**Razvoj sistema za proveru znanja iz programiranja**

Ova funkcionalnost se odnosi na razvoj sistema za proveru znanja iz programiranja unutar RAF platforme za učenje programiranja.

**Problemi koje rešava:**

Omogućava studentima polaganje kolokvijuma i ispita iz programiranja.

Pruža nastavnicima alat za pregledanje i ocenjivanje urađenih radova studenata.

**Tehnički Zahtevi:**

Implementacija sistema za skladištenje urađenih zadataka na serverskoj strani.

Integracija klijentske strane sa široko korišćenim razvojnim okruženjima (IntelliJ, Visual Studio Code itd.).

Mogućnost selekcije zadataka prilikom izrade kolokvijuma ili ispita.

Automatsko preuzimanje teksta zadatka sa serverske strane.

Mogućnost unosa i skladištenja koda studenata na serverskoj strani.

Konfiguracija podataka za izradu kolokvijuma ili ispita od strane nastavnika.

Postavljanje tekstova zadataka i definisanje grupa od strane nastavnika.

Pregled, ocenjivanje i unos poena za zadatke studenata od strane nastavnika.

Generisanje zbirnog prikaza osvojenih poena studenata.

Praćenje evidencije o pregledanim zadacima od strane nastavnika.

Slanje rezultata svim studentima putem mejla.

Mogućnost dodavanja komentara nastavnika na radove studenata.

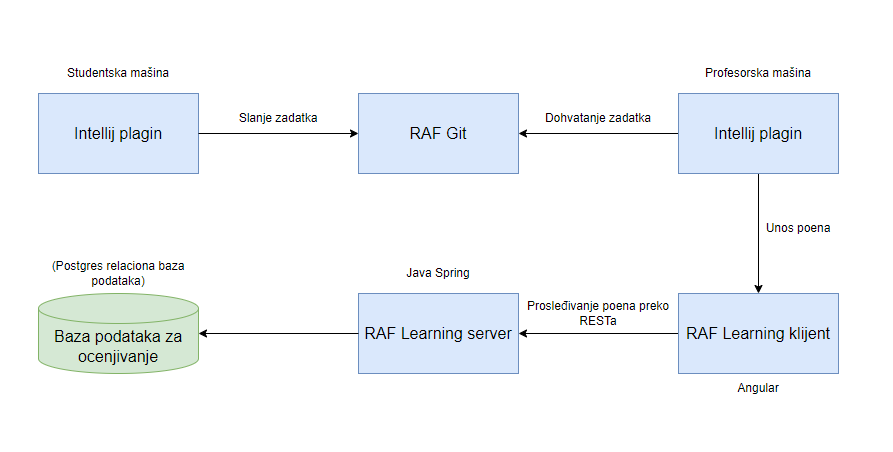
Pregled komentara zajedno sa osvojenim poenima od strane studenata.

**Van Dogovorenog Obima Projekta:**

Izostavljanje implementacije dodatnih funkcionalnosti koje nisu direktno vezane za proveru znanja.

Planiranje za budući rad na unapređenju sistema, kao što je dodavanje novih opcija za komunikaciju između nastavnika i studenata.

**Diagram rešenja:**



Ova funkcionalnost će se isporučiti kroz dva glavna dela: serversku stranu za skladištenje i upravljanje zadacima, i klijentsku stranu integrisanu sa razvojnim okruženjima za kreiranje i predaju radova.

**Detaljan Dizajn:**

Implementacija serverske strane za skladištenje zadataka pomocu RAF Git-a.

Implemetacija skladištenja podataka o broju bodova i prisustvu ispitima pomocu Postgres relacione baze podataka.

Integracija klijentske strane sa postojećim razvojnim okruženjima pomocu razvoja plugina za Intelij platformom.

Dodavanje funkcionalnosti za unos, pregled, i ocenjivanje radova studenata pomocu RAF Git-a i RaF learning platforme koja ce biti implementirana pomocu Angular.js-a na klijentskoj strani i Java Spring-a na serverskoj strani.

**Skladištenje Podataka:**

Korišćenje RAF Git-a za skladištenje zadataka, korišćenje baze podataka za rezultate ispita i informacija o studentima.

Struktura podataka za čuvanje informacija o zadacima, studentima i ocenama - **TODO**.

**Performance:**

Optimizacija sistema za brzo preuzimanje i skladištenje zadatka.

Proaktivna evaluacija performansi sistema tokom izrade kolokvijuma ili ispita.

**Infrastruktura, Rast, i Skaliranje:**

Korišćenje postojeće infrastrukture za serversku stranu.

Planiranje skalabilnosti sistema kako bi se nosio sa porastom broja studenata i nastavnika.

**Troškovi:**

Troškova infrastrukture za skladištenje i obradu zadataka nece biti od znacaja posto se planiran korišćenje postojeće infrastrukture.

**Pouzdanost, Otpornost i Ispravnost:**

Implementacija mehanizama pouzdanosti sistema za čuvanje ocena i zadataka pomocu durabilne memorije koja ce biti koriscenja od strane baze podataka i RAF git-a.

Proaktivna identifikacija i rešavanje potencijalnih problema u vezi sa ispravnošću ocena pomocu korisničkih naloga.

**Bezbednost:**

Identifikacija i implementacija sigurnosnih mera zaštite podataka studenata i zadataka pomocu korisničkih naloga i JWT tokena, firewall-a, i rate limiting-a.

**Implementacija:**

Postepeno uvođenje funkcionalnosti u produkciju kako bi se izbegli prekidi za korisnike.

Upotreba mehanizama za rollback u slučaju problema tokom implementacije.

**Opservabilnost, Metrike, Logovanje, i Praćenje:**

Praćenje metrika performansi i upozorenja na odstupanja **- TODO (Izbor alata i integracije)**

Logovanje relevantnih informacija za pregled i rešavanje problema. **- TODO (Izbor alata i integracije)**

**Operabilnost, Održavanje, Podrška, Debugovanje:**

Održavanje jednostavnosti koda radi lakšeg razumevanja i rešavanja problema.

Pisanje servisnog opisa i priručnika za rad sa novom funkcionalnošću.

Pružanje alatki i uvida za podršku i održavanje sistema.

**Test Strategija:**

Implementacija raznovrsnih testova (unit, integration, end-to-end) kako bi se garantovala kvalitetna isporuka sistema.

Sprovođenje testiranja u različitim okruženjima (lokalno, staging, produkcija).

**Faze implementacije:**

POC (Proof of concept), zatim kontinuirani razvoj po kratkim fazama do potpune funkcionalnosti.