



SE311 PROJEKTOVANJE I ARHITEKTURA SOFTVERA

Školska 2019/2020. godina

PROJEKTNi ZADATAK

Sadržaj

1. Opis projektnog zadatka	2
2. Plan realizacije projekta	5
2.1 Raspored aktivnosti projektnog zadatka po nedeljama:	5

1. Opis projektnog zadatka

Cilj projektnog zadatka je da student bude sposoban da samostalno projektuje softversku arhitekturu (primenom određenih metodologija), modeluje odgovarajuće dijagrame potrebne za prezentovanje softverske arhitekture i pravilno dokumentuje softversku arhitekturu (primenom definisanih pravila).

Zbog toga će student morati da u projektu primeni metodologiju planski vođenog razvoja softverskog sistema. S druge strane, projektni zadatak se radi u sledećim okolnostima:

- Vremensko ograničenje rada studenta na projektu od ukupno 60 sati rada (predmet ima 8 ESPB x 30 sati = 240 sati rada studenta na predmetu, od toga 60 na projektu).
- Svaki projektni zadatak mora da bude specifičan, tj. različit, tako da se ne sme desiti da dva studenta radi isti ili sličan projektni zadatak.
- Tema projekta treba da obuhvati sve faze razvoja projektnog rešenja softvera:
 - definisanje zahteva (funkcionalnih, nefunkcionalnih)
 - definisanje slučajeva korišćenja (sa primarnim i sekundarnim scenarijima)
 - definisanje arhitekture softvera
 - izrada projektnog rešenja softverskog sistema u skladu sa definisanim zahtevima, slučajevima korišćenja i postavljenom arhitekturom.

Imajući postavljen cilj i navedene okolnosti definiše se sledeći projektni zadatak koji važi za svakog studenta: Student treba da sam definiše vrstu softvera i područje njegove primene. Treba da definiše šta se traži od softverskog sistema, koje osnovne funkcije treba da obavi i koje performanse sistem treba da ostvari. To predstavlja opis funkcionalnosti softverskog sistema koji se izražava tekstualno.

Student može koristiti funkcionalne i nefunkcionalne zahteve definisane u projektnom zadatku iz predmeta " SE322 - Inženjerstvo zahteva".

Opisan problem koji softver treba da reši treba da bude primeren planiranom vremenu koji je predviđen za realizaciju projektnog zadatka (60 sati za rad na projektu), ali da bude i dovoljno veliki da ima bar tri glavna slučaja korišćenja.

Pre rada na projektu, student treba da pošalje asistentu OPIS FUNKCIONALNOSTI SOFTVERSKOG SISTEMA na verifikaciju, pre početka rada na projektu. Po dobijanju saglasnosti nastavnika za opis softverskog sistema za koji se radi projekat, student treba da uradi sledeće:

1. Definirati funkcionalne i sistemske (nefunkcionalne) zahteve koje softverski sistem treba da zadovolji. Kao rezultat, treba da se navede spisak svih funkcionalnih i nefunkcionalnih zahteva, a svaki zahtev ukratko opisan i obrazložen razlog za njegovo postojanje. Za izradu ovog zadatka potrebno je koristiti PowerDesigner specifikaciju zahteva.
2. Dati UML dijagrame stanja za sistem u PowerDesigner-u, kao i tekstualni opis primarnih i sekundarnih scenarija (izuzeci) definisanih za svaki slučaj korišćenja. Sistem treba da ima najmanje 3 slučaja korišćenja.
3. Definirati arhitekturu softverskog sistema što treba da obuhvati sledeće:
 - a. Primeniti jednu od tri grupe stilova arhitekture (stil modula, stil komponenta i konektor ili stil alokacije) na softversku arhitekturu.
 - b. Detaljno opisati odabrani stil, identifikovane delove sistema i veze između njih.
 - c. Odabrati i primeniti jedan od sledećih stilova softverske arhitekture:
 - stil razlaganja
 - stil upotrebe
 - stil generalizacije
 - stil slojeva
 - stil aspekta
 - d. Opisati primenjene stilove i identifikovane delove sistema. Obratiti pažnju da odabrani stil na pravi način predstavi funkcionisanje kompletnog sistema koji se projektuje.
 - e. Modelovati dijagrame shodno odabranom stilu softverske arhitekture koristeći pravilnu notaciju stila.
 - f. Primeniti jedan od stilova komponente i konektora
 - stil toka podataka
 - stil poziv-povratak
 - stil ravnopravnih računara

- stil zasnovan na događaju
 - servisno-orijentisani stil
 - stil deljenih podataka
- g. Opisati primenjeni stil i modelovati odgovarajući dijagram softverske arhitekture korišćenjem pravilne notacije odabranog stila
- h. Odgovarajućim dijagramom prikazati kombinovanje različitih pogleda na softversku arhitekturu i detaljno opisati modelovani dijagram
4. Uraditi dve varijante projektnog rešenja:
- a. Primenom gotovih komponenti, koje možete da pronađete, a ostale komponente projektovati u skladu sa definisanim zahtevima
 - b. Primenom servisa, kombinujući raspoložive servise i servise koje ćete razviti.
5. Izvršite detaljno projektovanje dela softverskog sistema za koji ne koristite gotove module, odnosno, raspoložive softverske servise, što obuhvati sledeće:
- a. Sve neophodne detaljne sekvencijalne dijagrame (nivo upotrebljenih klasa).
 - b. Dijagram svih klasa softverskog sistema, zajedno sa njihovim međusobnim vezama.
 - c. Za svaku klasu:
 - i. njen detaljan UML model, sa svim atributima i metodama
 - ii. detaljnu specifikaciju svakog atributa, tipa podatka i opsega mogućih vrednosti
 - iii. detaljan opis svakog metoda, tj. zaglavlja sa svim argumentima, svrhom i opisom metoda. Za svaki metod navesti u sekvencijalnom dijagramu koji sadrži poruku koju realizuje metod.
 - d. Specifikacija jednog ili više interfejsa koji se koriste i njihove veze sa klasama koji realizuju njihove servise.
 - e. Ako je potrebno, modele stanja za pojedine objekte.
 - f. Generisati sve kreirane Power Designer modele i dobiti inicijalni programski kod.
 - g. Dati listinge i opise razvijenih metoda u Javi
 - h. Dati i druge UML dijagrame, ako su potrebni.

U radu na projektu, student može koristiti sledeće SW alate: Power Designer, NetBeans IDE ili Eclipse IDE. Student može da koristi i druge SW alate umesto preporučenih, ako mu to više odgovara, ali uz posebnu dozvolu nastavnika. Student treba da obrazloži predlog za korišćenje drugih alata. Po završetku rada na projektu, student treba da preda Izveštaj o realizaciji projektnog zadatka. Pored toga, student treba da preda i datoteke koje sadrže neophodne podatke sa kojima su radili upotrebljeni SW alati.

Naslov predatih fajlova treba da budu sledeći:

- Za Izveštaj o realizaciji projektnog zadatka: SE311_Ime_Prezime_broj indeksa_Projekat
- Za datoteke koje se prilažu: SE311_Ime_Prezime_broji indeksa_Naziv alata_ident broj (ako ih ima više)

2. Plan realizacije projekta

2.1 Raspored aktivnosti projektnog zadatka po nedeljama:

Prva nedelja: Odabir sistema, odabir vrste softvera i područja njegove primene. Definisanje faza i projektnih aktivnosti unutar procesa projektovanja softvera. Definisanje osnovnih funkcija softverskog sistema i performansi sistema koje moraju biti ostvarene. Tekstualni opis funkcionalnosti softverskog sistema.

Očekivani rezultat nakon prve nedelje: Dokument sa spiskom svih funkcionalnih i sistemskih (nefunkcionalnih zahtevi) uz kratak opis svakog navedenog zahteva.

Druga nedelja: Razumevanje upotrebe softverske dokumentacije za opis softverske arhitekture sistema, definisanje šablona za detaljan opis softverske arhitekture. Razumevanje kombinovanja različitih stilova softverske arhitekture.

Očekivani rezultat nakon druge nedelje: Definisanje šablona koji će biti korišćen za opis softverske arhitekture. Ukoliko student želi, može odabrati stil cevi i filtera za predstavljanje softverske arhitekture odabranog sistema.

Treća nedelja: Odabir jednog stila softverske arhitekture iz stilova softverskih modula. Birati između: stila razlaganja, stila upotrebe, stila generalizacije, stila slojeva i stila aspekta.

Očekivani rezultat nakon treće nedelje: Tekstualni opis primenjenog stila modula softverske arhitekture, opis identifikovanih delova sistema i predstavljanje odgovarajućim dijagramom prateći definisanu notaciju.

Četvrta nedelja: Odabir jednog stila softverske arhitekture iz stilova povezivanja softverskih komponenata. Birati između: stila toka podataka, stila poziv-povratak, stila ravnopravnih računara, stila zasnovanog na događaju, servisno-orijentisanog stila i stila deljenih podataka.

Očekivani rezultat nakon četvrte nedelje: Tekstualni opis primenjenog stila povezivanja softverskih komponenata softverske arhitekture, opis identifikovanih delova sistema i predstavljanje odgovarajućim dijagramom prateći definisanu notaciju.

Peta nedelja: Odabir jednog stila softverske arhitekture iz stilova alokacije. Birati između: stila raspoređivanja, stila instalacije, stila dodeljivanja radnih zadataka.

Očekivani rezultat nakon pete nedelje: Tekstualni opis primenjenog stila alokacije softverskih komponenata softverske arhitekture, opis identifikovanih delova sistema i predstavljanje odgovarajućim dijagramom prateći definisanu notaciju.

Šesta nedelja: Izvršiti kombinovanje različitih pogleda na softversku arhitekturu.

Očekivani rezultat nakon šeste nedelje: Modelovan dijagram koji prikazuje kombinovanje softverske arhitekture i tekstualni opis navedenog dijagrama.

Sedma nedelja: Modelovanje UML dijagrama slučaja korišćenja. Sistem treba da ima najmanje tri slučaja korišćenja.

Očekivani rezultat nakon sedme nedelje: Dijagram slučaja korišćenja sa minimum tri slučaja korišćenja.

Osma nedelja: Modelovati UML dijagrame stanja za odabrani softverski sistem. Tekstualno opisati primarne i sekundarne scenarije (izuzetke) shodno svakom slučaju korišćenja.

Očekivani rezultat nakon osme nedelje: Dijagram stanja softverskog sistema uz tekstualni opis primarnih i sekundarnih scenarija (izuzetaka) za svaki slučaj korišćenja.

Deveta nedelja: Modelovati dijagram projektnog rešenja primenom razvijenih servisa i gotovih servisa (koje možete pronaći) a koje su u skladu sa definisanim funkcionalnim i nefunkcionalnim zahtevima definisanim u prvoj nedelji.

Očekivani rezultat nakon devete nedelje: Modelovani dijagram servisa projektovanog rešenja na kome se nalaze razvijeni servisi za potrebe softverskog sistema a i gotovi servisi koji su pronađeni i koji su u skladu sa definisanim zahtevima.

Deseta nedelja: Modelovati dijagram projektnog rešenja primenom razvijenih komponenti i gotovih komponenti (koje možete pronaći) a koje su u skladu sa definisanim funkcionalnim i nefunkcionalnim zahtevima definisanim u prvoj nedelji.

Očekivani rezultat nakon desete nedelje: Modelovani dijagram komponentata projektnog rešenja sa gotovim komponentama koje je moguće iskoristiti a koje su u skladu sa definisanim zahtevima.

Jedanaesta nedelja: Izvršiti detaljno projektovanje sistema u kome se ne koriste gotove komponente ili servisi kroz: modelovanje detaljnih sekvencijalnih dijagrama do nivoa upotrebljenih klasa.

Očekivani rezultat nakon jedanaeste nedelje: Modelovani sekvencijalni dijagrami do nivoa upotrebljenih klasa i tekstualni opis modelovanih dijagrama.

Dvanaesta nedelja: Izvršiti modelovanje svih klasa softverskog sistema zajedno sa njihovim vezama, atributima i metodama. Uraditi detaljnu specifikaciju svakog atributa, tipa podatka i opsega mogućih vrednosti. Predstaviti detaljan opis svakog metoda i svakog argumenta.

Očekivani rezultat nakon dvanaeste nedelje: Modelovani klasni dijagram sa svim klasama sistema (sa vezama, atributima, metodama i sa detaljnom specifikacijom svakog atributa, tipa podatka i opsega mogućih vrednosti).

Trinaesta nedelja: Izvršiti specifikaciju jednog ili više interfejsa koji se koriste u projektnom rešenju uz opis veza sa klasama koje realizuju njihove servise.

Očekivani rezultat nakon trinaeste nedelje: Detaljna specifikacija identifikovanih interfejsa unutar projektnog rešenja sa vezom sa klasama koje realizuju njihove servise.

Četrnaesta nedelja: Generisanje svih Power Designer modela u cilju dobijanja inicijalnog programskog koda. Preuzeti detaljan listing programskog koda i izvršiti opis razvijenih metoda u Javi.

Očekivani rezultat nakon četrnaeste nedelje: Incijalni programski kod i listing programskog koda sa opisom razvijenih metoda u Javi.

Petnaesta nedelja: Formiranje dokumenta projektnog zadatka (pravilno imenovanje dokumenta) po navedenim poglavljima datim u opisu projektnog zadatka.

Očekivani rezultat nakon petnaeste nedelje: Dokument projektnog zadatka sa svim dijagramima, opisima dijagrama i stavkama koje su navedene u tekstu projektnog zadatka.