

*Prolećni semestar, 2022/23*

*PREDMET:*

***SE325 - Upravljanje projektima razvoja softvera***

**Onlajn prodavnica ploča i CD-ova**

Projektni zadatak broj 1

|  |  |
| --- | --- |
| Asistent: | **Sara Nikolić** |
| Student: | **Bojana Stajić** |
| Indeks: | **4596** |

[Uvod 3](#_Toc72689324)

[Apstrakt aplikacije 3](#_Toc1738760570)

[Lista uspešnosti projekta (Project Success Chart) 4](#_Toc1224049471)

[Teoretska postavka 4](#_Toc1851251863)

[Primena liste uspešnosti projekta 4](#_Toc836606655)

[Struktura poslova WBS (Work Breakdown Structure) 6](#_Toc1435330507)

[Teorijska postavka 6](#_Toc1635996963)

[Primena strukture poslova 6](#_Toc1800302591)

[Lista zaduženja (Task Assignment List) 7](#_Toc2076665280)

[Teoretska postavka 7](#_Toc394825467)

[Primena liste zaduženja (Tast assignment List) 8](#_Toc221063814)

[Lista rokova (Task Duration List) 11](#_Toc1572214643)

[Teoretska postavka 11](#_Toc2105972364)

[Primena liste rokova (Task Duration List) 11](#_Toc1005664982)

[Mrežni plan projekta (Project Network Diagram) 12](#_Toc1518208007)

[Teoretska postavka 12](#_Toc371243579)

[Primena mrežnog projekta (Project Network Diagram) 13](#_Toc978870839)

[Gantov dijagram raspored poslova (Grantt chart schedule) 15](#_Toc1823906048)

[Teoretska postavka 15](#_Toc663906136)

[Primena gantovog dijagrama rasporeda poslova (Grantt chart schedule) 15](#_Toc1771130586)

[Lista budžeta projekta (Project Budget Chart) 16](#_Toc283455722)

[Teoretska postavka 16](#_Toc1376221546)

[Primena liste budžete projekta (Project Budget Chart) 17](#_Toc1007325864)

[Zaključak 19](#_Toc1365966922)

[Literatura 19](#_Toc503688571)

# Uvod

U ovom dokumentu biće predstavljene tehnike i načini upravljanja projektima razvoja demonstriranim na aplikaciji koja predstavlja onlajn prodavnicu ploča i CD-ova. U pitanju je sistem koji će olakšati kupovinu CD-ova i ploča krajnjem korisniku kao i sistem na kome će korisnik moći imati korisnički interfejs koji je lak i intuitivan za realizaciju kupovine. Veoma je važno pripremiti ceo proces kreiranja projekta od početka do kraja i u tome veoma pomažu standardizovane tehnike od kojih će neke biti primenjene ovde:

* Lista uspešnosti projekta (Project Success Chart)
* Struktura poslova WBS (Work Breakdown Structure)
* Lista zaduženja (Task assignment List)
* Lista rokova (Task Duration List)
* Mrežni plan projekta (Project Network Diagram)
* Gantov dijagram raspored poslova (Grantt chart schedule)
* Lista budžeta projekta (Project Budget Chart)

# Apstrakt aplikacije

Aplikacija za predlog za izradu projektnog zadatka jeste sistem koji je zamišljen kao onlajn prodavnica ploča i CD-ova. Namenjena je korisnicima i administratoru sistema. Svi korisnici sistemu pristupaju sa korisničkim imenom i šifrom, ali funkcionalnosti se razlikuju po ulogama korisnika.

Prodavnici se pristupa putem aplikacije za telefon ili tablet uređaj na kome korisnici kada se uloguju najpre imaju uvid u ponudu prodavnice, pretragu proizvoda, dodavanje proizvoda u korpu i kupovinu proizvoda. U uvidu pojedinačnog proizvoda, korisniku osim samo naziva i cene biće date i informacije o izvođaču i o trenutnoj ploči/CD-u, a u opciji kupovina proizvoda aplikacija će za korisnika na osnovu unete adrese dostave izračunati poštarinu i eventualne carinske troškove ukoliko postoje jer su to pitanja koja najčešće brinu korisnike ovakvih sistema, bili oni internacionalni ili ne. Korisnici takođe imaju pristup svom istorijatu kupovine u prodavnici. Admin korisnici sa druge strane će imati uvid u korisnike sistema i proizvode kao i mogućnost upravljanja svim tim podacima shodno potrebi.

Krajnji cilj aplikacije jeste sistem koji omogućava kupovinu pomenutih proizvoda na organizovan i pouzdan način koji će biti lak za korišćenje i intuitivan, a na samom kraju i inovantivan jer kombinuje razne moderne a i nove funkcionalnosti u ovakav tip sistema.

# Lista uspešnosti projekta (Project Success Chart)

## Teoretska postavka

Lista uspešnosti projekta predstavlja jedan od sedam koraka na putu ka uspešno razvijenom projektu, pogotovo prilikom razvoja projekata malih projekata. Ovaj korak podrazumeva korišćenje grafikona, tabela, dijagrama i drugih načina vizualnih prikaza podataka, upravljanjem kojih na adekvatan način pratimo usepešnost razvoja jednog projekta. Ova metoda nam lista faktore prema kojima se meri uspeh ishoda jednog projekta.

Dalje je važno definisati cilj sa funkcionalnostima koje aplikacija koju razvijamo mora posedovati, a prema kojoj se dalje vodi metrika uspešnosti samog projekta.

Ispunjenje zahteva i konkretizacija rokova u odnosu na vremenski okvir jeste od krucijalne važnosti tokom procesa planiranja projekta, što dalje znači da rangiranje uspešnosti projekta direktno zavisi od očekivanja stejkholdera u odnosu na jedinicu vremena. Zbog ovih navedenih stavki je važno da svi članovi tima u svakom trenutku budu informisani o očekivanjima stejkholdera koji su jedinstveni za svaki projekat, a koji neposredno utiču na merenje uspeha rukovođenja jednim projektom.

## Primena liste uspešnosti projekta

**Cilj projekta:**

Kreiranje aplikacije koja služi za kupovinu ploča i CD-ova online putem putem jednostavnog interfejsa. Rok za izradu projekta je 3 meseca, preciznije do kraja juna 2023. godine.

**Isporuka delova softvera:**

1. Na početku projekta nakon prikupljanja informacija i zahteva koji smo dobili iz razgovora sa stejkholderima, treba da analiziramo i definišemo kao i da validiramo te zahteve. Kao rezultat, dobija se SRS dokument(dokument zahteva). Za ovaj početni deo određuje se da je potrebno 10 dana.
2. Sledeći korak, nakon adekvatno definisanih zahteva, jeste projektovanje arhitekture sistema za šta neće biti potrebno više od 5 dana.
3. Nakon što je projektovanje arhitekture završeno i odobreno, počinje se faza implementacije sistema, za koju je potrebno 25 radnih nedelja (okvirno 6 meseci). U ovoj fazi se vrši komplentna implementacija svih podsistema uz njihovo kontinualno testiranje i integraciju.
4. Završetkom faze inicijalizacije, sistem se nalazi u alfa fazi. Vrši se alfa testiranje sa predodređenom grupom test korisnika. Testiranje i analiza se vrši iterativno. Nakon faze testiranja koja traje nedelju dana narednih nedelju dana posle toga se vrši popravka potencijalnih defekata sistema. Ovaj proces se ponvalja najviše 3 puta ili dok se ne dostigne zadovoljavajući nivo rada sistema.

**Stejkholderi:**

1. Naručilac softvera
2. Korisnici sistema

**Kriterijumi uspešnosti:**

1. Projekat zadovoljava/ispunjuje sve zahteve sjektholdera. Svi zahtevi definisani u SRS dokumentu su ispunjeni u fleksibilnost neispunjavanja određenih nefunkcionalnih zahteva.
2. Projekat nije probio definisani budžet.
3. Veći broj korisnika sistema je zadovoljen korisničkim iskustvom.
4. Sistem je stabilan i ima brz odziv.
5. Naručilac softvera brzo pridobija nove korisnike na sistem.

# Struktura poslova WBS (Work Breakdown Structure)

## Teorijska postavka

Svaki projekat koji se realizuje i razvija na nivou jednog tima implicira da se rad na jednoj funkcionalnosti sistema pojednostavljuje na veći broj manjih zadataka koji se dele u listu na članove tima. Svi ti manji zadaci su delovi jedne celine, odnosno funkcionalnosti, za čiju izradu je definisan rok prema stejkholderima, i koji treba predvideti i komunicirati transparentno, sa uračunatim rizikom.

## Primena strukture poslova

Da bi isporučili naš sistem, potrebno je veće celine sistema podeliti na više manjih delova.

1. **Planiranje**
   1. Definisanje cilja
   2. Prikupljanje stejkholdera
   3. Prikupljanje zahteva
   4. Specifikacija zahteva
   5. Definisanje budžeta
   6. Formiranje tima
   7. Definisanje rokova
2. **Implementacija**
   1. Razvoj prototipa (opciono)
   2. Razvoj arhitekture sistema
   3. Razvoj baze podataka
   4. Razvoj korisničkog interfejsa
   5. Razvoj funckionalnosti
3. **Testiranje**
   1. Testiranje funckionalnosti
   2. Testiranje korisničkog interfejsa
   3. Alfa testiranje
   4. Beta testiranje
4. **Validacija**
   1. Ispravljanje grešaka
5. **Produkcija**
   1. Isporuka sistema na tržište
6. **Održavanje**
   1. Anketiranje krajnjih korisnika
   2. Dodavanje novih zadataka
   3. Ispravljanje grešaka

# Lista zaduženja (Task Assignment List)

## Teoretska postavka

Tokom procesa razvoja softvera, raspodela odgovornosti ima veliki značaj. Ovo dalje podrazumeva konkretnu i transparentnu podelu poslova između članova tima. Konkretnost i transparentnost egzistiraju kao dva stuba raspodele zadataka a koriste se radi obavljanja posla, kao i međusobnoj informisanosti i komunikaciji članova tima čija važnost dolazi do izražaja u trenutku spajanja jednostavnih zadataka u jednu opđtu funkcionalnost samog proizvoda.

Vođa tima ima ulogu i obligaciju raspodele konkretnih zadataka pojedincima tima, u skladu sa njihovim znanjem, veštinama, iskustvom kao i nivoom senioriteta. Važan uticaj na proces raspodele poslova imaju doušnici, odnosno posrednici u transferu znanja koje se tiče obavljanja samih projektnih zadataka. Ovi pojedinci zaduženi su za prenos korisnih informacija od strane razvoja drugog dela softvera koji može biti od pomoći onima koji razvijaju neki drugi deo projekta, kao i pružati neophodna pojašnjenja kojima se dobija na vremenu i produktivnosti projekta.

Veliki deo odgovornosti nose pojedinci koji putem liste zaduženja dobiju zadatak da odobre isporučeni deo proizvoda. Ovaj zadatak se u nekim situacijama ostavlja stejkholderima, ali potrebno je uveriti se u kvalitet isporučenog dela proizvoda pre samog predstavljanja funkcionalnosti stejkholderima, odnosno krajnjim korisnicima sistema.

## Primena liste zaduženja (Tast assignment List)

Prilikom formiranja tima treba uzeti u obzir obim posla, budžet i rok projekta kako bi se formirao idealan tim. Na mom projektu tim čini 5 čoveka, definisani u tabeli ispod.

|  |  |
| --- | --- |
| Ime i prezime | Pozicija |
| Bojana Stajić | Project Leader |
| Stefan VELIČKOVIĆ | Java developer |
| ANA MARKOVIĆ | Frontend developer |
| BOGDAN MILOVANOVIĆ | Backend developer |
| MIODRAG KAŠIKOVIĆ | Designer |

**Lista zadataka koje potrebno izvršiti:**

1. Prikupljanje zahteva
2. Kreiranje arhitekture sistema
3. Kreiranje baze podataka
4. Dodeljivanje zadataka
5. Implementacija backend-a
6. Dizajn korisničkih interfejsa
7. Implementacija KI (Web)
8. Integraciono tesitranje
9. Ispravka grešaka
10. Implementacija bezbednostnih mehanizama
11. Postavljanje aplikacije na server

U narednoj tabeli, možemo videte raspodelu zadataka, podršku na zadacima, odakle se informacije za zadatak prikupljaju i ukoliko ima potrebe za tim, ko pruža odobrenje za izvršavanje zadataka.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zadatak | Vlasnik zadatka | Podrška | Pružalac info. | Odobrenje |
| Prikupljanje zahteva | Bojana Stajić |  | Stejkholderi |  |
| Kreiranje arhitekture sistema | Bogdan Milovanović | Bojana Stajić |  |  |
| Kreiranje baze podataka | Bogdan Milovanović | Bojana Stajić |  |  |
| Dodeljivanje zadataka | Stefan Veličković | Bogdan Milovanović |  |  |
| Implementacija backend-a | Bogdan Milovanović |  |  |  |
| Dizajn korisničkih interfejsa | Ana Marković | Stefan Veličković |  |  |
| Implementacija KI (Web) | Ana Markvić |  | Miodrag Kašiković | Miodrag Kašiković |
| Integraciono tesitranje | Bogdan Milovanović | Bojana Stajić, Ana Marković |  |  |
| Ispravka grešaka | Bojana Stajić, Stefan Veličković,  Ana Marković,  Bogdan Milovanović |  |  |  |
| Implementacija bezbednostnih mehanizama | Bogdan Milovanović | Bojana Stajić |  |  |
| Postavljanje aplikacije na server | Bogdan Milovanović | Bojana Stajić |  |  |

# Lista rokova (Task Duration List)

## Teoretska postavka

Lista rokova jeste faktor koji u velikom delu utiče na vreme izrade funkcionalnosti na softveru, a u krajnjoj liniji, vreme izrade kompletnog proizvoda. Organizacijski gledano, ova lista mora do detalja ispratiti listu zahteva o kojoj je prethodno bilo reči. Kako bi se izbeglo nerealno ubrzavanje ili usporavanje samog razvoja, važno je realno predvideti potrebno vreme izrade svake od predloženih funkcionalnosti. U velikoj meri će posao olakšati lista zahteva i podela tih zahteva na manje celine – na ovaj način, developeri lakše prave predikciju o vremenu potrebnom za izradu svakog zahteva zasebno.

Okvirno predviđanje vremena potrebnog za izradu softverskog proizvoda developeri i stejkholderi definišu uzevši u obzir:

1. Optimistično vreme
2. Verovatno vreme
3. Pesimistično vreme
4. Ispostavljeno vreme

Ovo znači da vreme izrade predstavlja varijablu koju nije uvek moguće precizno definisati, te se u obzir uzima da optimični vremenski okviri mogu biti ostvareni u 10-20% slučajeva, dok se verovatno vreme ostvaruje u 50% slučajeva, a pesimistične estimacije vode od 80% do 90% uspeha što dodatno potrđuje činjenicu da veliki broj projekata “probija rokove” zbog nedovoljno realne esitmacije vremenskog okvira.

## Primena liste rokova (Task Duration List)

U sledećoj tabeli možemo videti okvirno vreme izvršavanje svakog od zadatka u danima posebno:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zadatak | Optimistično vreme | Najverovatnije vreme | Pesimistično vreme | Ispostavljeno vreme |
| Prikupljanje zahteva | 3 | 4 | 5 |  |
| Kreiranje arhitekture sistema | 3 | 4 | 5 |  |
| Kreiranje baze podataka | 1 | 1 | 2 |  |
| Dodeljivanje zadataka | 1 | 1 | 1 |  |
| Implementacija backend-a | 25 | 30 | 35 |  |
| Dizajn korisničkih interfejsa | 3 | 4 | 5 |  |
| Implementacija KI (Web) | 20 | 25 | 30 |  |
| Integraciono tesitranje | 5 | 7 | 10 |  |
| Ispravka grešaka | 5 | 10 | 15 |  |
| Implementacija bezbednosnih mehanizama | 5 | 6 | 7 |  |
| Postavljanje aplikacije na server | 2 | 2 | 3 |  |
| Ukupno | 73 | 94 | 118 |  |

# Mrežni plan projekta (Project Network Diagram)

## Teoretska postavka

Nakon što smo definisali konkretne zadatke i vreme potrebno za njihovu implementaciju na određenom softverskom proizvodu, bitno je da implementiramo redosled faza razvoja samog softvera. Zbog toga, bitno je uzeti u obzir sve međusobne odnose faza razvoja i mere njihove međuzavisnosti. Baš zbog toga, krucijalno je napraviti mrežni plan projekta koji će definisati prioritete u procesu pravljenja funkcionalnosti, a posebno u onim delovima projekta kada se faze razvoja oslanjaju jedna na drugu. Time je prvenstveno važno srodne i paralelne zadatke grupisati kroz faze razvoj tako da svi zadaci koji budu eventualno od njih zavisili bili predviđeni za izradu u narednoj fazi razvoja softverskog proizvoda. Dakle, međusobno nezavisni zadaci se mogu obavljati paralelno od strane dvoje različitih ljudi, a svi zahtevi koji su u korelaciji sa njima se moraju obaviti u narednim fazama koje će zahtevati prethodno implementiranu osnovu za njihov nastanak i razvoj.

Mrežni plan projekta daje nam uvid i u srodne zahteve koji formiraju „linije života“ koje razvijaju različite funkcionalnosti zasebn i koje se na kraju povezuju u celinu koja predstavlja gotov softverski proizvod. To nam dalje daj daje uvid i u „Kritičnu liniju života“ koja u samom planiranju faza nosi važnost najvišeg prioriteta. To je najduža i najkompleksnija linija na dijagramu jednog projekta, od koje esencijalno zavisi ishod samog projekta, odnosno sam uspeh softverskog proizvoda. „Kritična linija života“ ne spaja nužno sve faze razvoja proizvoda, već ubraja sve neophodne komponente jednog funkcionalnog proizvoda, što znači da bez ove linije softversko rešenje ne može biti isporučeno. Ostale putanje jesu „Linije snabdevanja“ koje predstavljaju redunantni deo samog proizvoda, međutim imaju veliki značaj kako bi proizvod bio isporučen u celosti, i kako bi svi zahtevi stejkholdera bili adekvatno ispunjeni.

Mrežni plan takođe zavisi i od liste rokova, u istoj meri u kojoj zavisi i od liste zadataka. Prilikom kreiranja mrežnog plana, veoma je važno pronaći i grupisati sve neophodne funkcionalnosti kako bi Kritična linija bila što konkretnije i preciznije definisana. U ove svrhe je na listi rokova potrebno obratiti posebnu pažnju na kolonu verovatnog vremena, kako bi najduža linija na projektu bila optimalno predviđena.

## Primena mrežnog projekta (Project Network Diagram)

Primenom mrežnog plana projekta na ovom projektu/aplikaciji data nam je na uvid koja komponenta zavisi od razvoja drugih, kao i koje komponente se mogu paralelno razvijati i uvid u vremenski period realizacije komponenata.

Za početak nam je potrebno podsetiti se liste zadataka koje treba da obavimo.

**Lista zadataka koje potrebno izvršiti:**

1. Prikupljanje zahteva
2. Kreiranje arhitekture sistema
3. Kreiranje baze podataka
4. Dodeljivanje zadataka
5. Implementacija backend-a
6. Dizajn korisničkih interfejsa
7. Implementacija KI (Web)
8. Integraciono tesitranje
9. Ispravka grešaka
10. Implementacija bezbednostnih mehanizama
11. Postavljanje aplkacije na server

A diagram of a software development

Description automatically generated

# Gantov dijagram raspored poslova (Grantt chart schedule)

## Teoretska postavka

Ova tehnika se striktno fokusira na menadžment vremena kao važnog ograničavajućeg faktora i resursa. Osim predstavljanja podele poslova, važno je estimacijom odrediti dužinu trajanja faza projekata, u okviru kojih će se odvijati razvoj zacrtanih tokova razvoja funkcionalnosti jednog softverskog proizvoda.

Gantov dijagram, odnosno Gantogram, predstavlja grafički prikaz rokova sa analizom i podelama zaduženja u okviru jednog tima koji radi na softveru. Ovo znači da Gantov dijagram mora biti u koheziji sa svim ostalim tehnikama planiranja – listom zahteva, listom rokova, podelom zadataka itd. Ova tehnika nam pomaže da sve resurse rasporedimo optimalno u odnosu na konkretne vremenske rokove koji su nam omogućeni. Na taj način se sama estimacija potrebnog vremena precizira i konkretizuje, te menadžerima ostaje dodatnog prostora i za vreme baferovanja, odnosno višak vremena koje se čuva za krajnje kvalitetnije testiranje proizvoda. Ovde je bitno predvideti sve rokove u maksimalnoj preciznosti te stvoriti jednu celovitu sliku o projektu u datom vremenskom okviru.

## Primena gantovog dijagrama rasporeda poslova (Grantt chart schedule)

Za izradu Gantovog dijagrama za razvoj ovog projekta, korišćen je [https://www.onlinegantt.com/#/gantt](https://www.onlinegantt.com/" \l "/gantt) .

A graph with blue squares

Description automatically generatedPre početka izrade Gantovog dijagrama moraju da se definišu liste poslova I rokova koje su definisane u predhodnim tehnikama i opisane u prethodnim poglavljiva dokumenta. Zadaci se poklapaju za zadacima koje smo definisali, a za vreme je uzeta kolona za najverovatnije vreme.



# Lista budžeta projekta (Project Budget Chart)

## 



## Teoretska postavka

Menadžerima koji su u skladu sa načinom raspolaganja resursima pripada odgovornost planiranja budžeta jednog projekta i tako se dolazi do jednog optimalnog budžetiranja samog projekta. Ukoliko se u jednom trenutku ispostavi da je projekat neisplativ i da neće kompenzovati uloženi novac, on će biti odbačen. U procesu sastavljanja liste budžeta projekta, vrlo je važno uzeti u obzir Gantov dijagram sa listom rokova o kome je gore bilo više reči.

Isto kao i vreme, i troškovi mogu biti optimistični, verovatni, pesimistični i ispostavljeni. Upravo zbog toga se određenom formulom mora doći do takozvane „budžetske kontigencije“ koja podrazumeva krajnji ishod budžeta sa uračunatim onim faktorima koji imaju uticaj na njega. To dalje znači da od svih zadataka treba da oduzmemo polovinu pesimističnih i verovatnih troškova, te da ovu sumu saberemo sa sumom najverovatnijih troškova. Tada dobijamo najpribližniju i najbolju moguću vrednost koja je izračunata prema Goldratt aproksimaciji troškova.

## Primena liste budžete projekta (Project Budget Chart)

Pre predstavljanja budžeta prisetimo se tabele projektovanog vremena za ispunjavanje zadataka.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zadatak | Optimistično vreme | Najverovatnije vreme | Pesimistično vreme | Ispostavljeno vreme |
| Prikupljanje zahteva | 3 | 4 | 5 |  |
| Kreiranje arhitekture sistema | 3 | 4 | 5 |  |
| Kreiranje baze podataka | 1 | 1 | 2 |  |
| Dodeljivanje zadataka | 1 | 1 | 1 |  |
| Implementacija backend-a | 25 | 30 | 35 |  |
| Dizajn korisničkih interfejsa | 3 | 4 | 5 |  |
| Implementacija KI (Web) | 20 | 25 | 30 |  |
| Integraciono tesitranje | 5 | 7 | 10 |  |
| Ispravka grešaka | 5 | 10 | 15 |  |
| Implementacija bezbednostnih mehanizama | 5 | 6 | 7 |  |
| Postavljanje aplikacije na server | 2 | 2 | 3 |  |
| Ukupno | 73 | 94 | 118 |  |

Sledeća tabela predstavlja troškove po zadatku, gde su cene predstavljene u evrima. Kao i kod sati izrade zadataka, cena je predstavljena optimistički, najverovatniji troškovi, pesimistički troškovi I ispostavljeni troškovi, gde će kolona za ispostavljene troškove ostati prazna kako smo još u fazi planiranja projekta. Kako bi se optimalno predvideo budžet projekta, optimalnu sumu budžeta računamo kao pesimistički troškovi minus najverovatniji troškovi pa sve do pomnožimo sa 0.5 i zatim saberemo sa najverovatnijim troškovima.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zadatak | Optimistični troškovi | Najverovatniji troškovi | Pesimistični troškovi | Ispostavljeni troškovi |
| Prikupljanje zahteva | 60 | 80 | 100 |  |
| Kreiranje arhitekture sistema | 60 | 80 | 100 |  |
| Kreiranje baze podataka | 20 | 20 | 40 |  |
| Dodeljivanje zadataka | 20 | 20 | 20 |  |
| Implementacija backend-a | 500 | 600 | 700 |  |
| Dizajn korisničkih interfejsa | 60 | 80 | 100 |  |
| Implementacija KI (Web) | 400 | 500 | 600 |  |
| Integraciono tesitranje | 100 | 140 | 200 |  |
| Ispravka grešaka | 100 | 200 | 300 |  |
| Implementacija bezbednostnih mehanizama | 100 | 120 | 140 |  |
| Postavljanje aplikacije na server | 40 | 40 | 60 |  |
| Ukupno | 1460 | 1880 | 2360 |  |

# Zaključak

**Opisane tehnike u ovom dokumentu su se ispostavile kao krucijalne za rukovođenje projektom i ostale njegove dalje faze. Dosta primenjenih tehnika kao na primer dijagrami su se pokazali kao dobar metod planiranja i uvida u zadatke i funkcionalnosti. Ova dokumentacija konkretno je zaokružila jedan aspekt priče o poslu projektnog menadžera a i o samom upravljanju projektima.**

# Literatura

1. R. Stojić, "SE325 Timski razvoj softvera". Beograd: Univerzitet Metropolitan, 2023.