lahko dostopajo do pregleda komponent in njihove povezave s podatkovno bazo.

Testna različica spletne aplikacije je trenutno na voljo na naslednjem spletnem naslovu:

<http://angularprimeng.azurewebsites.net/>.

Orodja in programski paketi pri delu

- spletni strežnik: Microsoft Internet Information Server [6],
- podatkovni sloj: SQL Server,
- logični sloj: .NET Core 2.0,
- Predstavitveni sloj: Typescript, Angular, PrimeNG Atlantis,
- platforma v oblaku: Microsoft Azure,
- aplikacije: Visual Studio 2017 Community Edition, PowerDesigner.

3. Sklep

Podjetje me je na delovno prakso sprejelo z odprtimi rokami, kar je pripomoglo k motiviranosti pri delu, saj sem delo z veseljem opravljajal. Zelo vesel sem bil pa tega, ker sem dobil za delo že izdelano temo s strani PrimeNG Atlantis. Pri predlogi nisem skoraj nič urejal obliko, ampak dograjeval njen zaledni del in povezoval s podatkovno bazo s pomočjo najnovejših spletnih tehnologij, kot na primer Angular, Node.js ipd.

Skozi prakso sem bil popeljan na pravi način, saj sem bil v začetku, da sem se dela privadil, predstavljen s koncepti in pričel izdelovati poskusno aplikacijo, za tem mi je pa mentor predstavil pravo aplikacijo, ki pa sem jo počasi začel dograjevati. Prav tako sem od mentorja dobil veliko idej za implementacijo različnih delov aplikacije, ki pa sva jih skupaj argumentirala, če so ti res primerni.

Z opravljenim delom sem zadovoljen, saj sem projekt dovolj dokumentiral in strukturiral, da bodo lahko zaposleni naprej čim hitreje z njim delali. Še posebej sem vesel, ker se bo ta splošna aplikacija naprej uporabljala na več različnih načinov. Prav tako imam dober občutek, saj sem podjetju prispeval programsko opremo, ki jo bodo zaposleni v podjetju začeli razvijati in se bo uporabljala v prihodnosti. Z delovno prakso sem še izboljšal delovne navade, saj delo v skupini prisili do strukturiranosti in organiziranosti programa.

Prav tako pa je delovna praksa zelo dober uvod za delo po študiju. Kandidatu in zaposljevalcu ponudi čas, da se drug drugemu ustrezno prilagodita. Komu, ki še ni nikoli delal v podjetju, bi prakso močno priporočil, saj dobro privadi študenta na nov način dela, ki je pa povsem drugačen od tega na fakulteti.

Na nov način dela posebej vpliva komunikacija, ki je kar naenkrat odprta, in ekipa deluje skupno k cilju, kar je namen tudi marsikaterih nalog v šoli, vendar tam ta vez včasih ni tako močna.

4. Literatura in viri

- [1] (2018) NET Core 2.0. Dostopno na: <https://docs.microsoft.com/en-us/aspnet/core/>.
- [2] (2018) Angular. Dostopno na: <https://angular.io/>.
- [3] (2018) Hepra d.o.o. Dostopno na: <https://www.hepra.com/>.
- [4] (2018) Informacijski sistemi. Dostopno na: <https://ucilnica.fri.uni-lj.si/course/view.php?id=145>.
- [5] (2018) Microsoft Azure. Dostopno na: <https://azure.microsoft.com/en-us/>.
- [6] (2018) Microsoft Internet Information Server. Dostopno na: <https://www.iis.net/>.
- [7] (2018) Podatkovne baze 2. Dostopno na: <https://ucilnica.fri.uni-lj.si/course/view.php?id=144>.
- [8] (2018) PrimeNG Atlantis. Dostopno na: <https://www.primefaces.org/layouts/atlantis-ng>.
- [9] (2018) SAP PowerDesigner. Dostopno na: <https://www.sap.com/slovenia/products/powerdesigner-data-modeling-tools.html>.
- [10] (2018) SQL Server. Dostopno na: <https://www.microsoft.com/en-us/sql-server/>.
- [11] (2018) Spletne tehnologije. Dostopno na: <https://ucilnica.fri.uni-lj.si/course/view.php?id=141>.
- [12] (2018) TypeScript. Dostopno na: <https://www.typescriptlang.org/>.
- [13] (2018) Visual Studio 2017 Community Edition. Dostopno na: <https://www.visualstudio.com/>.