

**SE201 – Uvod u softversko inženjerstvo**

**Projektni zadatak**

Pre početka rada na rešavanju zadatka, potrebno je da asistentu koji je zadužen za vežbe vašoj grupi pošaljete apstrakt teme koju želite da obradite. Ukoliko apstrakt nije odobren od strane asistenta, računa se da niste odabrali temu projekta i takav projekat neće biti prihvaćen. Rešenje projektnog zadatka svaki student radi zasebno.

Ukoliko su navedeni kriterijumi zadovoljeni, zakazuje se odbrana projektnog zadatka. Tradicionalni studenti projekat brane počev od 15-te nedelje semestra ili u naknadno definisanim terminima, dok internet studenti mogu odbraniti projekat najkasnije 10 dana pre termina ispita u kome žele da polažu ispit. U slučaju da na odbrani ne pokažete dovoljan nivo znanja, ne umete da objasnite delove projekta ili ste na neki drugi način neuspešni u odbrani projekta, gubite sve poene stečene na projektu. Ukoliko se projekat uspešno odbrani u 15-toj nedelji nastave ili ne kasnije od 15 dana od kraja semestra, na projektu možete maksimalno osvojiti 25 poena.

**Zahtevi za projekat: (Nivo 1)**

1. **Izbor agilnog metodologije**: Izabrati jedan agilni metod (npr. Scrum, Kanban) i pružiti osnovnu razliku između metodologija. Objasniti zašto je izabrana ta metodologija.
2. **Murov obrazac za viziju proizvoda**: Kreirati osnovnu viziju proizvoda koristeći Murov obrazac sa najmanje 3 ključne tačke (misija, vrednosti, resursi).
3. **Spisak funkcionalnih zahteva**: Napisati između 5 i 7 funkcionalnih zahteva koji jasno definišu osnovne funkcije sistema.
4. **Nefunkcionalni zahtevi**: Napisati 3-5 osnovnih nefunkcionalnih zahteva kao što su performanse, bezbednost i skalabilnost sistema.
5. **Korisničke priče**: Napisati između 5 i 7 korisničkih priča sa osnovnim informacijama o potrebama i ciljevima korisnika.
6. **Persona**: Kreirati 3 osnovne Personae sa ključnim informacijama (ime, starost, cilj korisnika).
7. **Scenariji za Persona**: Kreirati po jedan scenarij za svaku Personu u kojem se opisuje tipičan način korišćenja sistema.
8. **Stilovi softverske arhitekture**: Izabrati jedan stil arhitekture (npr. monolit, klijent-server) i dati kratak opis sa prednostima i manama.
9. **Kreacija klasa**: Kreirati između 5 i 7 osnovnih klasa sa atributima i metodama koje obavljaju ključne funkcionalnosti.
10. **Baza podataka i Java logika**: Kreirati osnovnu bazu podataka sa 3-5 tabela i povezati je sa aplikacijom pomoću Java koda.
11. **Ponovno upotrebljive komponente**: Identifikovati 2-3 komponente sistema koje mogu biti ponovo upotrebljene u budućim projektima.
12. **Računarstvo u oblaku**: Identifikovati jednu komponentu sistema koja bi mogla koristiti računarstvo u oblaku (IaaS, SaaS, PaaS).
13. **Mikroservisi**: Identifikovati barem jednu komponentu sistema koja može biti mikroservis sa odvojenom bazom podataka.
14. **Napad Brute Force ili SQL injection**: Demonstrirati osnovni SQL injection napad sa objašnjenjem kako zaštititi sistem od ovog napada.
15. **Atributi kvaliteta sistema**: Navesti 3-5 ključnih atributa kvaliteta sistema, kao što su performanse i sigurnost.
16. **Pouzdano programiranje**: Implementirati osnovne tehnike pouzdanog programiranja kao što su validacija inputa, i osnovni regularni izrazi (RegEx).
17. **Scenariji testiranja**: Kreirati 5-7 jednostavnih test scenarija sa osnovnim koracima i očekivanim rezultatima.
18. **Junit testiranje**: Implementirati osnovne JUnit testove na funkcionalnostima sistema.
19. **Trello alat**: Planirati projekat u Trello sa najmanje 5 zadataka, prateći faze razvoja.
20. **Git verzionisanje**: Verzionariti kod na Git-u sa 3-5 commita tokom razvoja sistema.

### ****Nivo 2:****

1. **Izbor agilnog metodologije**: Izabrati jedan agilni metod (npr. Scrum, Kanban, XP) i pružiti detaljan opis metodologije sa praktičnim primerima. Objasniti kako će metodologija biti implementirana kroz faze projekta.
2. **Murov obrazac za viziju proizvoda**: Kreirati detaljnu viziju proizvoda koristeći Murov obrazac, pokrivajući misiju, vrednosti, ciljeve, interesne grupe i potrebne resurse. Objasniti kako se vizija uklapa u ukupnu strategiju proizvoda.
3. **Spisak funkcionalnih zahteva**: Napisati 10-15 detaljnih funkcionalnih zahteva sa prioritetima i povezanostima između funkcija.
4. **Nefunkcionalni zahtevi**: Napisati 5-7 detaljnih nefunkcionalnih zahteva sa specifičnim vrednostima kao što su vreme odziva, dostupnost, skalabilnost i bezbednost.
5. **Korisničke priče**: Napisati 10-15 korisničkih priča, sa detaljnim opisima korisnika, njihovih ciljeva i scenarija u kojima koriste sistem. Svaka priča treba da sadrži kriterijume prihvatanja.
6. **Persona**: Kreirati 5 detaljnih Personae sa informacijama kao što su motivacija, problemi, ciljevi, ponašanje i specifične potrebe.
7. **Scenariji za Persona**: Kreirati 3-5 scenarija za svaku Personu, obuhvatajući različite situacije i izazove s kojima se susreću korisnici prilikom korišćenja sistema.
8. **Stilovi softverske arhitekture**: Izabrati 2-3 stila arhitekture (npr. mikroservisi, monolit, klijent-server) i pružiti detaljan opis svakog stila, uključujući dijagrame, analize prednosti i mana za vaš sistem.

1. **Kreacija klasa**: Kreirati između 10 i 15 klasa sa detaljnim atributima, metodama i dijagramima klasne strukture koja pokriva sve potrebne funkcionalnosti sistema.
2. **Baza podataka i Java logika**: Kreirati kompleksnu bazu podataka sa 5-7 tabela i implementirati naprednu logiku za povezivanje sa bazom podataka u Javi, koristeći CRUD operacije i optimizaciju upita.
3. **Ponovno upotrebljive komponente**: Identifikovati 5-7 komponenata koje mogu biti ponovo upotrebljene i objasniti kako će te komponente doprineti efikasnosti i održivosti projekta.
4. **Računarstvo u oblaku**: Identifikovati 3-5 komponenti sistema koje mogu koristiti računarstvo u oblaku, detaljno obrazložiti koji tip oblaka je najpogodniji za svaku komponentu i objasniti zašto.
5. **Mikroservisi**: Identifikovati 3-5 komponenata sistema koje bi mogle biti implementirane kao mikroservisi, obuhvatajući prednosti, izazove i arhitektonske dijagrame.
6. **Napad Brute Force ili SQL injection**: Demonstrirati napad Brute Force ili SQL injection na sistemu, analizirati njegovu ozbiljnost i implementirati sigurnosne mere kako bi se sprečili takvi napadi.
7. **Atributi kvaliteta sistema**: Navesti 5-7 atributa kvaliteta, kao što su sigurnost, performanse, skalabilnost i pouzdanost, uz analizu kako svaki atribut utiče na ukupnu funkcionalnost sistema.
8. **Pouzdano programiranje**: Implementirati 3 napredne tehnike pouzdanog programiranja (validacija inputa, šabloni dizajna, napredni regularni izrazi) i dokumentovati sve sa primerima.
9. **Scenariji testiranja**: Kreirati 10-15 test scenarija koji obuhvataju sve ključne funkcionalnosti sistema, sa detaljnim očekivanim rezultatima i testiranjem različitih ulaza.
10. **Junit testiranje**: Implementirati Junit testove koji pokrivaju barem 80% funkcionalnosti sistema, sa detaljnim rezultatima testova i analizom uspešnosti.
11. **Trello alat**: Kreirati detaljan plan projekta u Trello sa svim fazama razvoja, zadacima i rokovima, prateći napredak na nivou tima i pojedinca.
12. **Git verzionisanje**: Verzionariti kod sa 10-15 commit-a tokom razvoja, koristeći grane za različite faze i implementacije, sa dokumentovanim objašnjenjima svakog commit-a.