



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Anja Lečnik

SKILLTRACK

Diplomski projekt

Maribor, maj 2025

KAZALO VSEBINE

1 OPIS PROJEKTA	1
2 DIAGRAMI.....	2
2.1 Use case diagram	2
2.2 Deployment diagram	2
2.3 E-R diagram.....	3
3 TEHNIČNE SPECIFIKACIJE	4
3.1 Frontend	4
3.2 Backend	4
3.3 Integracije.....	5
4 TESTIRANJE	6
4.1 Testi	6
4.2 SonarCloud	6
4.3 GitGuardian	6
5 DOKUMENTACIJA ODLOČITEV	8
5.1 Arhitektura sistema	8
5.2 Avtentikacija in varnost	8
5.3 Organizacija dela.....	8
5.4 Zagotavljanje kakovosti	9

5.5 Večjezičnost in prilagajanje vmesnika	9
--	----------

1 OPIS PROJEKTA

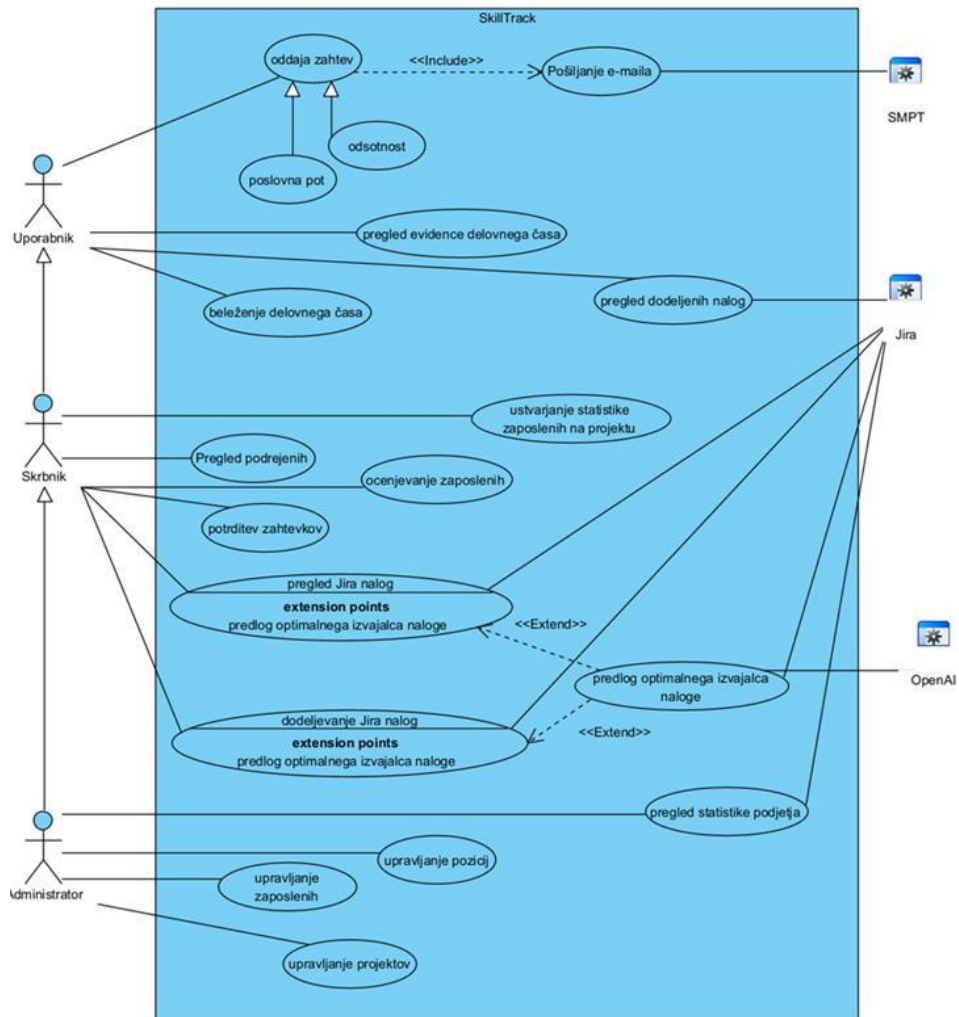
SkillTrack je napreden spletni sistem za celovito upravljanje zaposlenih, kompetenc, projektov in evidenc delovnega časa. Namenjen je podjetjem, ki želijo optimizirati učinkovitost ekip, pametno dodeljevati naloge ter imeti centraliziran pregled nad kadrovskimi in projektnimi procesi.

Glavne značilnosti sistema:

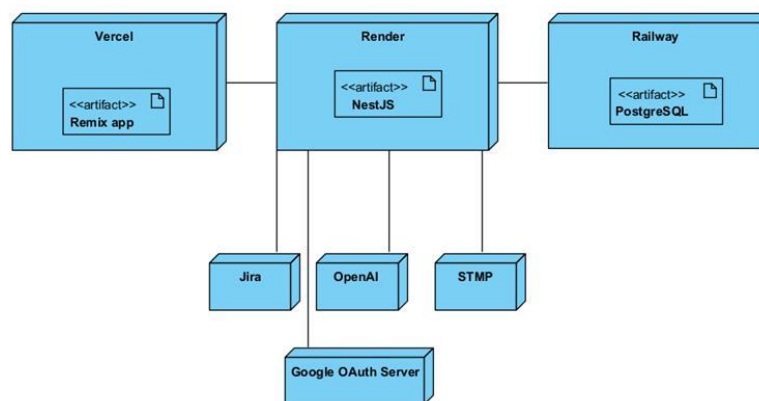
- Pametno upravljanje nalog: sistem uporablja OpenAI API za analizo Jira ticketov in predlaga najbolj primerne zaposlene glede na njihove kompetence.
- Evidenca delovnega časa in odsotnosti: uporabniki lahko beležijo prisotnost, dopuste in bolniške.
- Povezava z Jiro: avtomatski uvoz nalog preko Jira REST API, s povezovanjem ticketov z zaposlenimi in statusnim sledenjem.
- Vloga in pravice: podpora vlogam User, Supervisor in Admin, z različno stopnjo dostopa.
- Večjezičnost in lokalna prilagoditev: uporabniški vmesnik v slovenščini, prilagojen za uporabo v slovenskih podjetjih.
- Email obveščanje: sistem samodejno pošilja obvestila o pomembnih dogodkih (npr. potrjeni dopusti, službene poti).

2 DIAGRAMI

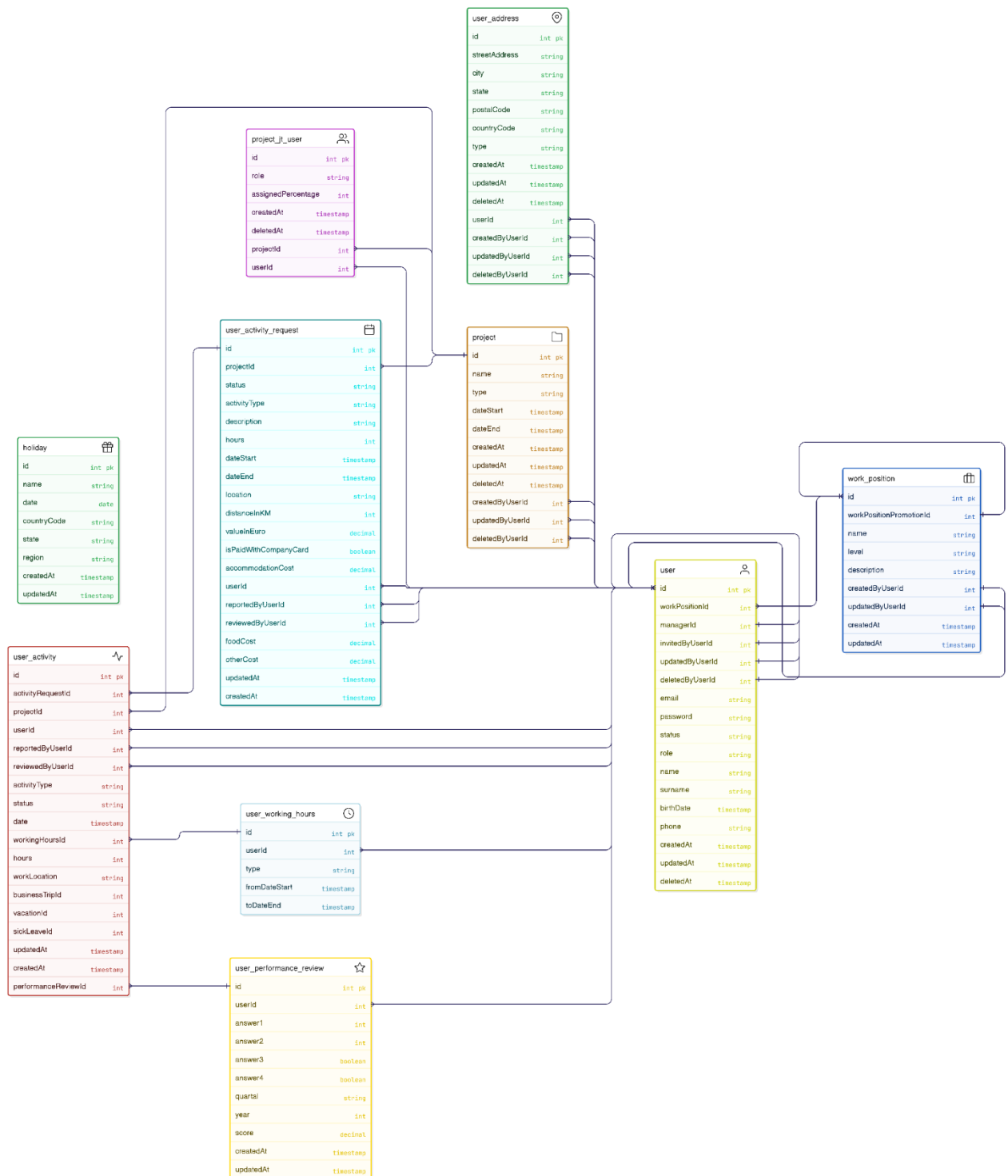
2.1 Use case diagram



2.2 Deployment diagram



2.3 E-R diagram



3 TEHNIČNE SPECIFIKACIJE

3.1 Frontend

Uporabniški vmesnik aplikacije SkillTrack je razvit z uporabo ogrodja Remix in programskega jezika TypeScript ter je gostovan na platformi Vercel, kar omogoča hitro nalaganje in enostavno vzdrževanje. Uporabniška izkušnja je prilagojena glede na vlogo prijavljenega uporabnika (User, Supervisor, Admin), vmesnik pa je preveden v slovenščino za boljšo lokalno dostopnost.

Frontend aplikacija za delovanje uporablja več okoljskih spremenljivk, definiranih v datoteki .env. Te vključujejo nastavitve za povezavo z API-jem (API_URL), Firebase konfiguracijo za obvestila in Google OAuth avtentikacijo (vključno z ID-jem in varnostnim ključem aplikacije).

3.2 Backend

Backend del aplikacije je implementiran z uporabo ogrodja NestJS, ki temelji na arhitekturi REST. Aplikacija je gostovana na platformi Render, medtem ko za podatkovno hrambo uporablja relacijsko bazo PostgreSQL, nameščeno preko storitve Railway. Za vsako novo objavo aplikacije je potrebna ročna gradnja backend kode z nest build, saj brezplačna različica Rendera ne podpira avtomatske gradnje zaradi omejitev pomnilnika.

API-ji so dokumentirani z uporabo Swagger in omogočajo enostavno testiranje. Backend podpira Google OAuth za avtentikacijo, za seje pa uporablja JWT žetone, ki se definirajo preko konfiguracije v .env datoteki. Določene funkcionalnosti (npr. omejitev števila dni za urejanje vnosa) so prav tako nastavljene preko konfiguracijskih spremenljivk.

3.3 Integracije

Aplikacija vključuje več ključnih integracij z zunanjimi storitvami, ki bistveno povečujejo njeno funkcionalnost in avtomatizacijo delovnih procesov.

Za prijavo uporabnikov se uporablja integracija z Google OAuth, ki omogoča varno avtentikacijo, brez potrebe po internem upravljanju gesel. Uporabniki se lahko prijavijo prek svojega Google računa, s čimer se poenostavi uporabniška izkušnja in izboljša varnost sistema.

Za sinhronizacijo projektnih nalog in povezovanje z zaposlenimi je implementirana integracija z Jira REST API. Sistem omogoča uvoz Jira ticketov, njihovo povezovanje s konkretnimi zaposlenimi ter prikaz stanja in napredka projektov znotraj aplikacije SkillTrack.

Ena ključnih inovacij sistema je tudi povezava z OpenAI API, ki omogoča analizo vsebine Jira nalog. Na podlagi analize vsebine in kompetenc zaposlenih sistem predlaga najprimernejšega izvajalca za določeno nalogo. S tem SkillTrack prinaša inteligentno dodeljevanje nalog, ki presega klasične ročne pristope.

Za učinkovito interno komunikacijo in obveščanje uporabnikov o pomembnih dogodkih (npr. oddaja zahtevka za dopust) je vzpostavljena integracija z elektronsko pošto prek SMTP strežnika. Aplikacija samodejno pošilja obvestila uporabnikom, s čimer se zmanjšuje potreba po ročnem komuniciranju in se izboljša odzivnost v delovnih procesih.

4 TESTIRANJE

4.1 Testi

Za preverjanje delovanja uporabniškega vmesnika in ključnih funkcionalnosti so bili vzpostavljeni E2E (end-to-end) testi z orodjem Playwright. Ti testi simulirajo dejansko uporabniško interakcijo z aplikacijo (npr. oddaja zahtevka za dopust, upravljanje z uporabniki) in tako omogočajo zgodnje odkrivanje napak ter zagotavljanje stabilnosti sistema ob nadgradnjah.

Playwright omogoča izvajanje testov tako v CLI okolju kot tudi preko interaktivnega uporabniškega vmesnika (--ui), kar olajša odpravljanje napak med razvojem. Testi se nahajajo v mapi playwright-tests, kjer je organizirana struktura po funkcionalnostih oziroma straneh.

4.2 SonarCloud

Za zagotavljanje kakovosti in vzdrževanosti kode se uporablja SonarCloud, ki omogoča statično analizo izvirne kode. Orodje preverja prisotnost t.i. "code smells", podaja varnostna opozorila, meri pokritost s testi in ocenjuje tehnični dolg.

SonarCloud je integriran v GitHub repozitorij in se samodejno sproži ob vsakem push ali pull request dogodku. Rezultati analize so dostopni preko SonarCloud spletnega vmesnika ter služijo kot orodje za sprotno izboljševanje kode in preventivno odpravo napak.

4.3 GitGuardian

Za dodatno zaščito pred morebitnimi varnostnimi tveganji je integriran GitGuardian, ki skenira repozitorij za naključno izpostavljene skrivnosti (npr. API ključe, gesla, JWT tokene). GitGuardian omogoča hitro zaznavanje takšnih napak že med razvojem in pred objavo kode, kar bistveno zmanjšuje možnost zlorab.

V primeru zaznave občutljivih podatkov razvijalec prejme opozorilo ter navodila za varno odstranitev in zamenjavo izpostavljenih vrednosti.

5 DOKUMENTACIJA ODLOČITEV

5.1 Arhitektura sistema

SkillTrack je zasnovan kot spletna aplikacija s tremi ločenimi komponentami:

- Frontend: Remix + TypeScript, gostovano na Vercel.
- Backend: NestJS (REST API), gostovan na Render.
- Baza podatkov: PostgreSQL na Railway.

Odločitev za ločen frontend in backend omogoča boljšo razširljivost, neodvisen razvoj, lažje testiranje ter fleksibilnost pri bodočih nadgradnjah. Remix in NestJS sta bila izbrana zaradi dobre podpore za TypeScript, SSR, modularne arhitekture in obstoječih izkušenj iz industrijskega okolja. PostgreSQL ustreza zaradi zanesljivosti in razširjenosti v povezavi z relacijskim modelom podatkov.

5.2 Avtentikacija in varnost

Za avtentikacijo uporabnikov je bila izbrana integracija z Google OAuth, ki omogoča varno prijavo brez upravljanja gesel. Sistem uporablja JWT za seje in osvežitvene tokene. Zavestno je bila izpuščena podpora drugim načinom prijave, kar poenostavi razvoj in zmanjša možnosti napak.

5.3 Organizacija dela

Projekt je bil razvit iterativno, s tedenskimi usklajevalnimi sestanki z mentorico (8., 16., 23., 30. maj in 6. junij 2025), na katerih so bili pregledani napredki, določene prioritete in potrjene tehnične odločitve. Med pomembnejšimi odločitvami pri organizaciji dela so bile:

- Odločitev, da sistem podpira le eno delovno okolje (*workspace*), je bila sprejeta zaradi poenostavitve izvedbe in osredotočenosti na osnovno funkcionalnost, s čimer se je zagotovila večja osredotočenost na temeljne funkcionalnosti.
- Združitev funkcionalnosti za službene poti in stroške v eno komponento je bila izvedena na podlagi predloga mentorice z namenom izboljšanja preglednosti in enostavnejšega upravljanja.

- Razvoj posameznih funkcionalnosti je potekal sproti, skladno s sprotnimi prioritetami in povratnimi informacijami mentorice, kar je omogočilo boljšo prilagoditev sistema dejanskim potrebam uporabnikov.

5.4 Zagotavljanje kakovosti

Sistem vključuje več pristopov za zagotavljanje kakovosti:

- Playwright testi za preverjanje uporabniških tokov.
- SonarCloud za statično analizo kode (code smells, varnost, pokritost).
- Swagger dokumentacija za opis REST API-jev.
- GitGuardian za zaznavanje izpostavljenih skrivnosti v kodi.

5.5 Večjezičnost in prilagajanje vmesnika

Aplikacija je prevedena v slovenščino za boljšo lokalno uporabniško izkušnjo. Vmesnik je prilagojen vlogam uporabnikov (User, Supervisor, Admin), kar izboljša uporabnost glede na njihove pristojnosti.