Home

AnamarijaKic edited this page on Nov 5, 2024 · 7 revisions

Programsko inženjerstvo ak. godina 2024/2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

StudyBuddy

Grupa: <TG12.2>

Tim:

- Erik Kranjec
- Mila Podrug
- Tena Osredečki
- Darian Begović
- Karlo Mezdić
- Ella Grković
- Anamarija Kic

Nastavnik: Vlado Sruk

Naziv tima: StudyBuddy

Pages 13

Find a page…

Home
- Programsko inženjerstvo ak. godina 2024/2025
  - Sveučilište u Zagrebu
  - Fakultet elektrotehnike i računarstva
- StudyBuddy

1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj…
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a…
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

1. Opis projektnog zadatka

Tena Dseredžki edited this page 2 hours ago · 18 revisions

Mnogi učenici postaju previše ovisni o digitalnim uređajima, što može dovesti do problema s koncentracijom i smanjenjem sposobnosti za duboko razmišljanje. Također, određena gradiva i predmeti učenicima mogu djelovati odbojno i pretjerano zahtjevno zbog čega gube interes za redoviti rad i svladavanje gradiva jer ano djeluje nemoguće za savladati. Neki učenici imaju poteškoća s organizacijom vremena i zadataka, što može dovesti do odgađanja obaveza, kašnjenja u učenju i mogućnosti zbog nedostatka postignuća. Svi gore navedeni problemi pridonose povećanju stresa zbog akademskih zahtjeva, pritiska vela i sezi s ispitima, društvenih očekivanja i neizvjesnosti u budućnosti. Anksioznost i depresija postaju sve češći problemi među mladima. Također, ne pružaju niti sve škole jednako adekvatan način poučavanja niti podršku svojih nastavnika, što im može otežati razumijevanje gradiva i ispitivanje problema na vrijeme. Zbog ovih problema i nedostataka osmišljena je platforma StudyBuddy.

StudyBuddy je platforma u obrazovanju učenika i studenata s profesorima, te međusobno jednih s drugima, u svrhu boljeg stjecanja znanja. U današnjem modernom svijetu gdje nam pažnju često uzimaju razne internetske zanimacije, StudyBuddy platforma, između ostaloga, pruža mogućnost okupljanja studenata i učenika u grupe za učenje – „StudyGroup“-e. To su grupe u kojima učenici i studenti jedni druge međusobno motiviraju na učenje i pomažu jedni drugima u svladavanju gradiva. Također, platforma omogućuje učenicima i studentima komunikaciju s profesorima koji im mogu pružiti svoje vlastite radne materijale, pružiti takozvane instrukcije "jedan na jedan", te omogućiti sudjelovanje na "masovnim instrukcijama" koje organiziraju profesori. Masovne instrukcije su instrukcije u kojima sudjeluju profesor i veći broj učenika ili studenata, odnosno slične su klasičnom predavanju u učionici.

Platforma ima početnu web-stranicu koja korisnicima omogućava jednostavan pristup opcijama za registraciju ili prijavu. Korisnicima je omogućena prijava preko njihovog Google računa ili upisivanje maila nekog drugog pružatelja mail usluge. Radi sigurnosti korisnika, od novih korisnika se traži da kreiraju korisničko profil, dok postojeći korisnici prijavom mogu lako pristupiti svom korisničkom računu. Prilikom kreiranja novog računa korisnici moraju odabrati rade li registraciju kao učenici, studenti ili profesori te unijeti sljedeće podatke:

• Studenti i učenici: ime, prezime, e-mail, lozinka, slika profila, dob, spol, mjesto, grad stanovanja i školovanja i opis

• Profesori dodatno unose: stručnost, cijena repeticija i mjesto rada

Neregistrirani korisnik vidi samo početnu stranicu i stranicu za prijavu i registraciju. Korisnici koji su već registrirani, prijavljuju se u sustav s postojećim e-mailom i lozinkom. Ako korisnik prilikom unosa podataka za prijavu unese podatke koji ne odgovaraju nijednom registriranom korisniku u bazi, šalje mu se obavijest o neispravnosti podataka. Ako je korisnik zaboravio lozinku, omogućen mu je oporavak lozinke putem e-mail adrese.

Prijavljeni i registrirani korisnici također imaju svoje profile koje mogu uređivati odabirom opcije „Uredi profil“. Pri uređivanju profila provjerava se jesu li novi podatci uneseni u ispravnom formatu, ako nisu korisnik dobiva obavijest o neispravnosti. Nakon unosa promjena, korisnik mora odabrati opciju "Spremi promjene" kako bi potvrdio potvrđivanje promjena u bazu podataka. Profil profesora sadrži osnovne podatke o profesoru: ime i prezime, spol, grad u kojem živi i grad u kojem radi, kontakt te recenziju koju je dobio od učenika ili studenata na temelju prethodno održanih instrukcija i materijala koje je postavio. Profesori na platformi imaju mogućnost objavljivanja poziva na "masovne instrukcije". U svakom pozivu profesori definiraju vrijeme i mjesto instrukcija, gradivo koje će se na instrukcijama obrađivati, uzrast kojem su namijenjene instrukcije, maksimalan broj polaznika, trajanje instrukcija te dokumente (prezentacije, tekstualne i pdf dokumente) koje je potrebno proći prije dolaska na instrukcije. Postoji rok do kojega se mora prijaviti minimalan broj ljudi kako bi se instrukcije održale (datum roka postavlja profesor).

Učenici i studenti imaju svoje javne profile koji uključuju profilnu sliku, ime i prezime, spol, dob, grad u kojem žive i grad u kojem studiraju, uspjehe i vlastiti opis samoga sebe. Oni također imaju mogućnost formiranja "oglasa" za grupe za učenje. Grupe za učenje kreirale bi se na temelju vremena i mjesta održavanja, područja i predmeta za čije su učenje namijenjene, maksimalnog broja članova koji im se mogu pridružiti, troškovima koje bi zbog takvog okupljanja mogli imati članovi (plaćanje prostora, hrane, materijala, ...) - samo informativno, način plaćanja određuju inicijatori grupa u razgovoru s ostalim članovima) te u ovakve grupe učenja, koje su organizirali sami učenici ili studenti, profesor ne bi imao pristup. Drugi učenici/studenti mogu se prijaviti na sudjelovanje u ovakvim zajedničkim sesijama učenja, a inicijator određene grupe za učenje ima mogućnost određivanja maksimalnog broja dolaznika, mjesta i lokacije sastanka.

Prijavljeni korisnici mogu pretraživati aktivne oglase za masovne instrukcije ili grupe za učenje na temelju ranije navedenih karakteristika koje one posjeduju i u cilju čega su stvorene. Također, prijavljenim korisnicima omogućena je opcija komunikacije sa svim drugim korisnicima. Učenici i studenti mogu se javljati jedni drugima putem razmjene poruka (Cavrljanjem). Isto tako putem poruka mogu se javiti profesorima s pitanjima u vezi dodatnih informacija o "masovnim" instrukcijama ili dogovaranju instrukcija jedan na jedan.

Korisnici imaju mogućnost označiti svoj profil kao neaktivan ukoliko odluče da im platforma trenutno nije potrebna. Za vrijeme neaktivnosti, komunikacija s tim korisnicima nije moguća, a profil se automatski ponovno aktivira nakon unaprijed definiranog roka. Osim ispravljanja isključenja, profil korisnika može biti deaktiviran i od strane administratora zbog neprikladnog ponašanja. Korisnici mogu prijaviti drugog korisnika ako uoče neprikladno ponašanje koje mogu i dokazati. Prijava uključuje obrazloženje, koje se šalje administratoru na pregled. Administrator sustava nadzire aktivnost svih korisnika i pregledava pristigle prijave. Nakon analize, administrator može odbaciti prijavu kao neosnovanu ili trajno deaktivirati profil prijavljenog korisnika, ovisno o težini prekršaja.

Ova aplikacija bila bi neprocjenjiv alat za sve učenike i studente suočene s akademskim izazovima. Omogućila bi brzo povezivanje i koordinaciju među vršnjacima s ciljem zajedničkog učenja, čime bi se poticala motivacija i usredotočenost na ostvarivanje ciljeva. Sastanci za učenje i rad u grupama pružili bi učenicima osjećaj podrške i zajednštva, što je često ključno za prevladavanje osjećaja izolacije, nesigurnosti ili frustracije koje nastaju kada je gradivo teško i obimno. Na ovaj način, aplikacija bi služila kao digitalni prostor koji okuplja ljude sa sličnim ciljevima, promovirajući ne samo akademski napredak nego i razvijanje socijalnih vještina kroz interakciju s vršnjacima. Osim što bi pomogla učenicima i studentima, aplikacija nudi profesorima priliku za dodatni izvor prihoda te im omogućuje da svoje znanje prenesu na veću i raznovrsniju skupinu učenika i studenata. Profesori kroz aplikaciju mogu doseći širi krug ljudi, bez geografskih ograničenja, te tako povećati svoj utjecaj i pomoći većem broju učenika koji traže dodatne poduke i objašnjenja.

Nadogradnja navedene platforme bila bi dodavanje značajki poput kalendara, podsjetnika i planera učenja, što bi korisnicima omogućilo bolju organizaciju i praćenje napretka. Ovi alati bi korisnicima pomogli da pravovremeno pripreme gradivo, postave jasne ciljeve i ostanu organizirani tijekom akademske godine. Pretvaranjem web aplikacije u mobilnu aplikaciju korisnicima bi bilo omogućeno jednostavnije korištenje i pristup funkcionalnostima u pokretu. Mobilna verzija bi omogućila i dodatne funkcionalnosti prilagođene korisnicima koji preferiraju učenje na pametnim telefonima, čime bi se povećala popularnost i praktičnost aplikacije. Za ostale nadogradnje možemo se ugledati na identične projekte i stranice pa, primjerice, implementirati pregled svih lokacija srodnih oglasa kao što je prikazano na slici 1.6. u nastavku. Ta bi funkcionalnost olakšala korisnicima prikaz instrukcija i grupa za učenje u blizini, bez potrebe za pregledom svakog oglasa zasebno.

Od sličnih stranica postoji Pascal. Pascal je stranica napravljena s ciljem da olakša studentima Ekonomskog fakulteta studiranje i svladavanje gradiva. Prijavljeni studenti podijeljeni su u manje radne skupine koje omogućuju individualni pristup svakom studentu čime je fokus učenja usmjeren na gradivo koje je polaznicima teško razumljivo. Stranica nudi učenje gradiva iz čak osam predmeta.

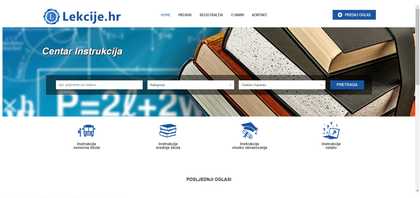
Slika 1.1. Naslovna slika Pascal stranice

Slika 1.2. Popis predmeta koje Pascal stranica nudi

Iako je Pascal sličan ideji naše aplikacije, Pascal je specijaliziran za gradiva koja se prolaze na Ekonomskom fakultetu, dok bi naša platforma pokrivala puno veće područje gradiva iz raznih predmeta. Ponekad je profesorima, ili učenicima i studentima, lakše objašnjavati jedno gradivo nego drugo, naša bi aplikacija omogućila da učenici i profesori instrukcije održavaju samo vezano uz gradivo koje najviše uživaju poučavati bez da su forsirani na prolazak gradiva cijelog polugodišta, odnosno semestra. Također, pružatelji instrukcija nerijetko risu najbolje reklamirani, ova bi im platforma omogućila veliku vidljivost među korisnicima koji potražuju instrukcije. Osim profesora, učenici i studenti bi također dobili priliku objasniti gradivo jedni drugima i time steći prijateljstva i kontakte koji kroz zajedničko učenje često postanu dragocjeni i trajni.

Slijede još dva primjera sličnih stranica - Instrukcije-hr.com te Lekcije.hr.

Lekcije.hr je stranica koja objedinjuje lekcije i instrukcije više kategorija (osnovna ili srednja škola, visoko obrazovanje...) te na sličan način prikazuje aktivne oglase na početnoj stranici, s pripadajućim opisom, lokacijom i cijenom. Klikom na oglas otvara se nova stranica s detaljnijim opisom i opcijom recenziranja.
Obje stranice, nalik našoj, nude opciju filtriranja i pretraživanja postojećih oglasa te opciju stvaranja vlastitog oglasa. Na obje je stranica omogućena prijava, no ne i izravno preko Google poslužitelja.

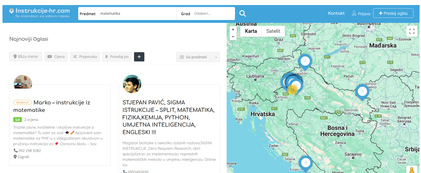


Slika 1.3. Početna stranica Lekcije.hr

Slika 1.4. Početna stranica Lekcije.hr - ponuđeni oglasi

Instrukcije-hr.com platforma je slične namjene i sličnih funkcionalnosti. Oglasi se pretražuju po predmetu i gradu održavanja. Nakon pretraživanja, prikazuju se dostupni oglasi te karta Hrvatske s istaknutim lokacijama odgovarajućih instrukcija.

Slika 1.5. Početna stranica Instrukcije-hr.com



Slika 1.6. Instrukcije-hr.com - oglasi vezani uz matematiku

Naziv tima: StudyBuddy

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Pages 13
Find a page ...
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj...
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a...
8. Upute za pušćanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

2. Analiza zahtjeva

Tena Osredečki edited this page 6 hours ago · 31 revisions

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-001	Sustav omogućuje korisnicima kreiranje računa pomoću e-mail adrese.	Visok	Zahtjev dionika	Korisnik se može registrirati e-mailom, primiti e-mail za potvrdu i uspješno se prijaviti.
F-002	Sustav omogućuje prijavu i odjavu korisnika u, odnosno iz web-aplikacije.	Visok	Zahtjev dionika	Korisnik se, nakon izrade korisničkog računa, može neograničen broj puta prijaviti i odjaviti iz sustava.
F-003	Sustav omogućuje korisnicima brisanje računa.	Srednji	Zahtjev dionika	Korisnički se podaci na zahtjev za brisanje brišu iz baze podataka. S istom se e-mail adresom može stvoriti novi račun.
F-004	Sustav omogućuje korisnicima deaktivaciju računa.	Nizak	Zahtjev dionika	Korisnički se podatci na zahtjev za deaktivaciju ne brišu iz baze podataka i nije moguće stvoriti novi račun s istom e-mail adresom. Korisnik može predati zahtjev za ponovnu aktivaciju računa.
F-005	Sustav omogućuje odabir uloge korisnika.	Visok	Zahtjev dionika	Korisnik može birati između uloga učenika, studenta i profesora, odabir mu je vidljiv na profilu i otvara adekvatne mogućnosti daljnjeg korištenja web-stranice.
F-006	Sustav omogućuje unos, izmjenu i brisanje podataka korisnika.	Visok	Zahtjev dionika	Korisnik može na svojem profilu unijeti, mijenjati, vidjeti i obrisati određene podatke o imenu, prezimenu, e-mail adresi, slici profila, dobi, spolu, mjestu stanovanja, mjestu školovanja, opisu sebe, ...
F-007	Sustav omogućuje učenicima i studentima davanja recenzija o profesorima.	Nizak	Dokument zahtjeva	Učenik/student može dati recenziju o profesoru čijim je instrukcijama prisustvovao, recenzija je vidljiva na profilu profesora.
F-008	Sustav omogućuje učenicima i studentima kreiranje "StuddyGroup"-e.	Visok	Dokument zahtjeva	Učenik/student može kreirati grupu za učenje, te učenik/student koji je kreator grupe jedini može odrediti osnovna obilježja kreirane grupe (poput mjesta održavanja, vremena održavanja, tematike sastajanja i sl.) i objaviti potrebne materijale.
F-009	Sustav omogućuje učenicima i studentima koji nisu kreatori određene "StuddyGroup"-e da joj se pridruže.	Visok	Dokument zahtjeva	Učenik/student koji nije kreirao grupu može se samo pridružiti grupi, ne može odrediti osnovna obilježja grupe, ali može objavljivati materijale.
F-010	Sustav omogućuje učenicima i studentima da se pridruže "Masovnim instrukcijama" po njihovom izboru.	Visok	Dokument zahtjeva	Učenik/student može se prijaviti za sudjelovanje na aktivne "Masovnim instrukcijama" koje njemu odgovaraju u kojima nije popunjen maksimalni broj sudionika.
F-011	Sustav omogućuje učenicima i studentima zatražiti od profesora instrukcije "Jedan na jedan".	Nizak	Zahtjev korisnika	Učenik/student može profesoru poslati zahtjev za instrukcije "Jedan na jedan" s relevantnim podatcima vezanim uz te instrukcije.
F-012	Sustav omogućuje profesorima prihvaćanje ili odbijanje zahtjeva učenika/studenata za instrukcije "Jedan na jedan".	Nizak	Zahtjev korisnika	Profesor može prihvatiti ili odbiti zahtjev za instrukcije "Jedan na jedan", sve daljnje dogovore nakon ove odluke profesora, profesor i učenik/student vode preko komunikacijskog kanala.
F-013	Sustav omogućuje profesorima kreiranje "Masovnih instrukcija".	Visok	Dokument zahtjeva	Profesor može kreirati instrukcije za veći broj učenika/studenata, te odrediti osnovna obilježja kreirane grupe i objaviti potrebne materijale.
F-014	Sustav omogućuje međusobnu komunikaciju korisnika.	Srednji	Postojeći sustav	Korisnici mogu započeti i nastaviti komunikaciju međusobno.
F-015	Sustav omogućuje učenicima i studentima pretragu svih aktivnih instrukcija i grupa za učenje.	Visok	Postojeći sustav	Učenik/student može pretraživati instrukcije i/ili grupe pomoću filtracije po vlastitim preferencijama. Dobiva listu aktualnih instrukcija.
F-016	Sustav omogućuje korisnicima prijavu drugih korisnika koji se neprimjereno ponašaju.	Srednji	Postojeći sustav	Korisnik može podnijeti prijavu prema drugom korisniku koju kasnije revidira administrator. Prijava se podnosi u slučaju neprimjerenog ponašanja korisnika.
F-017	Sustav omogućuje administratoru sustava da upravlja aktivnošću profila.	Nizak	Dokument zahtjeva	Administrator sustava može deaktivirati, obrisati i ponovno aktivirati korisnički račun ovisno o zahtjevu kojega dobi.
F-018	Sustav omogućuje administratoru sustava da pregledava i upravlja sadržajem svih korisnika.	Srednji	Dokument zahtjeva	Administrator sustava može pregledavati i ukloniti sadržaje bilo kojeg korisnika ukoliko je sadržaj neprimjeren ili prijavljen.
F-019	Sustav omogućuje pohranu u bazu podataka svih podataka o korisnicima, njihovim ograničenjima i mogućnostima, instrukcijama i grupama za učenje.	Visok	Postojeći sustav	Baza sustava pohranjuje sve relevantne podatke koji su određeni za pohranu.
F-020	Sustav omogućuje korisnicima kreiranje računa pomoću njihovog postojećeg Google računa	Visok	Zahtjev dionika	Korisnik se može registrirati postojećim Google računom, primiti e-mail za potvrdu i uspješno se prijaviti.

Zahtjevi za održavanje

> NF = ne-funkcijski zahtjevi
> NF 3.1 = zahtjevi performansi
> NF 3.1.7 = zahtjevi povezani s dokumentacijom
> NF 3.2 = zahtjevi sigurnosti
> NF 3.4 = zahtjevi skalabilnosti
> NF 3.7 = zahtjevi upotrebljivosti
> NF 3.8 = zahtjevi dostupnosti

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-3.1.1	Vrijeme učitavanja stranica i odgovora sustava na zahtjeve korisnika mora biti manje od 2 sekunde.	Visok
NF-3.1.2	sustav mora podržavati barem 100 istovremenih korisnika bez značajnog pada performansi.	Visok
NF-3.1.7	Sustav treba imati dovoljnu dokumentaciju.	Visok
NF-3.2	Osjetljivi podaci korisnika moraju biti zaštićeni pristupnim kontrolama i dostupni samo ovlaštenim korisnicima.	Visok
NF-3.4	Web stranica mora biti lako proširiva kako bi podržala rast broja korisnika i dodatnih funkcionalnosti.	Visok
NF-3.7	Korisničko sučelje mora biti intuitivno i jednostavno za navigaciju, posebno za učenike bez tehničkog iskustva.	Visok
NF-3.7.1	Web stranica mora imati responzivan dizajn i biti prilagođena za mobilne uređaje.	Srednji
NF-3.8	Sustav mora biti dostupan 99,9% vremena (visoka dostupnost).	Srednji
NF-3.8.1	Redovito održavanje mora biti zakazano u neaktivnim satima kako bi se minimaliziralo vrijeme nedostupnosti	Srednji

Dionici

Popis dionika:

1. Učenici/studenti
2. Profesori
3. Administratori sustava
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Učenik/student (inicijator) može:
   - odabrati je li učenik ili student (F-005)
   - unijeti osnovne podatke profila (F-001,F-006)
     - ime, prezime, email, dob, spol, mjesto stanovanja, mjesto školovanja, školovanje, dati vlastiti opis
     - sliku profila koju je moguće staviti, promijeniti i obrisati
   - unijeti email i lozinku da bi se prijavili (F-002)
   - odjaviti se s platforme (F-002)
   - obrisati svoj račun (F-003)
   - deaktivirati svoj račun (F-004)
   - pregledati i uređivati svoje podatke (F-006)
   - davati recenzije o profesorima čijim su instrukcijama prisustvovali (F-007)
   - kreirati grupe za učenje - "StudyGroup"-e (u ovu vrstu grupe profesori nemaju pristup) (F-008)
     - kreator grupe određuje mjesto, vrijeme, datum, maksimalan broj članova i gradivo radi čijeg učenja se okupljaju
     - prijavljeni u grupu mogu u nju stavljati materijale (tekst, dokumenti, slike, itd.)
   - pridružiti se postojećoj "StudyGroup"-i (F-009)
   - pridružiti se "masovnim instrukcijama" (F-010)
   - zatražiti od profesora instrukcije "jedan na jedan" (F-011)
   - komunicirati s drugim učenicima/studentima (F-014)
   - komunicirati s profesorima (F-014)
   - pretraživati objavljene instrukcije (F-015)
     - na temelju vrste instrukcija, gradiva, uzrasta, predmeta, cijene, datuma, vremena, lokacije (ako je fizički susret)
     - prijaviti druge korisnike (F-016).
2. Profesor (inicijator) može:
   - obaviti registraciju kao profesor (F-005)
   - unijeti osnovne podatke profila (F-001, F-006)
     - ime, prezime, email, spol, mjesto stanovanja, mjesto rada, stručnost, cijenu repeticija
     - sliku profila koju je moguće staviti, promijeniti i obrisati
   - unijeti email i lozinku da bi se prijavili (F-002)
   - odjaviti se s platforme (F-002)
   - obrisati svoj račun (F-003)
   - deaktivirati svoj račun (F-004)
   - pregledati i uređivati svoje podatke (F-006)
   - organizirati "masovne instrukcije" s učenicima/studentima (F-013)
     - određuje mjesto, vrijeme, datum, trajanje, gradivo, maksimalni broj sudionika, cijenu
     - odrediti krajnji rok za prijavu i minimalni broj sudionika koji se mora prijaviti da bi se instrukcije održale
     - staviti materijale za instrukcije
   - imati instrukcije "jedan na jedan" sa studentima (F-012)
   - komunicirati s učenicima/studentima (F-014).
3. Administrator sustava (inicijator) može:
   - upravljati aktivnošću profila - gasi profil privremeno ili trajno (F-017)
   - pregledati profile i objave prijavljenih korisnika te ukloniti neprimjerene objave, ukloniti profile korisnika s neprimjerenim objavama (F-017, F-018)
   - nakon što korisnik prijavi drugog korisnika, razmotriti prijavu i po potrebi ukloniti profil prijavljenog korisnika (F-017).
4. Baza podataka (sudionik) može:
   - pohraniti sve podatke o korisnicima i njihovim mogućnostima (F-019)
   - pohraniti sve podatke o instrukcijama i grupama za učenje (F-019).

Naziv tima: StudyBuddy

Pages 13
Find a page...
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
Funkcionalni zahtjevi
Zahtjevi za održavanje
Dionici
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj...
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a...
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

```
https://github.com/savipauk/study
```

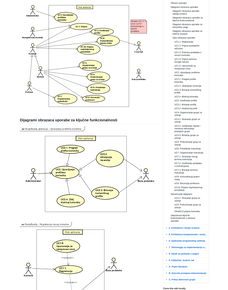
3.-Specifikacia zadania tvorivej skupiny

Last commit message and date not found. 2 months ago

Obrazovi uporabe

Diagram obrazovej uporabe

Diagram obrazova uporaba


Obrazovi uporabe
Obrazovi uporabe
Obrazovi uporabe

4. Architektura / design systému

4. Architektura / design systému · Edit · Delete

Architektura systému

Opis architektury

Cíl architektury

Hlavním cílem tohoto dokumentu je definovat celkovou strukturu a principy fungování systému, aby bylo zajištěno jasné oddělení zodpovědností, snadná rozšiřitelnost a udržovatelnost systému. Architektura musí podporovat modularitu, bezpečnost a výkonnost. Dále by měla, s ohledem na plánované rozšíření, umožnit snadné přidávání nových funkcí bez zásahu do stávajícího kódu. Cílem je vytvořit robustní základ, který bude schopný dlouhodobého vývoje a integrace nových technologií.

Publikování

Publikování probíhá ve dvou fázích:

- **Automatizovaná publikace:** Automatizovaná publikace, spuštěná každou noc, zajišťuje aktuálnost obsahu.
- **Publikace na požádání:** Ruční publikace, kterou lze kdykoliv spustit v rámci administrace, umožňuje okamžitou aktualizaci obsahu.

Frontend a Backend

- **Frontend:** Frontend je zodpovědný za vykreslování uživatelského rozhraní a zajišťuje interaktivní prvky pro uživatele. Komunikuje s backendem prostřednictvím API a zajišťuje responzivní design pro různé typy zařízení.
- **Backend:** Backend je zodpovědný za zpracování dat, ukládání do databáze a poskytování API pro frontend. Zajišťuje bezpečnostní mechanismy, validaci dat a efektivní zpracování požadavků. Backend by měl být navržen tak, aby byl snadno rozšiřitelný a udržovatelný.

Vícejazyčnost

Systém podporuje vícejazyčnost, umožňuje tedy vytvářet a spravovat obsah v různých jazycích. Tento princip zahrnuje strukturu databáze, uživatelské rozhraní i API.

Oddělení uživatelských rolí

Systém bude disponovat systémem oprávnění a přístupových rolí. Tento princip zajišťuje, že každý uživatel bude mít přístup pouze k těm funkcím a datům, ke kterým je oprávněn. Systém by měl podporovat minimálně tyto základní role: Administrátor, Redaktor, Uživatel.

Bezpečnostní principy systému

Systém zahrnuje mechanismy pro ochranu proti běžným bezpečnostním hrozbám, jako jsou SQL injekce, XSS útoky, CSRF útoky a neoprávněné přístupy. Veškerá komunikace mezi frontendem a backendem by měla probíhat bezpečně, s ohledem na citlivost přenášených dat.

Obecné principy systému

- **Modularita:** Systém by měl být navržen tak, aby byl snadno rozšiřitelný a udržovatelný. Jednotlivé komponenty by měly být co nejvíce nezávislé, aby bylo možné provádět změny bez ovlivnění celého systému.
- **Bezpečnost:** Bezpečnostní principy zahrnují ochranu proti běžným útokům, jako jsou SQL injekce, XSS útoky, CSRF útoky a neoprávněné přístupy. Veškerá komunikace mezi frontendem a backendem by měla probíhat bezpečně, s ohledem na citlivost přenášených dat.
- **Výkonnost:** Systém by měl být navržen tak, aby byl schopný zvládat očekávanou zátěž. Důraz je kladen na optimalizaci databázových dotazů a efektivní zpracování požadavků.
- **Vícejazyčnost:** Systém podporuje vícejazyčnost, umožňuje tedy vytvářet a spravovat obsah v různých jazycích. Tento princip zahrnuje strukturu databáze, uživatelské rozhraní i API.

Objektově orientovaná architektura

Systém bude vyvíjen v souladu s principy objektově orientované architektury. Tyto principy zahrnují dědičnost, polymorfismus, zapouzdření a abstrakci. Objektově orientovaná architektura umožňuje snadnou rozšiřitelnost a udržovatelnost systému. Dále umožňuje efektivní znovupoužití kódu a zjednodušuje testování jednotlivých komponent systému.

Organizace systému na vrstvy/části

Systém bude rozdělen do jednotlivých vrstev/částí, konkrétně: Backend, Frontend, databázová vrstva (ORM) a API. Tento princip zajišťuje oddělení zodpovědností, lepší udržovatelnost, možnost nezávislého vývoje jednotlivých částí systému, snadnější testování a možnost nasazení jednotlivých částí systému nezávisle na sobě.

Organizace kódu na vrstvy/části

Jednotlivé vrstvy/části systému by měly mít svoji strukturu kódu. Jednotlivé vrstvy/části zahrnují:

- **Backend:** Backend by měl mít strukturu kódu, která bude podporovat oddělení jednotlivých částí systému, jako jsou modely, pohledy, šablony, formuláře, URL adresy, administrační rozhraní a pomocné soubory. Tento princip zajišťuje lepší udržovatelnost, snadnější testování a možnost znovupoužití kódu.
- **Frontend:** Frontend by měl mít strukturu kódu, která bude podporovat oddělení jednotlivých částí systému, jako jsou komponenty, stránky, styly, assets a konfigurace. Tento princip zajišťuje lepší udržovatelnost, snadnější testování a možnost znovupoužití kódu.
- **Databáze:** Databáze by měla mít strukturu, která bude podporovat oddělení jednotlivých částí systému, jako jsou tabulky, pohledy, indexy a uložené procedury. Tento princip zajišťuje lepší udržovatelnost a možnost optimalizace databázových dotazů.
- **API:** Rozhraní by mělo mít strukturu, která bude podporovat oddělení jednotlivých částí systému, jako jsou endpoints, serializátory, autentizace a autorizace. Tento princip zajišťuje lepší udržovatelnost a možnost nezávislého vývoje jednotlivých částí systému.

Organizace aplikací

Frontend / backend aplikace

Systém bude rozdělen do dvou hlavních částí:

- **Backend:** Backend bude implementován jako jedna aplikace pro celý systém. Tento princip zajišťuje jednoduchou správu a nasazení, sdílení kódu a konfigurace mezi jednotlivými částmi systému a lepší bezpečnostní kontrolu. Backend bude vyvíjen v souladu s principy objektově orientované architektury, jak bylo popsáno výše.
- **Frontend:** Frontend bude implementován jako jedna aplikace pro celý systém. Frontend bude vyvíjen v souladu s principy komponentového vývoje, jak bylo popsáno výše. Tento princip zajišťuje lepší udržovatelnost a možnost znovupoužití kódu.

Backend API

Backend bude fungovat jako API pro frontend. Veškerá komunikace mezi frontendem a backendem bude probíhat prostřednictvím API. Tento princip zajišťuje oddělení zodpovědností, umožňuje nezávislý vývoj frontendu a backendu a zjednodušuje testování jednotlivých částí systému.

MVC Architektura

Systém bude implementován na principu MVC. Model-View-Controller (MVC) je architektonický vzor, který rozděluje aplikaci na tři hlavní komponenty:

- **Model:** Tato část zahrnuje logiku aplikace a zpracovává data. Model by měl být nezávislý na zbytku systému, aby bylo možné provádět změny bez ovlivnění ostatních částí systému.
- **View:** Tato část zahrnuje uživatelské rozhraní a zobrazuje data uživateli. View by mělo být co nejvíce nezávislé na logice aplikace, aby bylo možné provádět změny bez ovlivnění ostatních částí systému.
- **Controller:** Controller zpracovává požadavky uživatelů. Tato část zahrnuje jednotlivé pohledy a zajišťuje komunikaci mezi Model a View. Controller by měl být co nejvíce nezávislý na ostatních částech systému, aby bylo možné provádět změny bez ovlivnění ostatních částí systému.

Obr. 1: MVC Architektura

Řeší podstata

Řeší podstata jak data v rámci celého systému, protože se tento systém zaměřuje na vzdělávací aktivitu a tvorbu obsahu, je nutné mít dobře navrženou strukturu dat. Tato struktura by měla podporovat jak tvorbu obsahu, tak jeho publikování a zobrazení uživatelům. Dále by měla umožňovat efektivní vyhledávání a filtrování obsahu a podporovat vícejazyčnost. Struktura by měla být navržena tak, aby byla co nejvíce nezávislá na konkrétní implementaci systému.

Opis tabulka

Users

Každý uživatelský systém má alespoň tabulku uživatelů. Tabulka by měla mít: id (klíč generován Djangem), jméno, příjmení, email, heslo (hashované), datum vytvoření, datum poslední aktivity, aktivní/neaktivní, role (Administrátor, Redaktor, Uživatel), jazyk (preferovaný jazyk uživatele), avatar (URL nebo cesta k souboru), bio (krátký popis uživatele), telefon, adresa, organizace (pokud je uživatel přiřazen k organizaci), poslední IP adresa, počet neúspěšných pokusů o přihlášení, zablokovaný do (datum a čas, kdy je uživatel zablokován po neúspěšných pokusech o přihlášení), dvoufaktorové ověřování (boolean, zda je uživatel aktivoval 2FA), tajný klíč pro 2FA (šifrovaný), záložní kódy pro 2FA (šifrované).

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
email	VARCHAR	E-mail adresa (unikátní)
password_hash	VARCHAR	Hashované heslo
first_name	VARCHAR	Křestní jméno
last_name	VARCHAR	Příjmení
phone	VARCHAR	Telefonní číslo
avatar_path	VARCHAR	Cesta k souboru avatara
bio	TEXT	Krátký popis uživatele
role	VARCHAR	Role (admin, editor, user)
is_active	BOOLEAN	Aktivní/neaktivní
is_2fa_enabled	BOOLEAN	Dvoufaktorové ověřování zapnuto
created_at	TIMESTAMP	Datum vytvoření
updated_at	TIMESTAMP	Poslední aktualizace
last_login_at	TIMESTAMP	Poslední přihlášení
last_ip	VARCHAR	Poslední IP adresa
failed_attempts	INT	Počet neúspěšných pokusů o přihlášení
locked_until	TIMESTAMP	Zablokován do (NULL = není zablokován)

Roles

Tabulka rolí by měla obsahovat: id, název roly (unikátní), popis roly, systémová role (boolean, zda je systémová a nelze ji smazat), vytvořeno.

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
name	VARCHAR	Název roly (unikátní)
description	TEXT	Popis roly
is_system	BOOLEAN	Systémová role (nelze smazat)
created_at	TIMESTAMP	Datum vytvoření

Permissions

Tabulka oprávnění by měla obsahovat: id, kód oprávnění (unikátní, ve formátu resource.action), popis, modul (k kterému modulu patří).

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
code	VARCHAR	Kód oprávnění (unikátní, ve formátu resource.action)
description	TEXT	Popis oprávnění
module	VARCHAR	Modul (k kterému modulu patří)

Role_permissions

Vazební tabulka mezi rolemi a oprávněními (many-to-many). Sloupce: role_id, permission_id.

User_roles

Vazební tabulka mezi uživateli a rolemi (many-to-many). Sloupce: user_id, role_id, assigned_at (kdy byla role přiřazena), assigned_by (kdo přiřadil roli).

Sessions

Tabulka relací by měla obsahovat: id (session token), user_id, IP adresa, User-Agent, vytvořeno, poslední aktivita, platnost do.

Sloupec	Typ	Popis
id	VARCHAR	Session token (primární klíč)
user_id	BIGINT	Odkaz na uživatele
ip_address	VARCHAR	IP adresa
user_agent	TEXT	User-Agent prohlížeče
created_at	TIMESTAMP	Vytvořeno
expiry_date	TIMESTAMP	Platnost do

Audit_log

Tabulka auditního logu by měla obsahovat: id, uživatel, akce (např. login, logout, create_page, update_user, delete_media, change_permissions), typ entity (page, user, media, role, settings), ID entity, staré hodnoty (JSON), nové hodnoty (JSON), IP adresa, User-Agent, vytvořeno.

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
user_id	BIGINT	Uživatel, který provedl akci
action	VARCHAR	Akce (login, logout, create_page, update_user, delete_media, change_permissions)
entity_type	VARCHAR	Typ entity (page, user, media, role, settings)
entity_id	BIGINT	ID entity
old_values	JSON	Staré hodnoty (JSON)
new_values	JSON	Nové hodnoty (JSON)
ip_address	VARCHAR	IP adresa
user_agent	TEXT	User-Agent
created_at	TIMESTAMP	Vytvořeno

Media

Tabulka médií by měla obsahovat: id, název souboru, originální název, MIME typ, velikost, cesta k souboru, URL, šířka, výška, alternativní text, popis, uživatel, vytvořeno, upraveno.

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
filename	VARCHAR	Název souboru na disku
original_name	VARCHAR	Originální název souboru
mime_type	VARCHAR	MIME typ (image/jpeg, image/png, application/pdf, ...)
size	BIGINT	Velikost v bajtech
path	VARCHAR	Cesta k souboru
width	INT	Šířka (pro obrázky)
height	INT	Výška (pro obrázky)
alt_text	VARCHAR	Alternativní text
caption	TEXT	Popis
uploaded_by	BIGINT	Uživatel
created_at	TIMESTAMP	Vytvořeno
updated_at	TIMESTAMP	Upraveno

Media_variants

Tabulka variant médií by měla obsahovat: id, media_id, typ varianty (thumbnail, small, medium, large), cesta k souboru, URL, šířka, výška, velikost v bajtech, MIME typ.

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
media_id	BIGINT	Odkaz na médium
variant_type	VARCHAR	Typ varianty (thumbnail, small, medium, large)
path	VARCHAR	Cesta k souboru
width	INT	Šířka
height	INT	Výška
size	BIGINT	Velikost v bajtech
mime_type	VARCHAR	MIME typ
created_at	TIMESTAMP	Vytvořeno

Media_folders

Tabulka složek médií by měla obsahovat: id, název, rodičovská složka (pro stromovou strukturu), cesta, vytvořeno.

Sloupec	Typ	Popis
id	BIGAUTO	Primární klíč
name	VARCHAR	Název složky
parent_id	BIGINT	Rodičovská složka (NULL = kořenová)
path	VARCHAR	Cesta (např. /2026/03/)
created_at	TIMESTAMP	Vytvořeno

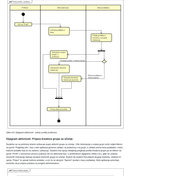
Media_folder_assignment

Vazební tabulka mezi médii a složkami (many-to-many). Sloupce: media_id, folder_id.

Diagram tabulek

Obr. 2: Diagram tabulek

Diagram napojení



Obr. 3: Diagram napojení

Diagram

Diagram tabulek ukazuje jak jednotlivé tabulky v databázovém systému spolu souvisejí. Hlavní tabulky jsou: Users (uživatelé), Roles (role), Permissions (oprávnění), Role_permissions (vazba rolí a oprávnění), User_roles (vazba uživatelů a rolí), Sessions (relace), Audit_log (auditní log), Media (soubory), Media_variants (varianty souborů), Media_folders (složky médií), Media_folder_assignment (vazba médií a složek). Vazby jsou znázorněny cizími klíči (FK) a jejich kardinalitami (1:N, N:M).

Diagram

Diagram ukazuje vazby mezi jednotlivými modely v systému. Hlavní entity jsou: User (uživatel), Role (role), Permission (oprávnění), UserRole (vazba uživatelů a rolí), RolePermission (vazba rolí a oprávnění), Session (relace), AuditLog (auditní log), Media (soubory), MediaVariant (varianty souborů), MediaFolder (složky médií), MediaFolderAss

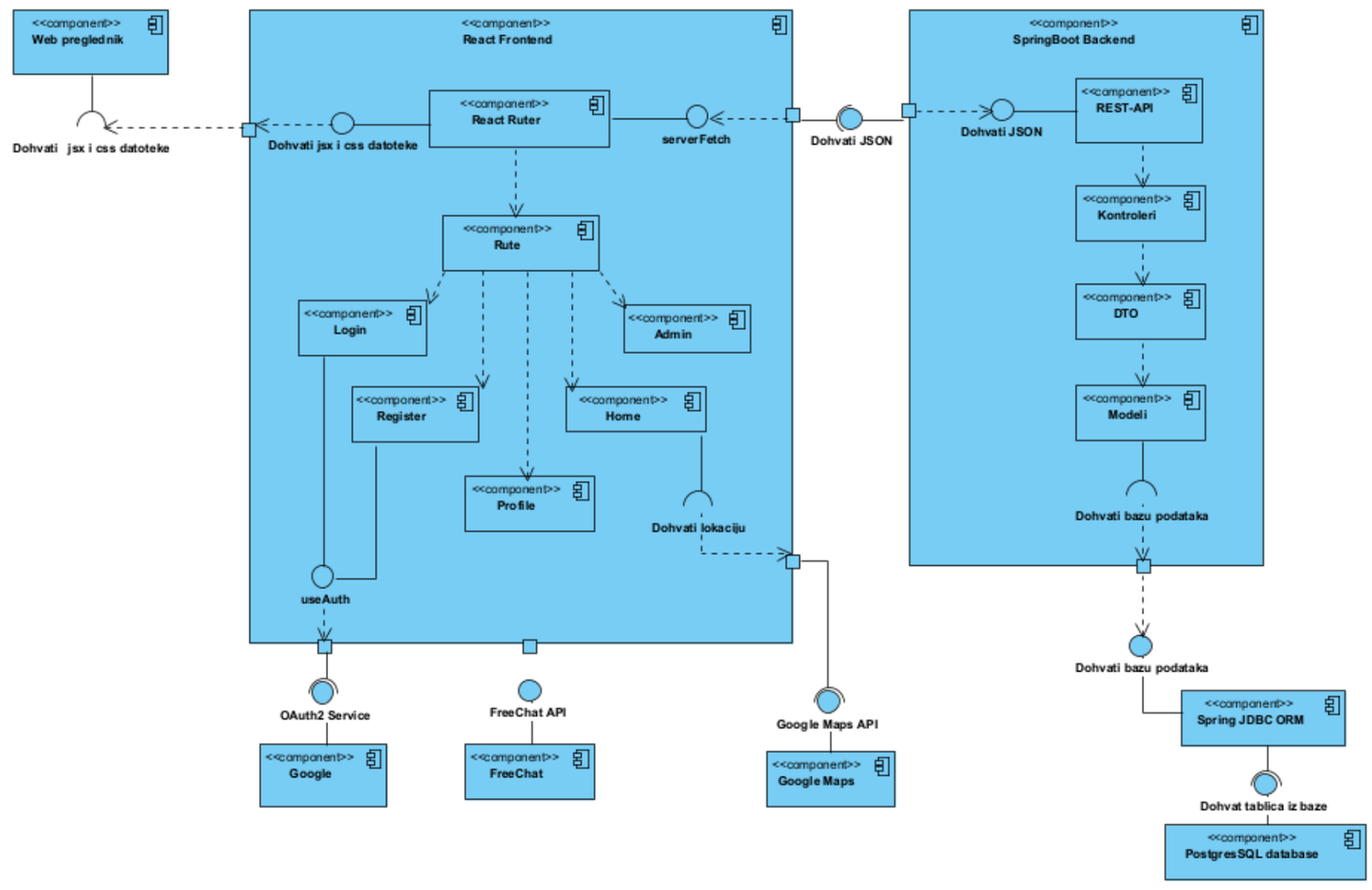
5. Arhitektura komponenata i razmještaja

Tena Osredečki edited this page yesterday · 12 revisions

Dijagram komponenti

Dijagram komponenti koriste se za modeliranje arhitekture programskog sustava na visokoj razini. Oni pružaju jasan i sažet način vizualizacije različitih komponenti ili građevnih blokova sustava i njihovih odnosa, čineći ih vrijednim alatom u programskom inženjerstvu.

Glavni elementi dijagrama su komponente, sučelja i veze između komponenti. Komponente predstavljaju modularne dijelove sustava koji mogu biti programski moduli, knjižnice, radni okviri i slično. Sučelja definiraju ugovor ili komunikacijski protokol između komponenti, specificirajući metode, operacije ili poruke koje izlažu.



Slika 5.1. Dijagram komponenti

Ovaj dijagram prikazuje arhitekturu aplikacije koja se sastoji od React frontenda i SpringBoot backenda, kao i njihove interakcije s vanjskim servisima i bazom podataka.

Web preglednik se koristi za dohvaćanje i prikaz korisničkog sučelja. Dohvaća jsx i css datoteke koje su potrebne za prikaz aplikacije. React Router upravlja glavnim rutama unutar aplikacije, a osnovne funkcionalne rute (stranice) aplikacije su: Login, Register, Profile, Home i Admin. Pri registraciji i prijavi korisnik ima mogućnost korištenja Google autentifikacije, kojom rukuje useAuth, a također može i pregledati lokaciju lekcija ili grupnog učenja, za što se brine Google Maps API.

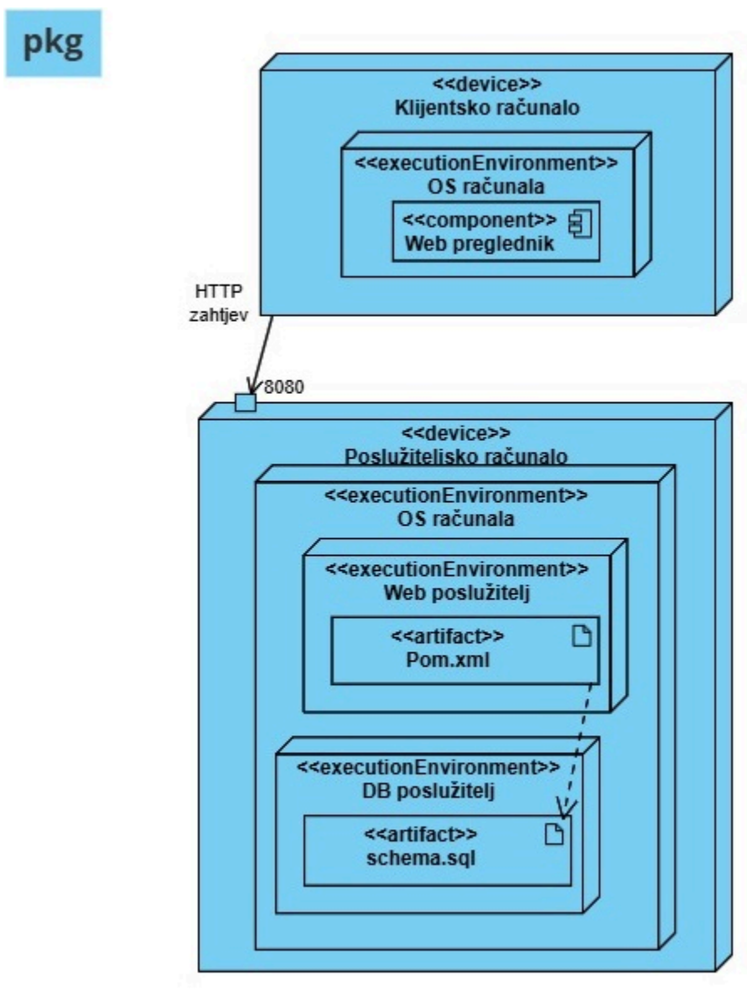
Putem serverFetch poziva dohvaća JSON podatke s REST API-a, glavne ulazne točke podataka s frontenda. Kontroleri upravljaju zahtjevima iz frontenda i prosljeđuju ih dalje kroz aplikaciju. DTO (Data Transfer Object) se koristi za prijenos podataka između kontrolera i modela, a modeli predstavljaju strukturu podataka koja se koristi u aplikaciji.

Interakciju s bazom podataka nam omogućava Spring JDBC ORM, a kao relacijsku bazu za pohranu aplikacijskih podataka koristimo PostgresSQL bazu podataka.

Dijagram razmještaja

Dijagram razmještaja koristi se za modeliranje fizičke arhitekture programskog sustava i pruža jasan i sažet način vizualizacije uređaja, okruženja za izvršavanje, artefakata i komunikacijskih veza između različitih elemenata sustava, čineći ga ključnim alatom u programskom inženjerstvu.

Glavni elementi dijagrama uključuju uređaje koji predstavljaju fizičke ili virtualne uređaje na kojima se aplikacija izvodi, okruženja za izvršavanje koja se odnose na operativne sustave, poslužitelje ili druga softverska okruženja potrebna za izvođenje aplikacije, artefakte koji predstavljaju konkretne datoteke, binarne objekte ili druge entitete potrebne za izvođenje aplikacije te veze između uređaja koje definiraju komunikacijske protokole i tok podataka između različitih dijelova sustava.



Slika 5.2. Dijagram razmještaja

Ovaj dijagram prikazuje fizičku arhitekturu aplikacije koja se sastoji od klijentskog računala i poslužiteljskog računala te njihove interakcije putem HTTP protokola. Klijentsko računalo predstavlja uređaj s operativnim sustavom, unutar kojeg je web preglednik okruženje za izvršavanje koje se koristi za dohvaćanje i prikaz korisničkog sučelja, dok su artefakti datoteke potrebne za rad korisničkog sučelja, uključujući index.html, main.js i style.css.

Klijentsko računalo šalje HTTP zahtjeve prema poslužiteljskom računalu putem porta 8080. Poslužiteljsko računalo uključuje operativni sustav kao uređaj te dva okruženja za izvršavanje: web poslužitelj koji izvodi Spring Boot aplikaciju i obrađuje zahtjeve s frontenda te poslužitelj baze podataka koji pohranjuje aplikacijske podatke pomoću PostgresSQL relacijske baze podataka. Artefakti na poslužitelju uključuju application.jar, što je konačna build datoteka Spring Boot aplikacije, i schema.sql, skriptu koja definira strukturu baze podataka. Klijentski zahtjevi šalju se prema Spring Boot aplikaciji putem REST API poziva, dok Spring JDBC ORM upravlja komunikacijom između aplikacije i PostgresSQL baze podataka.

Naziv tima: StudyBuddy

Pages 13

Find a page…

Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj…
Dijagram komponenti
Dijagram razmještaja
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a…
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

5. Systemy programowej sortowania

Dodane 2 lata temu przez admin1 Edytowane 2 lata temu przez admin1 Wyświetlenia: 1 razy

Najdłuższy wyraz

Zadanie 1 (Kolejka) - Zbiorowego Dziennika Kolejowego wyraz najdłuższy

Napisz program, który wczytuje z pliku tekstowego kolejka.txt wyrazy oddzielone spacjami i znakami interpunkcyjnymi (kropkami, przecinkami, dwukropkami, średnikami, nawiasami, cudzysłowami, myślnikami, spacjami) i wypisuje na ekranie wyraz najdłuższy oraz jego długość. Jeśli najdłuższych wyrazów jest kilka, program wypisuje pierwszy z nich. Wielkość liczy się tak samo jak w zadaniach z poprzedniej lekcji.

Wymagania: Plik kolejka.txt istnieje i zawiera co najmniej jedno słowo. Wielkość liczy się tak samo jak w zadaniach z poprzedniej lekcji. Program nie rozróżnia wielkości liczy się tak samo jak w zadaniach z poprzedniej lekcji.

Przykład: Jeśli kolejka.txt zawiera: Pociąg, Kolejka, Dziennik. Program wypisze: Kolejka 7 (Kolejka, Pociąg, Dziennik — wszystkie mają 7 liczy się tak samo jak w zadaniach z poprzedniej lekcji).

```
[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40]
```

7. Tehnologije za implementaciju aplikacije

Tena Osredečki edited this page yesterday · 13 revisions

Korištene tehnologije i alati

Tehnologije korištene za izradu aplikacije

Za frontend naše aplikacije koristili smo React.js i Javascript. Za uspostavu autorizacije i autentifikacije korisnika upotrijebili smo JWT. Za backend smo koristili SpringBoot i Javu. Backend i frontend smo kontenjerizirali s docker-om i docker-compose-om te smo za deployment iskoristili Linux VM na cloud platformi Azure. Koristimo Husky i ESlint za automatsko formatiranje datoteka u frontend/ mapi. Kako bi dopustili korisnicima pristupanje samo frontend aplikaciji i lokalizirali backend, napravili smo nginx reverse proxy. Za deployment PostgresSQL baze koristilI smo Azure Database for PostgresSQL - Flexible Server na cloud platformi Azure. Za dio aplikacije vezan za lokaciju poslužio nam je Google Maps. Za uspostavu komunikacije između korisnika koristili smo Javascript FreeChat-a čiji smo widget ugradili na frontendu. Bazu podataka idejno smo ostvarili aplikacijom ERDplus te je realizirali pomoću PostgresSQL-a, sustava za upravljanje open-source bazama. Kod smo pisali u IntelliJ, integrirano razvojno okruženje napisano u Javi za razvoj računalnog Java sustava.

Tehnologije korištene za izradu dokumentacije

Dokumentacija je pisana na Wiki dijelu GitHub-a. Za modeliranje dijagrama obrazaca uporabe, sekvencijskih dijagrama, dijagrama komponenti te klasnih dijagrama (dijagrami stanja, aktivnosti i razmještaja) koristili smo Visual Paradigm Enterprise te Astah UML, aplikacije za izradu raznih dijagrama i grafova. Aplikacije smo naizmjenično koristili po potrebi, ovisno o vizualizaciji svakog dijagrama zasebno. Također, pomogao nam je PlantUML Integration, dodatak u programu IntelliJ, koji može automatski generirati jednostavnije, manje apstraktne dijagrame.

Ostale tehnologije

Komunikaciju među članovima tima ostvarili smo korištenjem aplikacija WhatsApp i Discord. WhatsApp aplikacijom smo razmijenjivali poruke, a Discord nam je uz razmjenu poruka omogućio i komunikaciju videopozivima. Na Discordu smo temeljito oformljavali plan rada i grupe za rješavanje specifičnih problema, a za raspodjelu zadataka smo koristili Trello. Trello je alat za upravljanje projektima i organizaciju zadataka koji koristi vizualni kanban sustav. Omogućuje korisnicima da kreiraju projekte i prate njihov napredak. Pomoću njega smo imali brzi pristup popisu zadataka koje treba riješiti te smo imali uvid u to tko ih trenutno rješava ili jesu li već riješeni. Za testiranje ispravnosti koda koristili smo koristili smo ugrađeni testing framework od SpringBoot-a i Mockito, a za ispitivanje sustava koristili smo Selenium, radni okvir za izradu ispitnih slučajeva. Sustav za upravljanje izvornim kodom koji smo koristili na projektu je Git, a na platformi GitHub nalazi se udaljeni repozitorij s kodom projekta.

Internet poveznice za korištene tehnologije

1. React.js : https://reactjs.org/
2. JWT: https://jwt.io/
3. SpringBoot: https://spring.io/projects/spring-boot
4. Docker: https://www.docker.com/
5. Azure Cloud Linux VM: https://azure.microsoft.com/en-us/products/virtual-machines/linux
6. NGNIX: https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/
7. Azure Database for PostgresSQL: https://azure.microsoft.com/en-us/products/postgresql
8. Husky: https://medium.com/@mariokandut/how-to-add-husky-to-react-233f0ca48752
9. ESlint: https://eslint.org/
10. PostgreSQL: https://www.postgresql.org/
11. ERDplus: https://erdplus.com/
12. Google Maps: https://www.google.com/maps
13. Freechat PopUp: https://freechatpopup.popupsmart.com/
14. Visual Paradigm Enterprise: https://online.visual-paradigm.com/
15. Astah UML: https://astah.net/products/astah-uml/
16. IntelliJ: https://www.jetbrains.com/idea/
17. WhatsApp: https://www.whatsapp.com/
18. Discord: https://discord.com/
19. Trello: https://trello.com
20. Selenium: https://www.selenium.dev/selenium-ide/
21. Mockito: https://site.mockito.org/
22. Git: https://git-scm.com/
23. GitHub: https://github.com/

Naziv tima: StudyBuddy

Pages 13
Find a page…
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj…
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a…
Korištene tehnologije i alati
Tehnologije korištene za izradu aplikacije
Tehnologije korištene za izradu dokumentacije
Ostale tehnologije
Internet poveznice za korištene tehnologije
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

8. Upute za puštanje u pogon

Erik Kranjec edited this page 3 hours ago · 18 revisions

Opis pristupa aplikaciji na javnom poslužitelju

Pristupi http://study-buddy.westeurope.cloudapp.azure.com/

Detaljni raspis koraka koje smo pratili da bismo pustili aplikaciju u pogon

Izbor dizajna i obrazloženja

1. Ručno puštanje u pogon i ažuriranje aplikacije

Budući da za projekt imamo točno definirane checkpointe, nismo vidjeli potrebu stvarati cijeli pipeline automatskog ažuriranja aplikacije u produkciji. Umjesto toga, ručno se spojimo na server, napravimo `git pull` i slijedimo nekoliko jednostavnih koraka za deployanje aplikacije. Ovo uključuje ručno postavljanje baze i manjak migracija.

2. Deployana aplikacija u development environmentu

Većinski isti razlog kao natuknica prije.

3. Azure kao odabrana Cloud platforma

Kao studenti, dobijemo $100 kredita na Azure cloud platformi. Nudi nam mnoštvo opcija i kvalitetno je dokumentirana. Odlučili smo koristiti nešto što je standard u današnjoj industriji, a pronaći najjeftinije rješenje koje i dalje dobro funkcionira.

Baza podataka

Koristimo PostgresSQL bazu verzije 16.

Način deployanja

Koristili Azure-ov dostupni SQL Server, Azure Database for PostgreSQL - Flexible Server. Odabrali smo taj Cloud servis zbog jednostavnosti korištenja specijalizirane virtualne mašine.

Upute za deployment baze i korištenje iste u SpringBootu

Napravili smo novi Azure Database for PostgreSQL - Flexible Server resurs u Azure-u, postavili postavke servera na najslabije performanse i stavili ga u zonu Germany West Central. Postavili smo username i password. Specifična konfiguracija je Burstable, B1ms, 1 vCores, 2 GiB RAM, 32 GiB storage.

Zatim, komandom `psql --host=ime-database-resursa.database.azure.com --port=5432 --username=username --dbname=ime_baze` te upisivanjem passworda se spojimo direktno na bazu koju želimo koristiti. Istom komandom, ali ako još dodamo flag `-f /path/do/schema.sql` možemo direktno izvršiti neki SQL upit. Na taj način ručno postavimo tablice i "mock" podatke koje smo prije definirali.

Uputu za puštanje aplikacije u pogon

Aplikacija je deployana na Ubuntu virtualnoj mašini. Konfiguracija je Standard DS1 v2 (1 vcpu, 3.5 GiB memory). Uključena je Availibility zone 2 postavka. Mašina ima otvoren HTTP port za cijeli IP spektar.

Potrebno je u `frontend/` staviti .env.production datoteku:

```
VITE_API_BASE_URL=/api
VITE_GOOGLE_MAPS_API = api key
VITE_GOOGLE_MAPS_MAPID = id
VITE_GOOGLE_CLIENT_ID = id
```

U `docker-compose.yml` datoteci moramo zamijeniti linije označene s `# Dummy data` sa serverom na kojem je baza te ispravnim korisničim imenom i lozinkom.

Na mašini se instalira Docker verzije 27.2.0, s `sudo snap install docker`. Potrebni su maven:3.8.4-openjdk-17 i react-app imagei. `Dockerfile` i `docker-compose.yml` su dostupni u root folderu projekta. Nginx konfiruacija se nalazi na `frontend/.docker/nginx.conf`. U `docker-compose.yml` moramo zamijeniti dummy potadke s istim argumentima kao u `application.properties`

Potrebno je pokrenuti komandu `sudo chmod 666 /var/run/docker.sock` jer ne pokrećemo kao root user, pa Docker ne može pristupiti Docker daemon socketu. Zatim s `docker compose up --build -d` pokrenemo docker kontejner.

Naziv tima: StudyBuddy

Pages 13
Find a page…
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj…
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a…
8. Upute za puštanje u pogon
Opis pristupa aplikaciji na javnom poslužitelju
Detaljni raspis koraka koje smo pratili da bismo pustili aplikaciju u pogon
Izbor dizajna i obrazloženja
Baza podataka
Način deployanja
Upute za deployment baze i korištenje iste u SpringBootu
Uputu za puštanje aplikacije u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

9. Zaključak i budući rad

Tena Osredečki edited this page yesterday · 4 revisions

Naš zadatak bio je razviti aplikaciju koja omogućava studentima formiranje grupa za učenje s drugim studentima te profesorima formiranje lekcija za studente. Tijekom 15 tjedana timskog rada i razvoja zadatka, kroz 2 ključne faze, ostvarili smo naš cilj.

U prvoj smo fazi okupili tim od 7 ljudi i podijelili se na grupu koja radi backend dio aplikacije, grupu koja radi frontend dio aplikacije i voditelja, koji je obnašao dužnost full stack razvojnog inženjera. Za početak je dokumentacija bila ključna, tako da smo se posvetili izradi vizualnih rješenja aplikacije, pisanju obrazaca uporabe i funkcionalnih zahtjeva, izradi početnih dijagrama, detaljima ponašanja aplikacije, itd.

U drugoj fazi fokusirali smo se na razvoj same aplikacije u SpringBoot i React razvojnim okruženjima te krenuli izrađivati popis zadataka na aplikaciji Trello, koja nam je omogućila direktan uvid u količinu i težinu zadataka te nam olakšala raspodjelu posla. Dokumentirali smo ostatak potrebnih dijagrama (dijagram stanja, komponenti, razmještaja i aktivnosti) kako bi funkcionalnosti bile jasnije budućim korisnicima. Smišljali smo pametan izgled i lijepu prezentaciju aplikacije, kako bi bila izvrsno prilagođena korisnicima i intuitivna za korištenje. Komunikacija je bila ključna jer smo u ovoj fazi mnogo ovisili o ostalim članovima tima i njihovom radu.

Najviše vremena nam je oduzelo upoznavanje s alatima potrebnima za izradu naše aplikacije, ali ti problemi su prevladani kvalitetnom komunikacijom putem WhatsAppa i Discorda, čestim sastancima i marljivim radom. Moguća nadogradnja bila bi izrada mobilne aplikacije, koja bi omogućila korisnicima jednostavnost pri istraživanju grupa dok nemaju laptop i lakšu komunikaciju s drugim studentima i profesorima.

Ovaj projekt je bio vrlo vrijedno iskustvo koje ćemo moći dalje primijeniti. Doživjeli smo rad u grupi gdje je važna dobra komunikacija, organizacija i suradnja te smo stekli bitna znanja za nastavak fakulteta i poslovnog života.

Naravno, prostora za usavršavanje uvijek ima, ali postigli smo značajni napredak u razumijevanju izrade i implementacije web aplikacija te iskustvu funkcioniranja u timu za razvoj aplikacija. Zadovoljni smo našim stečenim znanjima i našom aplikacijom.

Naziv tima: StudyBuddy

Pages 13

Find a page…

- Home
- 1. Opis projektnog zadatka
- 2. Analiza zahtjeva
- 3. Specifikacija zahtjeva sustava
- 4. Arhitektura i dizajn sustava
- 5. Arhitektura komponenata i razmj…
- 6. Ispitivanje programskog rješenja
- 7. Tehnologije za implementaciju a…
- 8. Upute za puštanje u pogon
- 9. Zaključak i budući rad
- A. Popis literature
- B. Dnevnik promjena dokumentacije
- C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

A. Popis literature

Mila Podrug edited this page 2 days ago · 4 revisions

1. Programsko inženjerstvo, FER ZEMRIS, https://www.fer.unizg.hr/predmet/proinz
2. Astah Community, https://astah.net/products/astah-uml/
3. The Unified Modeling Language, https://www.uml-diagrams.org/
4. Stack Overflow, https://stackoverflow.com/
5. Visual Paradigm, https://www.visual-paradigm.com/
6. PostgresSQL Docs, https://www.postgresql.org/
7. ChatGPT, https://chatgpt.com/
8. React Docs, https://legacy.reactjs.org/docs/getting-started.html
9. SpringBoot Docs, https://spring.io/projects/spring-boot
10. IntelliJ IDEA Docs, https://www.jetbrains.com/idea/

Naziv tima: StudyBuddy

Pages 13
Find a page…
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj…
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a…
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

B. Dnevnik promjena dokumentacije

Tena Osredečki edited this page 2 hours ago · 4 revisions

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Anamarija Kic	23.10.2024
0.2	Raspisana većina obrazaca uporabe	Tena Osredečki, Ella Grković	28. i 29.10.2024
0.3	Razrađena analiza zahtjeva	Anamarija Kic	30.10.2024
0.4	Dodano nekoliko dijagrama obrazaca uporabe	Tena Osredečki	30.10.2024
0.5	Raspisana tablica uključenosti ključnih funkcionalnosti u obrasce uporabe	Tena Osredečki	31.10.2024
0.6	Napisan cjelovit opis projektnog zadatka	Anamarija Kic	5.11.2024
0.7	Dodan opis arhitekture sustava	Ella Grković	8.11.2024
0.8	Dijagram baze podataka i opisi tablica baze podataka	Ella Grković	9.11.2024
0.9	Sekvencijski dijagrami	Ella Grković	10. i 11.11.2024
0.10	Završni ispravci opisa obrazaca uporabe	Tena Osredečki	12.11.2024
0.11	Dodan opis sekvencijskih dijagrama	Ella Grković	14.11.2024
0.12	Završni ispravci opisa arhitekture sustava	Ella Grković	15.11.2024
0.13	Dijagram razreda	Tena Osredečki	15.11.2024
0.14	Opis dijagrama razreda	Anamarija Kic	15.11.2024
0.15	Ispitivanje programskog rješenja	Anamarija Kic	15.1.2025
0.16	Dijagram razmještaja i opis	Darian Begović	16.1.2025
0.17	Općeniti opis dinamičkog ponašanja aplikacije	Tena Osredečki	17.1.2025
0.19	Uređivanje popisa literature	Mila Podrug	20.1.2025
0.20	Raspis zaključka projekta	Mila Podrug	20.1.2025
0.21	Dijagrami aktivnosti i opisi dijagrama	Tena Osredečki	21.1.2025
0.22	Dijagram stanja i opis	Tena Osredečki	22.1.2025
0.23	Upute za puštanje u pogon	Erik Kranjec	22.1.2025
0.24	Opisi tehnologija za implementaciju aplikacije	Mila Podrug	22.1.2025
0.25	Kompletan opis arhitekture komponenata i razmještaja, dijagram komponenata	Mila Podrug	22.1.2025
0.26	Završni ispravci opisa baze podataka	Ella Grković	24.1.2025
0.27	Završni ispravci ostatka dokumentacije	Mila Podrug, Erik Kranjec, Tena Osredečki	24.1.2025

Naziv tima: StudyBuddy

Pages 13
Find a page…
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj…
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a…
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe

Clone this wiki locally

https://github.com/savipauk/study

C. Prikaz aktivnosti grupe

Anamarija Kic edited this page 1 hour ago · 20 revisions

Dnevnik sastajanja

1. sastanak

- Datum: 15. listopada 2024.
- Prisustvovali: T. Osredečki, M. Podrug, D. Begović, E. Grković, E. Kranjec, K. Mezdić, A. Kic
- Teme sastanka:

> - upoznavanje članova tima
> - proučavanje zadatka
> - predlaganje vlastite projektne ideje
> - raspodjela prvih zadataka
> - postavljanje GitHuba
> - uspostavljena Discord grupa svih članova
> - uspostavljena WhatsApp grupa svih članova

2. sastanak

- Datum: 22. listopada 2024.
- Prisustvovali: T. Osredečki, M. Podrug, D. Begović, E. Grković, E. Kranjec, K. Mezdić, A. Kic
- Teme sastanka:

> - prihvaćena naša vlastita projektna ideja - daljnja razrada ideje
> - određivanje funkcijskih i nefunckijskih zahtjeva
> - početna vizualizacija izgleda aplikacije
> - raspodjela novih zadataka

3. sastanak

- Datum: 28. listopada 2024.
- Prisustvovali: T. Osredečki, M. Podrug, D. Begović, E. Grković, E. Kranjec, K. Mezdić, A. Kic
- Teme sastanka:

> - razrada daljnjih zadataka dokumentacije - tablica (ne)funkcionalnih zahtjeva, crtanje UML dijagrama
> - početak pisanja koda frontenda i backenda
> - raspodjela novih zadataka

4. sastanak

- Datum: 6. studenog 2024.
- Prisustvovali: T. Osredečki, M. Podrug, D. Begović, E. Grković, E. Kranjec, K. Mezdić, A. Kic
- Teme sastanka:

> - razrada daljnjih zadataka dokumentacije - crtanje sekvencijskog dijagrama, završna razrada opisa projektnog zadatka, rad opisa baze podataka
> - povezivanje baze podataka s programom
> - implementacija Oauth-a u projekt
> - raspodjela novih zadataka

5. sastanak

- Datum: 13. prosinca 2024.
- Prisustvovali: T. Osredečki, M. Podrug, D. Begović, E. Grković, E. Kranjec, K. Mezdić, A. Kic
- Teme sastanka:

> - retrospekcija na aktivnosti prvog ciklusa
> - raspodjela novih zadataka
> - početak kreiranja StuddyGrupa u aplikaciji

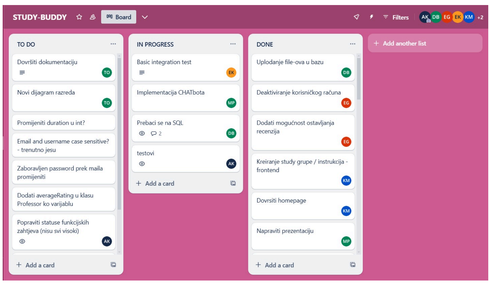
6. sastanak

- Datum: 9. siječnja 2024.
- Prisustvovali: T. Osredečki, M. Podrug, D. Begović, E. Grković, E. Kranjec, K. Mezdić, A. Kic
- Teme sastanka:

> - komentiranje i određivanje finalnih ciljeva i mogućnosti
> - podjela završnih zadataka
> - međusobno objašnjavanje koda

Plan rada

Za lakše upravljanje zadacima, koristili smo aplikaciju Trello.

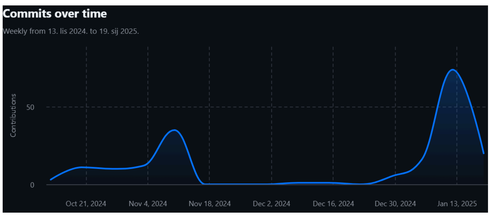


Tablica aktivnosti

napomena: u upravljanje projektom naveli smo i vrijeme provedeno na sastancima

aktivnost	T.O.	M.P.	D.B.	E.G.	A.K.	K.M.	E.K
Upravljanje projektom	7	7	7	7	7	7	7
Opis projektnog zadatka	1	1			5		
Funkcionalni zahtjevi	2	2	2	2	4	2	2
Opis pojedinih obrazaca	10	10		2			
Dijagrami obrazaca	5						
Sekvencijski dijagrami				5			
Opis ostalih zahtjeva	1	1	1	1	2	1	1
Arhitektura i dizajn sustava		2		12			1
Baza podataka		15	5	2.5			1
Dijagram razreda	1.5				0.5		
Dijagram stanja	4						
Dijagram aktivnosti	4						
Dijagram komponenti		5					
Korištene tehnologije i alati		2					
Ispitivanje programskog rješenja					20	0.5	1
Dijagram razmještaja			3				
Upute za puštanje u pogon							1
Dnevnik sastajanja					0.5		
Zaključak i budući rad		2					
Popis literature		1					
Backend		22	5		70		15
Frontend	2			50		95	12
Puštanje u pogon							10
Učenje programskih alata	3	3	5	7	8	6	2

Dijagram pregleda promjena



Ključni izazovi i rješenja

Opis izazova

Tijekom projekta suočili smo se s kašnjenjem u razvoju, nedovoljno komunikacije i malim iskustvom. Problemi su se postepeno smanjivali s vremenom, pogotovo nakon uspostave dobre dinamike tima. Najveći izazov na početku bio je učenje alata, stoga je 2. ciklus bio osjetno lakši. Trebali smo imati više sastanaka uživo, što bi nam olakšalo pri raspoređivanju zadataka, sklapanju daljnjih dogovora i idejnom razvoju aplikacije.

Rješenja

Izazove koji su nas snašli rješavali smo korak po korak, uz konstantno učenje alata i razradu dokumentacije. Stekli smo iskustvo rada u timu, što je bilo ključno u završnim fazama izrade aplikacije. Bilo je ključno upoznati članove tima ponaosob kako bi se što bolje poticali na rad i razvili kvalitetnu komunikaciju. Puno su nam pomogle aplikacije poput Discord-a i Trello-a, pomoću kojih smo održavali česte sastanke te rasprave o kodu i daljnjim fazama rada na aplikaciji.

Naziv tima: StudyBuddy

Pages 13
Find a page...
Home
1. Opis projektnog zadatka
2. Analiza zahtjeva
3. Specifikacija zahtjeva sustava
4. Arhitektura i dizajn sustava
5. Arhitektura komponenata i razmj...
6. Ispitivanje programskog rješenja
7. Tehnologije za implementaciju a...
8. Upute za puštanje u pogon
9. Zaključak i budući rad
A. Popis literature
B. Dnevnik promjena dokumentacije
C. Prikaz aktivnosti grupe
Dnevnik sastajanja
Plan rada
Tablica aktivnosti
Dijagram pregleda promjena
Ključni izazovi i rješenja
Opis izazova
Rješenja

Clone this wiki locally

https://github.com/savipauk/study