

*Prolećni semestar, 2022/23*

*PREDMET:*

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

**Razvoj aplikacije pametna kuca**

Domaći zadatak broj 3

Profesor: **Nemanja Zdravković**

Asistent: **Tamara Vukadinovic**

Student: **Uros Milovanovic**

Indeks: **4191**

Sadržaj

[Uvod 3](#_Toc103204088)

[Mrežni plan projekta (Project Network Diagram) 3](#_Toc103204089)

[Teoretska postavka 4](#_Toc103204090)

[Primena mrežnog projekta (Project Network Diagram) 4](#_Toc103204091)

[Zaključak 6](#_Toc103204092)

[Literatura 7](#_Toc103204093)

# Uvod

U ovom dokumentu biće predstavljene tehnike i načini upravljanja projektima razvoja demonstrirane na aplikaciji pametna kuca. U pitanju je sistem koji će olakšati upravljanje kuce krajnjeg korisnika kao i sistem na kome će korisnik moći imati korisnički interfejs po dizajnu koji on izabere. Tehnika upravljanja projektima koju ovaj dokument pokriva jeste **razvoj Mrežnog plana projekta (Project Network Diagram**).

Pomenuta tehnika koristi se u praksi razvoja softverskih rešenja i proizvoda. Čitalac će na kraju ovog dokumenta imati realan uvid u tehnike i veštine neophodne za primenu Mrežnog plana projekta.

Apstrakt aplikacije

Sistem se sastoji od sledecih komponenti. Aplikacija za tablet ili telefon preko koje ce se upravljati ceo sistem. On ce sadrzati sve informacije kao sto su: da li su vreata zakljucana, ukoliko je navijen alarm, da li su zavese zatvorene ili otvorene… Zatim imacemo komponente kao sto su zavesa, muzika – alarm, vrata, prozori. Isto jedna od bitnijih komponenti ce biti asistent koji ce da prepozna glas I uraditi odredjenu komandu ( zakljucati vrata, otvoriti prozor, zatvoriti zavesu). Sistem ce funkcionistati tako sto korisnik udje u aplikaciju koja prikupi sve podatke o komponentama I ponudi korisniku opcije. Na primer zavesa je otvorena korisnik ce imati opciju da je zatvori nece moci da je otvori jer je vec otvorena. Ili korisnik ce imati opciju da udje u kucu I pozove asistenta gde ce aplikacija prepoznati glas I u tom trenutku slusati koju komandu daje korisnik. U slucaju da ta komanda postoji aplikacija ce je izvrsiti u suprotnom ce obavestiti da ne postoji ta komanda.

# Mrežni plan projekta (Project Network Diagram)

## Teoretska postavka

Nakon definisanja konkretnih zadataka i vremena potrebnog za njihovu implementaciju na određenom softverskom proizvodu, važno je isplanirati redosled faza razvoja samog softvera. Iz tog razloga je važno uzeti u obzir međusobne odnose faza razvoja i mere njihove medjuzavisnosti. Zbog toga je važno napraviti Mrežni plan projekta koji definiše prioritete u izradi funkcionalnosti, pogotovo u delovima projekata u kojima se faze razvoja naslanjaju jedna na drugu. Prvenstveno je važno srodne i paralelne zadatke grupisati kroz faze razvoja na takav način da svi zadaci koji od njih budu zavisili budu 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 zasebno, koje se na kraju povezuju u celovit gotov softverski proizvod. Sa tim u vezi, mrežni plan nam daje uvid i u „Kritičnu liniju života“ koja u samom planiranju faza nosi najveću važnost. To je najduža i najkompleksnija linija na dijagramu jednog projekta, od koje esencijalno zavisi ishod samog projekta, odnosno 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 jednakoj meri u kojoj zavisi od liste zadataka. Prilikom kreiranja mrežnog plana, suštinski je važno pronaći i grupisati sve neophodne funkcionalnosti kako bi Kritična linija bila što konkretnije 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.

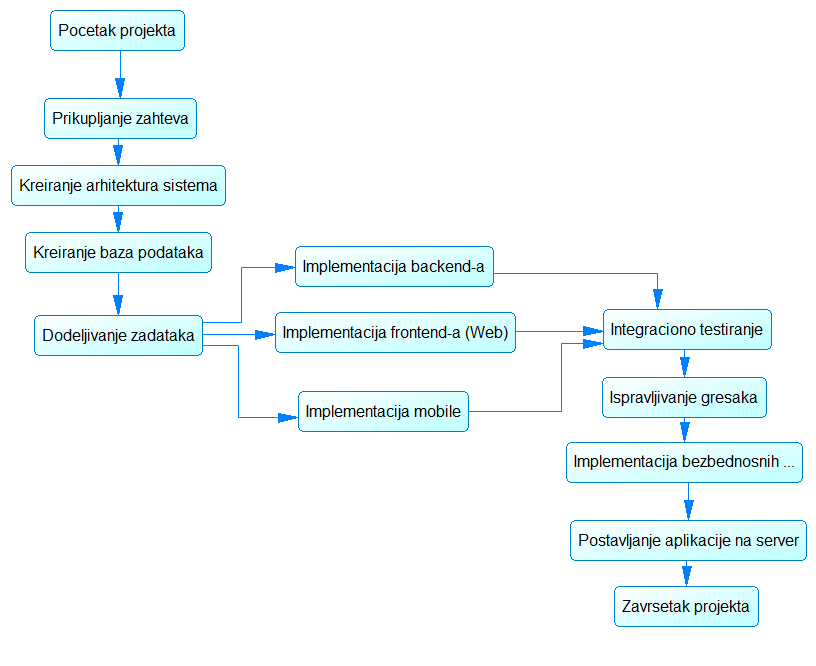
## Primena mrežnog projekta (Project Network Diagram)

Primenom mrežnog plana projekta na ovo mojoj aplikaciji daje nam na uvid koja komponenta zavisi od razvoja drugih, koje komponente se mogu paralelno razvijati kao 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. Implementacija KI (Mobile)
9. Integraciono tesitranje
10. Ispravka grešaka
11. Implementacija bezbednostnih mehanizama
12. Postavljanje aplkacije na server



# Zaključak

Nakon što smo predhodno definisali zadatke i poslove, od jakog značaja je implementirati mrežni plan vođenja projekta kako bismo prezentovali tok izvršenja zadataka na samom projektu. Ovom tehnikom osim što možemo videti koja komponenta zavisi od koje, tj. koj zadatak zavisi od kog, vidimo okvirno vreme izvršavanja projekta, gde npr. Ukoliko smo u fazi implementacije funkcionalnosti pogledom na dijagram tačno možemo videti koje komponente su ostale za realizovati u narednom periodu. Time vodimo tačan plan izrade našeg ravoja samog proizvoda kojim se projekat bavi.