TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Eliise Karafin 233541IAIB
Eva Kazakovskaia 233034IAIB

Veebipõhise reisihaldusrakenduse kavandamine ja arendamine

Bakalaureusetöö

Juhendaja: Ago Luberg
PhD

Tallinn 2026

Autorideklaratsioon

Kinnitame, et oleme koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autorid: Eliise Karafin ja Eva Kazakovskaia

01.06.2026

Lühikokkuvõte

Käesoleva bakalaureusetöö eesmärgiks on luua veebipõhine reisihaldusrakendus, mis võimaldab kasutajatel planeerida ja hallata reise ühtses keskkonnas. Töö keskendus eelkõige organisatsiooniliste projektireiside haldamise kasutusjuhtudele, analüüsides ühe organisatsiooni tööprotsesse ja tellija vajadusi.

Rakenduse peamisteks funktsionaalsusteks on reiside ja päevapõhiste ajakavade haldamine, osalejate ja rollide haldamine, kulude jälgimine, dokumentide lisamine ning aruannete genereerimine. Lisaks sisaldab süsteem tehisintellektil põhinevat komponenti, mis võimaldab genereerida reisiplaane ja soovitusi.

Meeskonnatööna valminud süsteem koosneb eraldiseisvast klientrakendusest, tagarakendusest ning LLM-põhisest tehisintellekti komponendist. Klientrakendus on realiseeritud Reacti ja TypeScripti abil ning tagarakendus Java Spring Boot raamistikus, kasutades PostgreSQL andmebaasi.

Süsteemi valideerimiseks kasutati nii automaat- kui ka kasutajatestimist, mille käigus hinnati süsteemi funktsionaalsust, kasutusmugavust ning vastavust nõuetele.

Tulemuseks valmis terviklik rakendus, mis vastab töö käigus kaardistatud põhinõuetele ning aitab vähendada projektireiside haldamisega seotud killustatud töökorraldust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 42 leheküljel, 7 peatükki, 3 joonist ja 2 tabelit.

Abstract

Design and Development of a Web-Based Travel Management Application

The aim of this bachelor's thesis is to develop a web-based travel management application that enables users to plan and manage trips within a unified environment. The thesis focused primarily on organizational project travel use cases by analyzing the workflows and requirements of a specific organization.

The main functionalities of the application include travel and itinerary management, participant and role management, expense tracking, document uploading, and report generation. In addition, the system includes an artificial intelligence based component that enables the generation of travel plans and recommendations.

The system, developed as a team project, consists of a separate client application, backend application, and an LLM-based artificial intelligence component. The client application was implemented using React and TypeScript, while the backend application was developed using the Java Spring Boot framework with a PostgreSQL database. The system architecture follows a modular multi-layered approach, where the frontend, backend, database, and AI components are separated into independent layers to improve maintainability, scalability, and extensibility. The artificial intelligence component was implemented as a separate service communicating with the main application through HTTP-based requests.

The thesis followed a Design Science Research (DSR) approach, where the development process focused on solving a practical real-world problem through the creation and evaluation of a functioning software artifact. The work included requirement analysis, comparison of existing travel management solutions, system design and implementation, as well as validation with both the client and end users.

Both automated and user-based testing methods were used to validate the system, evaluating its functionality, usability, and compliance with the defined requirements. Validation by

the client confirmed that the final product successfully met organizational expectations by improving information accessibility, centralizing travel-related data, and supporting role-based access control.

Usability testing conducted with 10 participants produced highly positive feedback regarding the system's ease of use, navigation logic, and task completion. Users especially highlighted the automated expense settlement functionality and AI-driven "Autofill Day" feature as practical strengths that reduced manual work and accelerated trip planning. Participants also valued the ability to manage schedules, expenses, files, and participant information within a single environment, which reduced the need to switch between multiple tools and communication channels. The testing results demonstrated that the system is suitable not only for organizational project travel management but also for individual and group-based travel planning scenarios.

As a result, a complete application was developed that satisfies the main requirements identified during the thesis and helps reduce the fragmented workflow associated with project travel management. The developed solution demonstrates how integrating travel planning, collaboration, expense management, document handling, reporting, and AI-assisted recommendations into a single web-based environment can improve workflow efficiency and reduce administrative overhead in project travel management.

The thesis is written in Estonian and contains 42 pages of text, 7 chapters, 3 figures and 2 tables.

Lühendite ja mõistete sõnastik

ACID	Transaktsioonide omadused: atomaarsus, järjepidevus, isoleeritus ja püsivus (*Atomicity, Consistency, Isolation, Durability*)
AI	Tehisintellekt (*Artificial Intelligence*)
API	Rakendusliides (*Application Programming Interface*)
DSR	Disainiteaduslik uurimislähenemine (*Design Science Research*)
DTO	Andmeedastusobjekt (*Data Transfer Object*)
HTTP	Hüperteksti edastusprotokoll (*Hypertext Transfer Protocol*)
JDBC	Java andmebaasiühendus (*Java Database Connectivity*)
JPA	Java püsivusraamistik (*Java Persistence API*)
JSON	JavaScripti objektiesitus (*JavaScript Object Notation*)
JWT	JSON autentimistoken (*JSON Web Token*)
LLM	Suur keelemudel (*Large Language Model*)
MUI	Material UI kasutajaliidese raamistik (*Material UI*)
MVC	Mudel-vaade-kontroller arhitektuurimuster (*Model-View-Controller*)
MVP	Minimaalne töötav toode (*Minimum Viable Product*)
ORM	Objekt-relatsiooniline kaardistus (*Object-Relational Mapping*)
PDF	Porditav dokumendivorming (*Portable Document Format*)
RBAC	Rollipõhine ligipääsuhaldus (*Role-based Access Control*)
REST API	REST-arhitektuuristiilil põhinev rakendusliides (*Representational State Transfer Application Programming Interface*)
SMTP	Lihtne e-posti edastusprotokoll (*Simple Mail Transfer Protocol*)
SPA	Üheleherakendus (*Single-page Application*)
SQL	Struktureeritud päringukeel (*Structured Query Language*)
UI	Kasutajaliides (*User Interface*)

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Tänapäevane transport, suhteliselt taskukohased lennupiletid ning digitaalsete platvormide areng on märkimisväärselt suurendanud reisimise võimalusi võrreldes eelmise sajandiga, mil reisimine oli sageli luksuslik ja piiratud tegevus. Reisimine on muutunud tänapäeva ühiskonnas tavapäraseks osaks nii era- kui ka tööelus. [1]

Inimesed planeerivad puhkusereise, linnapuhkusi, seljakotireise ning grupireise sõprade ja tuttavatega. Kuigi turul on mitmeid reisi planeerimisega seotud digitaalseid tööriistu, toimub reiside korraldamine praktikas sageli mitmes erinevas keskkonnas korraga, mistõttu muutub kogu protsess kasutajate jaoks ebaefektiivseks, killustatuks ja raskemini hallatavaks.

Eriti keerukaks muutub reiside planeerimine organisatsioonide kontekstis, kus ühe või mitme inimese vastutada on suurema hulga osalejatega reisi korraldamine ja haldamine. Sellisel juhul tuleb lisaks transpordi ja majutuse planeerimisele hallata ka osalejaid, koordineerida muudatusi ning tagada, et kõik osapooled saaksid vajaliku teabe õigel ajal kätte. Sageli kaasneb selliste reisidega ka kulude jälgimine, dokumentide kogumine ning hilisem aruandlus, mis muudab protsessi ajamahukaks ja administratiivselt koormavaks.

Üheks levinud näiteks sellistest kasutusjuhtudest on Erasmus+ programmi raames toimuvad rahvusvahelised õpiränded ja noorteprojektid. Nende raames reisivad erinevatest Euroopa riikidest osalejad tavaliselt mõneks päevaks või nädalaks ühisesse sihtkohta, kus viiakse läbi koolitusi, noortevahetusi või muid koostöötegevusi. Kuna tegemist on Euroopa Liidu rahastatud programmidega, kaasneb selliste projektidega vajadus säilitada kuludokumente, hallata osalejate infot ning koostada aruandeid vastavalt rahastaja nõuetele. [2]

Käesolev lõputöö keskendub organisatsiooniliste reiside toetamisele ning töö eesmärgiks on arendada veebipõhine reisihaldussüsteem, mis koondab reiside planeerimise, osalejate haldamise, kulude jälgimise ja aruandluse ühtsesse keskkonda. Süsteemi eesmärk on vähendada erinevate tööriistade kasutamisest tulenevat killustatust ning muuta organisatsiooniliste reiside haldamine efektiivsemaks ja kasutajasõbralikumaks.

2 Probleem

Tänapäeval on reiside planeerimiseks ja korraldamiseks kasutusel mitmesugused digitaalsed tööriistad, mis toetavad erinevaid tegevusi, nagu reisiplaanide koostamine, info jagamine, dokumentide haldamine ning kulude jälgimine. Kuigi olemasolevad lahendused aitavad lahendada üksikuid ülesandeid, puudub sageli terviklik süsteem, mis koondaks kogu projektireiside korraldamise protsessi ühte keskkonda.

Käesoleva töö eesmärk on käsitleda killustatuse probleemi, mis puudutab reisiplaneerimise ja projektiga seotud infohalduse protsessi tervikuna. Projektireiside korraldamine toimub praktikas sageli mitme erineva tööriista ja keskkonna abil. Eraldi lahendusi kasutatakse näiteks reisiplaani koostamiseks, osalejate andmete haldamiseks, dokumentide säilitamiseks, projektis osalejatega suhtlemiseks ja kulude haldamiseks. Selle tulemusena on projektireisiga seotud info hajutatud erinevate rakenduste, failide ja suhtluskanalite vahel, mis muudab tervikliku ülevaate säilitamise keeruliseks ning vähendab tööprotsessi efektiivsust.

Üheks peamiseks probleemiks on info liikumine projektis osalejate ja korraldajate vahel. Sageli edastatakse oluline teave osalejatele e-posti teel, näiteks reisiajad, majutuse andmed, päevakavad või muudatused reisiplaanis. Kui informatsiooni saadetakse mitmes eraldi kirjas või erinevatel ajahetkedel, võib osa teabest jääda tähelepanuta, kaduda varasemasse kirjavahetusse või muutuda raskesti leitavaks. Eriti probleemseks muutub olukord siis, kui projektireisiga seotud detailid muutuvad ning osalejad peavad otsima uusimat infot varasemate kirjade ja failide hulgast.

Sarnane probleem esineb ka projektikorraldajate poolel. Korraldajatel tuleb sageli hallata reisiga seotud infot paralleelselt mitmes erinevas keskkonnas. Selline hajutatud infohaldus suurendab vigade tekkimise riski, dubleerib tegevusi ning muudab tööprotsessi ajamahukaks. Samuti puudub sageli kiire ülevaade sellest, milline info on ajakohane ning millised tegevused on juba tehtud.

Hajutatud töökorraldus tähendab, et kasutajad peavad töö käigus pidevalt liikuma erinevate

digitaalsete keskkondade vahel. 2004. aastal avaldatud uuringus leiti, et sagedane süsteemide vahetamine suurendab kasutaja kognitiivset koormust ning vähendab tööefektiivsust. Iga kord, kui kasutaja peab katkestama käimasoleva tegevuse ja liikuma teise süsteemi, kulub täiendavat aega uue keskkonna mõistmiseks ning varasema tööülesande juurde naasmiseks. [3]

Projektireiside korraldamisel väljendub see olukordades, kus reisikorraldajad peavad töö käigus pidevalt vahetama suhtlus-, dokumentide haldamise ja planeerimise keskkondade vahel. Selline mitmekeskkonnaline töövoog muudab tööprotsessi aeglasemaks, suurendab eksimuste võimalust ning raskendab ajakohase info säilitamist.

Probleem muutub veelgi olulisemaks rahvusvaheliste projektide puhul, kus ühe projekti raames tuleb koordineerida suuremat hulka osalejaid erinevatest riikidest. Sellistes olukordades on vajalik ühtne süsteem, kus oleks võimalik hallata reisiplaani, jagada ajakohast infot, jälgida muudatusi ning tagada, et kõik osapooled kasutaksid sama ajakohast teavet. Eraldi tööriistade kasutamisel muutub koostöö keerulisemaks ning suhtlus sõltub suurel määral käsitsi tehtavast koordineerimisest.

Üheks levinud näiteks sellistest kasutusjuhtudest on Erasmus+ programmi raames toimuvad noortevahetused ja õpiränded. Selliste projektide puhul lisandub tavapärasele reisiplaneerimisele ka vajadus koguda kuludokumente, säilitada tõendusmaterjale ning koostada hilisemaid aruandeid rahastajale. Kui vajalik info on hajutatud erinevatesse süsteemidesse, muutub ka aruandlusprotsess keeruliseks ja ajamahukaks. [4]

Kuigi olemasolevad tööriistad lahendavad üksikuid reisikorraldusega seotud ülesandeid, puudub terviklik ja kasutajasõbralik lahendus, mis ühendaks projektireiside planeerimise, koostöö, dokumentide haldamise ja aruandluse ühtsesse süsteemi. Sellise süsteemi puudumine põhjustab ebaefektiivsust, infokadu ning täiendavat halduskoormust nii projektikorraldajatele kui ka reisil osalejatele.

3 Uurimis- ja arendusstrateegia

Käesoleva töö eesmärk oli lahendada praktiline probleem, mis on seotud projektireiside planeerimise ja haldamise killustatud töökorraldusega. Töö keskendus ühe organisatsiooni tööprotsesside ja vajaduste analüüsimisele ning nende põhjal sobiva veebipõhise lahenduse kavandamisele ja arendamisele.

Töö käigus lähtuti disainiteaduslikust uurimislähenemisest (*Design Science Research*, DSR), mille puhul luuakse olemasoleva probleemi lahendamiseks toimiv tehniline artefakt ning hinnatakse selle sobivust reaalses kasutuskontekstis [5]. Käesoleva töö artefaktiks on veebipõhine reisihaldusrakendus koos tehisintellektil põhinevate funktsionaalsustega.

Arendusprotsess lähtus Peffersi jt kirjeldatud disainiteadusliku uurimismetoodika (*Design Science Research Methodology*, DSRM) etappidest. Metoodika järgi koosneb protsess kuuest peamisest etapist: probleemi tuvastamine, eesmärkide määratlemine, lahenduse kavandamine ja arendamine, demonstratsioon, hindamine ning tulemuste kommunikatsioon. [6]

Esimeses etapis kaardistati olemasolev probleem ning analüüsiti tellija tööprotsesse ja kasutatavaid tööriistu. Selle käigus tuvastati peamised kitsaskohad projektireiside planeerimisel, infohalduses, suhtluskorralduses ja aruandluses. Lisaks analüüsiti olemasolevaid reisiplaneerimise ja reisihalduse lahendusi, et hinnata nende sobivust tellija vajaduste kontekstis ning tuvastada funktsionaalsused, mida olemasolevad süsteemid piisaval määral ei toeta.

Teises etapis määratleti arendatava süsteemi eesmärgid ja nõuded. Eesmärgiks seati ühtse veebipõhise lahenduse loomine, mis võimaldaks hallata reisiplaane, osalejaid, kulusid, dokumente ja aruandlust ühes keskkonnas. Lisaks määratleti süsteemile funktsionaalsed ja mittefunktsionaalsed nõuded.

Kolmandas etapis kavandati süsteemi arhitektuur ning arendati veebirakendus koos klient-

rakenduse, tagarakenduse ja eraldiseisva tehisintellekti komponendiga. Arendusprotsessi käigus realiseeriti süsteemi põhifunktsionaalsused ning integreeriti erinevad tehnilised komponendid ühtseks tervikuks.

Neljandas etapis demonstreeriti valminud lahendust tellijale. Demonstreerimise eesmärk oli hinnata, kas süsteemi põhifunktsionaalsused vastavad kasutajate ootustele ning toetavad planeeritud töövooge reaalses kasutuskontekstis. Selle käigus koguti esmast tagasisidet kasutajaliidese, funktsionaalsuste ning üldise kasutuskogemuse kohta.

Viiendas etapis hinnati loodud lahenduse sobivust ja kasutatavust. Selleks viidi läbi valideerimine nii tellija kui ka tavakasutajatega. Tellija kasutas rakendust oma igapäeva töös. Kasutajatele anti stsenaariumipõhiseid ülesandeid ning koguti kvalitatiivset tagasisidet ja küsimustikupõhiseid hinnanguid. Selline lähenemine võimaldas hinnata nii süsteemi tehnilist toimivust kui ka kasutusmugavust reaalses kasutusolukorras.

Kuuendas etapis dokumenteeriti arendusprotsess, süsteemi arhitektuur, valideerimise tulemused ning töö peamised järeldused käesolevas bakalaureusetöös. Selline dokumenteerimine võimaldab loodud lahendust ja selle arendusprotsessi analüüsida, edasi arendada ning kasutada alusena võimalikele tulevastele uurimis- ja arendustöödele.

4 Taustauuring

Selleks et kaardistada arendatavale süsteemile esitatavad nõuded ja ootused, analüüsiti esmalt tellija senist tööprotsessi ning kasutatavaid tööriistu. See võimaldas tuvastada peamised kitsaskohad reiside ja projektidega seotud info haldamisel, suhtluskorralduses ning aruandluses.

Lisaks hinnati turul olemasolevaid reisiplaneerimise lahendusi, et selgitada välja nende tugevused, puudused ja sobivus tellija kasutusvajaduste kontekstis. Saadud tulemuste põhjal sõnastati süsteemile funktsionaalsed ja mittefunktsionaalsed nõuded, mis lõid aluse edasisele arendusprotsessile.

4.1 Olemasolev protsess tellija näitel

Tellija igapäevane töö rahvusvaheliste projektireiside planeerimisel ja haldamisel toimub hetkel mitme eraldiseisva tööriista abil. Erinevad andmed ja dokumendid on hajutatud erinevatesse keskkondadesse, mis muudab info haldamise keeruliseks ning ajamahukaks.

Reisidega seotud suhtlus toimub peamiselt e-posti teel, kasutades näiteks Gmaili või teisi e-posti keskkondi. E-kirjade kaudu jagatakse osalejatele reisiga seotud infot, saadetakse erinevaid kuludega seotud dokumente ning koordineeritakse tegevusi. Selline suhtlusviis põhjustab sageli olukordi, kus oluline teave jääb erinevatesse kirjavahetustesse ning selle hilisem leidmine võib olla keeruline.

Dokumentide haldamiseks kasutatakse pilvepõhiseid lahendusi, näiteks Google Docs ja Google Drive, kuhu salvestatakse reisiga seotud failid, nagu kuludega seotud failid ja tegevuste ajakava. Kuigi need tööriistad võimaldavad dokumente jagada ja koostöös redigeerida, puudub neil spetsiaalne struktuur reiside haldamiseks ning info võib olla hajutatud mitmesse kausta ja dokumenti.

Reisidega seotud kulude ja osalejate andmete haldamiseks kasutatakse sageli tabelarvu-

tustarkvara, näiteks Microsoft Excel või Google Sheets. Tabelid võimaldavad andmeid struktureerida, kuid nende käsitsi haldamine võib olla aeganõudev ning vigade tekkimise risk on suurem. Samuti puudub tabelipõhistes lahendustes struktureeritud aruandluse genereerimine ning rollipõhine ligipääsuhaldus. See tähendab, et aruandeid tuleb koostada käsitsi ning puudub võimalus piirata ligipääsu erinevatele andmetele vastavalt kasutaja rollile projektis. Tellija kasutuskontekstis on see oluline, kuna projektireisidel osalevad nii korraldajad, projektijuhid kui ka reisil osalejad, kellel on erinevad õigused ja vajadused süsteemis oleva informatsiooni kasutamiseks. Tabelipõhised lahendused ei võimalda sellist ligipääsu paindlikult hallata ning muutuvad suurema osalejate arvu korral raskemini hallatavaks.

Seetõttu on tellijal tekkinud vajadus ühtse veebipõhise süsteemi järele, mis koondaks kogu reisidega seotud info ühte keskkonda, millele oleks tagatud rollipõhine ligipääs ja mis lihtsustaks andmete haldamist ning võimaldaks koostada aruandeid efektiivsemalt.

4.2 Olemasolevad lahendused

Turul eksisteerib mitmeid reisiplaneerimise ja -haldusega seotud veebilahendusi, mis pakuvad erinevaid funktsionaalsusi alates lihtsast reisiplaani koostamisest kuni broneeringute ja kulude haldamiseni. Enamik olemasolevaid lahendusi on suunatud individuaalsetele reisijatele või väiksematele gruppidele ning ei arvesta spetsiifiliste organisatsiooniliste vajadustega.

Töö autorid analüüsisid valitud populaarseid platvorme ning hindasid nende sobivust tellija vajaduste kontekstis, milleks on rahvusvaheliste projektireiside (nt Erasmus+ projektide) planeerimine, osalejate haldamine ja aruandluse koostamine.

4.2.1 Wanderlog

Wanderlog on veebipõhine reisiplaneerimise keskkond, mis võimaldab kasutajatel luua reisiplaane, koostada päevakavasid ning hallata broneeringuid. Rakendus toetab ka reiside jagamist teiste kasutajatega ning pakub kaardipõhist visualiseerimist, mis lihtsustab marsruudi planeerimist ja tegevuste ajastamist. Kuigi lahendus on funktsionaalne ja kasutajasõbralik, on osa täiendavaid võimalusi piiratud tasulise versiooniga. Samuti on süsteem suunatud eelkõige individuaalsetele kasutajatele ning puudub rollipõhine ligipääsuhaldus,

mis on oluline suuremate gruppide ja organisatsiooniliste reiside haldamisel. [7]

Tellija hinnangul on selle lahenduse suurimaks puuduseks rollide puudumine, mis on nende kasutuskontekstis üks kriitilisemaid funktsionaalsusi. Lisaks võib kasutajaliides olla informatsioonirohke ja keeruline, mistõttu ei ole alati lihtne kiiresti olulist teavet leida ning ei ole võimalust ka projektis osalejate eest mingit tüüpi infot peita. Samuti puudub võimalus genereerida struktureeritud aruandeid või eksportida andmeid PDF-formaadis, mis on vajalik projektide dokumenteerimiseks ja aruandluseks.

4.2.2 Stippl

Stippl võimaldab kasutajatel luua reise, lisada päevapõhiseid tegevusi ning hallata reisi eelarvet. Reisi saab jagada teiste kasutajatega ning rakendus pakub ka marsruudi visualiseerimist kaardil. Lisaks sisaldab süsteem tehisintellektil põhinevat funktsionaalsust, mis aitab koostada soovitusi ja planeerida tegevusi. AI-funktsionaalsus on siiski kättesaadav peamiselt mobiilirakenduses ning piiratud premium-kasutajatele. Veebipõhises keskkonnas on selle kasutusvõimalused piiratud, mis vähendab süsteemi paindlikkust organisatsioonilise töö kontekstis. [8]

Tellija vaatenurgast puudub süsteemis rollipõhine kasutajahaldus ning samuti ei ole võimalik koostada ametlikke aruandeid või eksportida andmeid PDF-formaadis. Seetõttu ei vasta lahendus täielikult projektipõhise reisihalduse vajadustele.

4.2.3 Tripplanner.ai

Tripplanner.ai on tehisintellektil põhinev reisiplaneerimise tööriist, mis toimib vestlusliidese kaudu. Kasutaja saab suhelda süsteemiga ning genereerida automaatselt reisiplaani vastavalt sisestatud sihtkohale, ajaperioodile ja eelistustele. [9]

Kuigi lahendus pakub kiiret ja automatiseeritud planeerimist, on selle kasutusvõimalused piiratud reisiplaani detailse haldamise osas. Kasutajal puudub võimalus struktureeritult hallata kulusid, osalejaid või dokumente. Samuti ei sisalda süsteem rollipõhist kasutajahaldust ega aruandluse funktsionaalsust.

Seetõttu sobib lahendus pigem individuaalse reisi planeerimise abivahendiks kui terviklikuks projektireiside haldamise süsteemiks.

4.2.4 Perk

Perk on ettevõtetele suunatud reisihalduse ja kulude juhtimise platvorm, mis võimaldab hallata ärireise, kulusid ja aruandlust ühes süsteemis. Platvorm sisaldab rollipõhist kasutajahaldust ning võimaldab organisatsioonidel hallata reisidega seotud kulusid ja dokumente tsentraliseeritult. [10, 11]

Perki tugevuseks on terviklik ettevõtetele suunatud reisihalduse lähenemine ning lai valik kulude ja reiside administreerimise funktsionaalsusi. Samas keskendub süsteem peamiselt ärireiside logistilisele korraldamisele, näiteks lendude, hotellide ja transpordi broneerimisele ning ettevõtte reisipoliitikate haldamisele. Käesoleva töö kontekstis ei ole transpordi või majutuse broneerimine süsteemi peamine eesmärk, vaid fookus on projektireisidega seotud koostöö, infohalduse, aruandluse ja osalejate haldamise toetamisel. Lisaks ei paku süsteem detailset projektipõhist koostööfunktsionaalsust, näiteks ühiseid päevaplaane, tegevuste haldamist või projektispetsiifilist failihaldust, mis on olulised rahvusvaheliste projektide ja noortevahetuste kontekstis.

4.2.5 Booking.com for Business

Booking.com for Business on ettevõtetele suunatud reisihaldusplatvorm, mis võimaldab hallata tööreise, broneerida majutust, lende ja autorenti ning jälgida reisidega seotud kulusid ühes süsteemis. Platvorm toetab meeskondade haldamist, reiside ülevaateid ning aruandlust, võimaldades organisatsioonidel paremini hallata töötajate reisidega seotud tegevusi. [12, 13]

Lahenduse tugevuseks on ulatuslik broneerimisvõimekus ning kasutajasõbralik tööreiside haldamine. Samas keskendub süsteem peamiselt ärireiside logistilisele korraldamisele ega paku detailset projektipõhist koostööfunktsionaalsust, näiteks päevapõhist tegevuste haldust, projektidega seotud failide koondamist või paindlikku rollihaldust projektireiside kontekstis. Lisaks puuduvad süsteemis funktsioonid, mis toetaksid Erasmus+ tüüpi projektide aruandlusvajadusi ja dokumentatsiooni struktureeritud haldamist.

4.2.6 TripAdvisor

Tripadvisor on veebiplatvorm, mis keskendub eelkõige sihtkohtade, majutuste, restoranide ja vaatamisväärsuste arvustustele ning soovitustele. Kasutajad saavad lugeda teiste reisijate kogemusi ja hinnanguid, mis aitab teha informeeritud otsuseid reisiplaneerimisel. [14]

Kuigi platvorm pakub väärtuslikku informatsiooni ja inspiratsiooni, ei ole tegemist reisihaldussüsteemiga. Puudub võimalus hallata reisi osalejaid, koostada struktureeritud reisiplaane või jälgida kulusid. Samuti ei sisalda süsteem rollipõhist ligipääsuhaldust ega aruandluse funktsionaalsust.

4.2.7 Skyscanner

Skyscanner on veebipõhine otsingumootor, mis võimaldab kasutajatel leida ja võrrelda lennupileteid, majutust ning autorendivõimalusi. Platvormi kasutatakse sageli reiside planeerimise esimeses etapis, eriti transpordi leidmiseks. Kuigi Skyscanner on kasulik tööriist reisikulude optimeerimiseks ja broneeringute leidmiseks, ei ole tegemist reisihaldussüsteemiga. Süsteem ei võimalda hallata osalejaid, planeerida tegevusi ega koostada aruandeid. Seetõttu täidab see pigem toetavat rolli reisiplaneerimise protsessis. [15]

4.3 Olemasolevate lahenduste analüüs

Tabel 1. Olemasolevate reisiplaneerimise lahenduste võrdlus

Funktsionaalsus / omadus	**Wanderlog**	**Stippl**	**Tripplanner.ai**	**Perk**	**Booking.com**
Veebipõhine lahendus	Jah	Jah	Jah	Jah	Jah
Reisiplaani koostamine	Jah	Jah	Jah	Piiratud	Piiratud
Päevakava haldamine	Jah	Jah	Piiratud	Ei	Ei
Osalejate haldamine	Piiratud	Piiratud	Ei	Jah	Jah
Kulude haldamine	Piiratud	Jah	Ei	Jah	Piiratud
Rollipõhine kasutajahaldus	Ei	Ei	Ei	Jah	Piiratud
Koostöö mitme kasutajaga	Jah	Jah	Piiratud	Jah	Jah
Tehisintellekti funktsionaalsus	Tasuline	Tasuline	Jah	Ei	Ei
Aruannete genereerimine	Ei	Ei	Ei	Jah	Piiratud
PDF eksportimise võimalus	Ei	Ei	Ei	Piiratud	Ei
Sobivus organisatsioonidele	Piiratud	Piiratud	Madal	Kõrge	Keskmine
Sobivus projektireiside haldamiseks	Piiratud	Piiratud	Madal	Keskmine	Madal

Tabelis 1 on esitatud analüüsitud reisiplaneerimise ja reisihalduse lahenduste võrdlus organisatsiooniliste projektireiside kontekstis. Võrdlusesse valitud süsteemid jagunevad üldjoontes kaheks: individuaalsetele kasutajatele suunatud reisiplaneerimise rakendused ning ettevõtetele ja organisatsioonidele mõeldud tööreiside haldamise platvormid.

Wanderlog, Stippl ja Tripplanner.ai on suunatud eelkõige individuaalsete reiside või väiksemate gruppide planeerimisele. Kõik kolm lahendust võimaldavad koostada reisiplaane ning hallata reisiga seotud tegevusi. Wanderlog ja Stippl toetavad ka mitme kasutaja koostööd, mis lihtsustab reisiplaanide jagamist ja ühiste tegevuste organiseerimist väiksemates gruppides.

Nende lahenduste peamisteks tugevusteks võib pidada kasutajasõbralikkust, kaardipõhist visualiseerimist ning võimalust koondada reisiga seotud tegevused ühte keskkonda. Lisaks sisaldavad Stippl ja Tripplanner.ai tehisintellektil põhinevaid funktsioone, mis aitavad automatiseerida reisiplaanide koostamist ning vähendada käsitsi tehtavat tööd. Sellised omadused muudavad süsteemid mugavaks ja ajasäästlikuks tavakasutajatele, eriti olukordades, kus soovitakse kiiresti koostada esialgset reisiplaani või saada inspiratsiooni sihtkoha tegevuste kohta.

Samas ilmnes võrdluse käigus, et individuaalsetele kasutajatele suunatud lahendused ei paku piisavat tuge organisatsiooniliste projektireiside haldamiseks. Tabelist 1 on näha, et Wanderlog, Stippl ja Tripplanner.ai ei toeta rollipõhist kasutajahaldust ega struktureeritud aruandlust. Samuti puudub neil võimalus koostada projektipõhiseid aruandeid või eksportida andmeid PDF-formaadis, mis on oluline projektide dokumenteerimise ja finantsaruandluse seisukohalt. Lisaks võivad kasutajaliidesed olla suure infomahu tõttu vähem ülevaatlikud olukordades, kus samaaegselt tuleb hallata suuremat hulka osalejaid ja projektidega seotud informatsiooni.

Perk ja Booking.com for Business erinevad eelnevatest lahendustest selle poolest, et nende peamine eesmärk on ettevõtete tööreiside logistiline korraldamine. Mõlemad süsteemid võimaldavad hallata ärireise, broneerida majutust ja transporti ning toetavad organisatsioonisisest reisihaldust. Perk sisaldab lisaks ka rollipõhist kasutajahaldust ning kulude ja aruandluse funktsionaalsusi, mis muudab selle organisatsiooniliste protsesside seisukohalt terviklikumaks lahenduseks kui klassikalised reisiplaneerimise rakendused.

Kuigi Perk ja Booking.com for Business pakuvad ettevõtetele suunatud reisihalduse funktsionaalsusi, keskenduvad need peamiselt transpordi, majutuse ja tööreiside logistilisele korraldamisele. Käesoleva töö kontekstis ei ole peamiseks probleemiks broneerimine, vaid projektireisidega seotud koostöö, infohaldus, osalejate koordineerimine ning aruandluseks vajalike dokumentide koondamine ühte keskkonda. Lisaks puudub ka nendes süsteemides detailne projektipõhine koostööfunktsionaalsus, näiteks ühised päevaplaanid, tegevuste haldamine ja projektispetsiifiline failihaldus.

Enamik analüüsitud lahendusi on seega suunatud kas individuaalsetele kasutajatele või klassikalistele ärireisidele ning ei arvesta olukorraga, kus samaaegselt tuleb hallata mitut projekti, suuremat hulka osalejaid ning projektipõhiseid kulusid. Seetõttu ei paku olemasolevad süsteemid piisavat tuge organisatsiooniliste protsesside automatiseerimiseks ning nõuavad jätkuvalt käsitsi andmete haldamist teistes tööriistades.

Reisikorraldajate vajadustele lisaks tuleb arvestada ka projektil osalejate vajadustega. Osalejate jaoks on olulisem kasutusmugavus ja kiire info kättesaadavus kui keerukad haldusfunktsioonid. Seetõttu on oluline, et süsteem oleks piisavalt paindlik, et toetada mõlemat poolt. See tähendab, et süsteem peaks võimaldama reisiplaani koostamist ja info jagamist ka kasutajatele, kellel puudub vajadus detailse projektihalduse või aruandluse järele.

Lisaks organisatsioonilistele projektireisidele on oluline arvestada ka individuaalreiside planeerimise kontekstiga, kus reisija korraldab oma reisi iseseisvalt või väikese seltskonnaga. Sellisel juhul on peamisteks vajadusteks reisiplaani koostamine, tegevuste ajastamine, kulude haldamine ning vajaliku info koondamine ühte keskkonda.

Individuaalreiside puhul ei ole tavaliselt vajalik keerukas rollipõhine kasutajahaldus ega detailne aruandlus, mistõttu võivad lihtsamad ja kasutajasõbralikumad lahendused olla piisavad. Samas võib tehisintellektil põhinev tööriist, nagu Tripplanner.ai, olla kasulik inspiratsiooni ja esialgse reisiplaani genereerimiseks, eriti olukordades, kus kasutaja ei ole sihtkohaga varasemalt tuttav.

Seetõttu on oluline, et süsteem toetaks lisaks projektireiside haldamisele ka individuaalreiside planeerimist, pakkudes lihtsat ja selget kasutajaliidest ning võimalust koostada paindlikke reisiplaane. Selline funktsionaalsus suurendab süsteemi kasutusvõimalusi ning

võimaldab rakendust kasutada erinevates reisiplaneerimise olukordades.

Kokkuvõttes võib tabeli 1 põhjal järeldada, et kuigi olemasolevad lahendused pakuvad mitmeid kasulikke funktsionaalsusi reiside planeerimiseks ja tööreiside haldamiseks, ei vasta need täielikult käesoleva töö kasutusstsenaariumi vajadustele. Sellest tulenevalt on põhjendatud uue süsteemi arendamine, mis koondab reiside planeerimise, osalejate haldamise, kulude jälgimise, koostöö ja aruandluse ühte keskkonda ning toetab projektipõhist rollihaldust ja dokumentatsiooni haldamist.

4.4 Nõuded

Arendatavale süsteemile esitatavad nõuded kujunesid tellija vajaduste, olemasolevate lahenduste analüüsi ning töö käigus läbiviidud arutelude põhjal. Süsteemile seatud nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks nõueteks.

4.4.1 Funktsionaalsed nõuded

Käesolev süsteem peab toetama järgmisi funktsionaalseid nõudeid:

- Süsteem peab võimaldama kasutajatel registreerida konto ning sisse logida autentitud kasutajana.
- Süsteem peab võimaldama kasutajatel luua, vaadata, muuta ja kustutada kahte tüüpi reisi - tavareis ja projektireis.
- Süsteem peab võimaldama koostada reiside päevapõhiseid ajakavasid ning lisada tegevusi.
- Süsteem peab võimaldama kasutajatel kutsuda teisi osalejaid reisile e-posti teel ning rakenduse siseselt.
- Süsteem peab toetama rollipõhist ligipääsuhaldust, kus erinevatel kasutajatel on erinevad õigused.
- Süsteem peab võimaldama hallata reisiga seotud kulusid ning siduda need konkreetsete osalejatega.
- Süsteem peab võimaldama üles laadida kuludega seotud dokumente (nt arved ja kviitungid).
- Süsteem peab võimaldama genereerida reisiga seotud aruandeid struktureeritud kujul.

- Süsteem peab võimaldama eksportida aruandeid PDF-formaadis.
- Süsteem peab võimaldama kasutajatel vaadata ja hallata oma profiili ning seadeid.
- Süsteem peab võimaldama saata teavitusi kasutajatele oluliste muudatuste korral (nt ajakava uuendused).
- Süsteem peab pakkuma tehisintellektil põhinevat funktsionaalsust reisiplaanide ja soovituste genereerimiseks.

4.4.2 Mittefunktsionaalsed nõuded

Lisaks funktsionaalsetele nõuetele peab süsteem vastama järgmistele mittefunktsionaalsetele nõuetele:

- Süsteem peab tagama piisava jõudluse, võimaldades kasutajatel sooritada põhilisi toiminguid ilma märgatava viivituseta.
- Süsteem peab olema turvaline, kaitstes kasutajate andmeid autentimise ja autoriseerimise mehhanismidega.
- Süsteem peab olema kasutajasõbralik ning intuitiivne, võimaldades kasutajatel funktsioone kasutada ilma põhjaliku juhendamiseta.
- Süsteem peab olema skaleeritav, et toetada kasutajate arvu ja andmemahtude kasvu tulevikus.
- Süsteem peab olema töökindel ning tagama andmete säilimise ka võimalike vigade või katkestuste korral.
- Süsteem peab olema hooldatav, võimaldades arendajatel koodi lihtsasti täiendada ja muuta.
- Süsteem peab tagama ühilduvuse erinevate seadmete ja veebilehitsejatega, et kasutajad saaksid rakendust kasutada erinevates keskkondades.

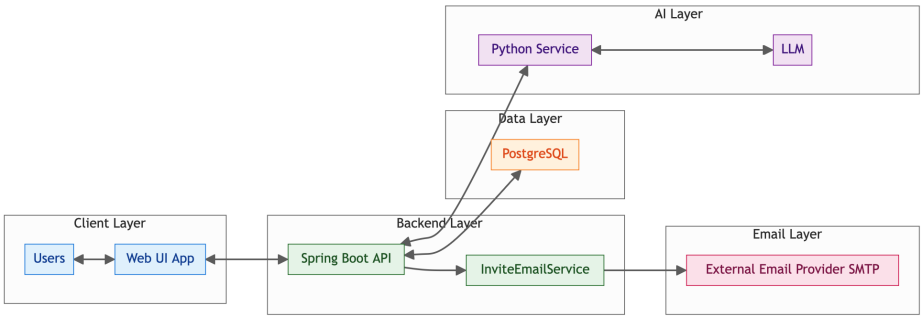
5 Süsteemi disain ja arhitektuur

Süsteem on realiseeritud mitmekihilise veebirakendusena, mis koosneb klientrakendusest, tagarakendusest, andmebaasist ning eraldiseisvast tehisintellekti komponendist. Klientrakendus vastutab kasutajaliidese ja kasutajaga suhtlemise eest. Tagarakendus haldab äriloogikat, autentimist, andmete töötlemist ning suhtlust andmebaasi ja väliste teenustega. Andmete püsivaks salvestamiseks kasutatakse relatsioonilist andmebaasi PostgreSQL. Tehisintellekti komponent on realiseeritud eraldi teenusena, mis suhtleb põhirakendusega HTTP-päringute kaudu.

Selline arhitektuur võimaldab süsteemi komponente arendada ja hooldada sõltumatult ning toetab lahenduse skaleeritavust tulevikus.

5.1 Süsteemi arhitektuur

Joonisel 1 on esitatud süsteemi üldine arhitektuur. Süsteem koosneb veebipõhisest klientrakendusest, tagarakendusest, PostgreSQL andmebaasist, eraldiseisvast tehisintellekti teenusest ning välise SMTP teenusepakkujaga suhtlevast e-posti kihist.



Joonis 1. Süsteemi üldine arhitektuur

Süsteemi arhitektuur järgib mitmekihilise arhitektuuri põhimõtteid, kus kasutajaliides, äriloogika, andmehaldus ja tehisintellekti funktsionaalsused on eraldatud iseseisvateks kihtideks ja komponentideks. Selline lähenemine võimaldab vähendada süsteemi erinevate osade vahelist sidusust ning parandab rakenduse hooldatavust, testitavust ja edasiaren-

datavust. Vastutuste eraldamine erinevate kihtide vahel võimaldab teha muudatusi ühes süsteemi osas ilma, et see mõjutaks otseselt ülejäänud komponente. [16]

Arhitektuuris on kasutatud klient-server mudelit, kus kasutajaliides ja äriloogika on eraldatud erinevateks komponentideks. Klientrakendus vastutab kasutajaliidese kuvamise ning kasutajapoolsete tegevuste edastamise eest, samas kui tagarakendus tegeleb äriloogika, autentimise, andmete töötlemise ja süsteemi sisemiste töövoogude haldamisega. Selline lähenemine võimaldab klientrakendust ja tagarakendust arendada ning skaleerida sõltumatult. Komponentide omavaheline suhtlus toimub REST API kaudu, mis toetab süsteemi tehnoloogilist sõltumatust ja hajutatud arhitektuuri põhimõtteid [17].

Andmehaldus on koondatud eraldiseisvasse relatsioonilisse andmebaasi, mille kaudu hallatakse kasutajate, reiside, tegevuste, kulude ja dokumentidega seotud andmeid. Keskne andmebaas võimaldab tagada süsteemi andmete terviklikkuse ning vähendab dubleeriva info tekkimise riski.

Tehisintellektil põhinevad funktsionaalsused on realiseeritud eraldiseisva teenusena, mis suhtleb põhirakendusega HTTP-põhiste päringute kaudu. Selline arhitektuurne eraldamine võimaldab AI-komponenti arendada ja hallata sõltumatult ülejäänud süsteemist ning kasutada vajadusel erinevat tehnoloogilist ökosüsteemi. Teenuste eraldamine vähendab süsteemi komponentide omavahelist sõltuvust ning toetab modulaarsust ja skaleeritavust [18].

E-kirjade saatmise funktsionaalsus on realiseeritud eraldi teenusena tagarakenduse kihis, mis suhtleb välise e-posti teenusepakkujaga SMTP protokolli kaudu. Selline eraldamine võimaldab hoida e-kirjade koostamise ja saatmise loogika eraldi ülejäänud äriloogikast ning lihtsustab vajadusel e-posti teenusepakkuja vahetamist või laiendamist.

Valitud arhitektuur võimaldab süsteemi erinevaid komponente sõltumatult laiendada ja hooldada ning loob aluse süsteemi edasiseks funktsionaalseks ja tehniliseks arendamiseks.

5.2 Klientrakendus

Klientrakendus on realiseeritud üheleherakendusena (SPA), mille eesmärk on pakkuda kasutajale kiiret ja ühtset töövoogu reiside haldamiseks.

5.2.1 Tehnoloogiad

Klientrakendus on ehitatud React 19 ja TypeScripti peale. Kasutajaliidese komponentide loomiseks kasutatakse Material UI (MUI) raamistikku, millele on lisatud projekti enda visuaalne teema, sealhulgas värvipalett, tüpograafia ja kujunduse stiilireeglid. HTTP-päringute tegemiseks kasutatakse Axios klienti, mille kaudu suheldakse tagarakenduse REST API-ga.

React valiti klientrakenduse tehnoloogiaks selle komponentpõhise arhitektuuri, laialdase kasutuse ning tugeva ökosüsteemi tõttu [19]. Komponentpõhine lähenemine võimaldab luua korduvkasutatavaid kasutajaliidese elemente ning parandab suurema rakenduse hooldatavust ja laiendatavust. Lisaks võimaldab React eraldada kasutajaliidese väiksemateks iseseisvateks komponentideks, mis lihtsustab arendust ja testimist.

TypeScript valiti JavaScripti asemel eelkõige staatilise tüübisüsteemi tõttu, mis võimaldab tuvastada osa vigadest juba kompileerimise ajal ning parandab suurema koodibaasi arusaadavust ja hooldatavust. Tugev tüübisüsteem oli oluline eriti rakenduse keerukamate andmestruktuuride ja API vastuste käsitlemisel.

HTTP-kliendina kasutatakse Axios teeki, mis võimaldab hallata päringuid tsentraalselt ning lihtsustab autentimispäiste, veahandluse ja interceptorite kasutamist. Selline lähenemine vähendab dubleerivat koodi ning parandab suhtlust tagarakenduse REST API-ga.

5.2.2 Disain ja kasutajaliides

Kasutajaliides on üles ehitatud komponentpõhiselt, mis lihtsustab funktsionaalsuste laiendamist ja hooldust. Kujundus kasutab keskset teemat, mille kaudu on defineeritud värviskeemid nii heleda kui tumeda režiimi jaoks. See tagab visuaalse järjepidevuse eri vaadete vahel (nt reiside nimekiri, reisivaade, seadete vaade).

Kasutajaliidese kujundamisel lähtuti põhimõttest, et süsteemi kasutavad ka mittetehnilise taustaga kasutajad. Seetõttu eelistati selget navigeerimist, loogilist infohierarhiat ning võimalikult vähest kasutaja kognitiivset koormust. Ekraanipildid kasutajaliidesest on toodud lisas 8.

Navigeerimine on teostatud SPA-loogikaga, kus vaadete vahetus toimub ilma täislehe

uuestilaadimiseta.

5.3 Tagarakendus

Tagarakendus on realiseeritud Javas, kasutades Spring Boot raamistikku versiooniga 4.0.2. Java valiti peamise programmeerimiskeelena selle laialdase ettevõttetasemel kasutuse, stabiilsuse ning tugeva standardiseeritud teekide ja raamistike ökosüsteemi tõttu. Java virtuaalmasinal põhinev arhitektuur võimaldab head platvormiülesust ning pikaajalist tuge, mis muudab selle sobivaks suuremate ja hooldatavate serverirakenduste arendamiseks. Lisaks pakub Java tugev tüübisüsteem ja küps ökosüsteem head tuge keerukamate äriloogikaga süsteemide realiseerimiseks.

Spring Boot raamistik valiti eelkõige, sest see vähendab oluliselt rakenduse konfiguratsiooni hulka ning võimaldab kiiresti arendada REST-põhiseid veebiteenuseid. Raamistik pakub sisseehitatud lahendusi sõltuvuste haldamiseks, turvalisuseks, andmebaasiga suhtlemiseks ja konfiguratsiooniks, mis vähendab arenduse keerukust ning võimaldab keskenduda süsteemi äriloogikale. Lisaks toetab Spring Boot hästi modulaarsust ja kihilist arhitektuuri, mis oli oluline süsteemi pikaajalise hooldatavuse seisukohalt. [20]

Rakendus kasutab Spring Web MVC moodulit REST API realiseerimiseks. REST-põhine lähenemine võimaldab eraldada klientrakenduse tagarakendusest ning toetab süsteemi tehnoloogilist sõltumatust. Selline arhitektuur lihtsustab erinevate klientrakenduste ühendamist sama backend teenusega ning toetab süsteemi laiendatavust tulevikus.

Autentimise ja autoriseerimise realiseerimiseks kasutatakse Spring Security moodulit. Spring Security pakub laialdaselt kasutatud ja hästi testitud turvaarhitektuuri, mis toetab autentimist, õiguste haldamist ning ligipääsukontrolli [21]. Kuna süsteemis hallatakse kasutajate isiklikke andmeid, projektireiside informatsiooni ja kuludokumente, oli oluline kasutada standardiseeritud ja turvalist lahendust kasutajate autentimiseks ning rollipõhise ligipääsu kontrollimiseks.

Andmebaasiga suhtlemiseks kasutatakse Spring Data JPA raamistikku, mis toetab objekt-relatsioonilist kaardistamist (ORM). ORM-lähenemine võimaldab käsitleda andmebaasi objekte rakenduse domeenimudeli kaudu ning vähendab vajadust kirjutada käsitsi suurel hulgal SQL-päringuid. See parandab koodi loetavust, vähendab dubleerimist ning lihtsustab

andmehaldusega seotud äriloogika realiseerimist.

Autentimine on realiseeritud JWT (JSON Web Token) mehhanismi abil. JWT võimaldab stateless autentimist, kus server ei pea säilitama kasutajaseansse [22]. Selline lähenemine sobib hästi REST API arhitektuuriga ning toetab süsteemi skaleeritavust ja hajutatud ülesehitust. Lisaks võimaldab JWT-põhine autentimine hoida klient- ja serverrakenduse vahelise suhtluse lihtsama ja sõltumatumana.

Tagarakenduse arhitektuur järgib kihilist ülesehitust, kus kood on organiseeritud pakettidesse `controller`, `service`, `repository` jne. Selline kihiline arhitektuur toetab vastutuste eraldamist ning aitab vähendada erinevate süsteemiosade omavahelist sidusust [23]. Äriloogika, andmehaldus ja HTTP-päringute töötlemine on hoitud eraldi kihtides, mis parandab süsteemi hooldatavust, testitavust ja laiendatavust.

5.3.1 Rollid ja autoriseerimine

Rakenduses on realiseeritud rollipõhine ligipääsukontroll (RBAC), mille eesmärk on piirata kasutajate õiguseid vastavalt nende rollile konkreetse reisi kontekstis. RBAC on laialdaselt kasutatav autoriseerimismudel, kus õigused seotakse rollidega ning kasutajatele omistatakse vastavad rollid [24]. Selline lähenemine võimaldab hallata õiguseid tsentraalselt ning vähendab keerukust võrreldes mudeliga, kus õiguseid hallatakse individuaalselt iga kasutaja jaoks.

RBAC mudel valiti käesoleva süsteemi autoriseerimislahenduseks, kuna projektireiside haldamisel esinevad selgelt eristatavad kasutajarollid erinevate vastutus- ja õigustasemetega. Näiteks vajavad mõned kasutajad ainult reisiandmete vaatamise õigust, samas kui teised peavad saama hallata kulusid, kasutajaid või kogu reisi konfiguratsiooni. Rollipõhine lähenemine võimaldab sellised õigused standardiseerida ning rakendada ühtselt kogu süsteemis.

Lisaks võimaldab RBAC lihtsustada uute kasutajate lisamist süsteemi, kuna kasutajale rakenduvad automaatselt tema rolliga seotud õigused. Samuti muudab see süsteemi hooldatavamaks ja laiendatavaks, sest uute operatsioonide lisamisel piisab nende sidumisest olemasolevate rollidega [25].

Rakenduses realiseeritud autoriseerimismudel on domeenipõhine, kuna kasutaja roll sõltub konkreetsest reisist. See võimaldab samal kasutajal omada erinevates projektireisides erinevaid õigustasemeid. Käesolevas süsteemis on realiseeritud viis erinevat rolli, mis on loetletud tabelis 2.

Tabel 2. Süsteemis realiseeritud rollid ja nende õigused.

Roll	**Õigused**
`PARTICIPANT`	Saab vaadata reisi põhiteavet ja reisikava. Ei näe teiste kasutajate kulusid ega detailset finantsinfot.
`VIEWER`	Saab vaadata kõiki reisi andmeid, sealhulgas kulusid, asukohti ja osalejaid, kuid ei saa andmeid muuta.
`CONTRIBUTOR`	Omab kõiki `VIEWER` õiguseid ning saab muuta reisikava, lisada asukohti ja kulusid ning laadida üles kuludega seotud faile.
`ADMIN`	Omab kõiki `CONTRIBUTOR` õiguseid ning saab hallata kasutajaid, muuta rolle, eksportida reisi andmeid ning hallata failide ja reisipiltide kustutamist.
`OWNER`	Reisi looja, kellel on kõik süsteemi õigused, sealhulgas õigus reis kustutada. Rolli ei ole võimalik eemaldada.

Autoriseerimine toimub domeenipõhiselt reisi tasemel. Iga kasutaja roll salvestatakse konkreetse reisi kontekstis andmebaasi tabelisse `trip_invites`, kus hoitakse lisaks kasutaja rollile ka kutse staatust. Pärast kutse vastuvõtmist (`ACCEPTED`) rakenduvad kasutajale vastava rolli õigused. Selline lähenemine võimaldab sama kasutajat siduda erinevate reisidega erinevate õigustasemete alusel.

Õiguste kontrollimiseks on realiseeritud eraldi teenus `TripAuthorizationService`, mida kasutavad kõik peamised äriloogika teenused enne tundlike operatsioonide teostamist. Teenuse ülesandeks on kontrollida:

- kasutaja identiteeti;
- kasutaja rolli konkreetsel reisil;
- operatsiooni jaoks vajalikku minimaalset õigustaset.

Õiguste kontrolli tsentraliseerimine vähendab dubleeriva loogika hulka ning tagab kogu süsteemis ühtse autoriseerimiskäitumise. Lisaks lihtsustab selline arhitektuuriline lahendus süsteemi laiendamist ja hooldamist.

Operatsioonide ja vajalike rollide seosed on defineeritud klassis `TripPermissions`, kus iga süsteemi operatsioon on seotud minimaalse vajaliku rolliga. Näiteks operatsiooni `TRIP_VIEW` teostamiseks piisab rollist `PARTICIPANT`, kulude lisamiseks (`EXPENSE_ADD`) on vajalik vähemalt roll `CONTRIBUTOR` ning reisi kustutamiseks (`TRIP_DELETE`) roll `OWNER`.

Rakenduses kontrollitakse õiguseid mitmete erinevate operatsioonide puhul, sealhulgas:

- reisi vaatamine ja muutmine;
- kasutajate kutsumine ja rollide muutmine;
- reisikava elementide lisamine ja muutmine;
- kulude lisamine ja kustutamine;
- failide üleslaadimine ja kustutamine;
- reisiandmete eksportimine.

Autoriseerimissüsteem on integreeritud kõikide peamiste moodulitega, sealhulgas reiside, kulude, failide ja reisikava haldusega. Selline lahendus võimaldab tagada, et kasutaja saab teha ainult talle määratud rollile vastavaid tegevusi ning aitab vähendada volitamata ligipääsu riski süsteemi andmetele.

5.3.2 Reiside ja projektireiside eristamine

Süsteemis eristatakse kahte erinevat reisi tüüpi: tavareis (*trip*) ja projektireis (*project*). Selline eristus loodi, kuna süsteemi kasutusstsenaariumid erinevad oluliselt sõltuvalt sellest, kas tegemist on väiksema isikliku või grupireisiga või organisatsioonilise projektireisiga.

Süsteemi kavandamisel lähtuti põhimõttest, et kasutajale kuvatakse ainult tema kasutusstsenaariumi jaoks vajalik funktsionaalsus. Tavakasutajatele suunatud vaadetes on vähendatud keerukamate haldus- ja projektifunktsioonide hulka, et muuta kasutajaliides selgemaks ja lihtsamini kasutatavaks. Selline lähenemine aitab vähendada kasutaja kognitiivset koormust ning võimaldab kasutajatel keskenduda peamistele tegevustele liigse informatsioonita. Projektireiside puhul on seevastu vajalik kuvada rohkem haldus- ja koostööfunktsionaal-

susi, kuna nende kasutuskontekst eeldab detailsemat rollihaldust, aruandlust ja osalejate koordineerimist.

Üheks peamiseks erinevuseks on aruannete genereerimise võimalus, mis on vajalik näiteks Erasmus+ projektide dokumenteerimiseks ja hilisemaks aruandluseks. Tavareiside puhul sellist funktsionaalsust ei pakuta, kuna individuaalsete reiside kontekstis ei ole struktureeritud aruandlus üldjuhul vajalik.

Lisaks sisaldavad projektireisid eraldi *Important Info* sektsiooni, kuhu saab lisada projektiga seotud olulist teavet, näiteks juhiseid, muudatusi või osalejatele vajalikku infot. Süsteem võimaldab selle sektsiooniga seotud uuenduste korral saata osalejatele automaatseid teavitusi e-posti teel. Selline lahendus aitab tagada, et kõik projektis osalejad kasutavad ajakohast informatsiooni ning vähendab killustunud suhtlusest tekkivaid probleeme.

Oluline erinevus seisneb ka kasutajate õiguste mudelis. Projektireiside puhul kasutatakse rollipõhist ligipääsuhaldust, kus kasutajatel võivad olla erinevad õigused vastavalt nende vastutusele projektis. Tavareiside puhul käsitletakse kõiki osalejaid automaatselt `ADMIN` õigustes kasutajatena, kuna väiksemates ja mitteformaalsetes reisides puudub vajadus detailse rollihalduse järele. Selline lähenemine võimaldab hoida tavareiside kasutuskogemuse lihtsamana, säilitades samal ajal projektireiside jaoks vajalikud haldusvõimalused.

5.3.3 Automaattestid

Tagarakenduse testimiseks kasutatakse JUnit 5 testiraamistikku koos Spring Boot testimoodulitega, Mockito ja AssertJ teekidega. Testid jagunevad ühiktestideks ja integratsioonitestideks.

Testidega on kaetud mitmed süsteemi komponendid, sealhulgas teenusekiht (näiteks InviteServiceTest, NotificationServiceTest ja IntelligenceServiceTest), kontrollerite käitumine (ControllerUnitTest), turvafilter (JwtRequestFilterTest) ning erindite käsitlemine (GlobalExceptionHandlerTest). Koodikatvus on ligikaudu 90%.

5.3.4 Google kontoga autentimine

Lisaks tavapärasele e-posti ja parooliga autentimisele toetab süsteem sisselogimist Google konto kaudu. Kolmanda osapoole autentimise kasutamise eesmärk on lihtsustada kasu-

tajakonto loomise ja sisselogimise protsessi ning vähendada vajadust eraldi paroolide haldamiseks.

Google autentimine on realiseeritud OAuth 2.0 protokolli abil, mis võimaldab kasutajal anda rakendusele piiratud ligipääsu oma kasutajainfole ilma parooli otse rakendusele edastamata. Autentimisprotsessi käigus suunatakse kasutaja Google autentimisteenusesse, kus toimub kasutaja identiteedi kontroll. Eduka autentimise järel väljastatakse rakendusele autentimistoken, mille põhjal seotakse kasutaja süsteemis olemasoleva kontoga või luuakse uus kasutajakonto. [26]

Kolmanda osapoole autentimise kasutamine parandab kasutusmugavust ning vähendab paroolidega seotud turvariske, kuna süsteem ei pea käsitlema kasutaja Google konto paroole. Lisaks võimaldab selline lahendus kiirendada kasutajate liitumisprotsessi ning vähendada autentimisega seotud kasutajapoolset kognitiivset koormust.

5.3.5 Aruannete eksportimine

Süsteem võimaldab genereerida reisidega seotud aruandeid PDF-formaadis, et lihtsustada projektidega seotud dokumentatsiooni koondamist, aruandlust ning kuludokumentide esitamise protsessi rahastajatele ja partnerorganisatsioonidele.

PDF-aruannete genereerimine toimub klientrakenduse poolel. Klientrakendus kogub tagarakendusest REST API kaudu aruande koostamiseks vajalikud andmed, sealhulgas reisi üldinfo, osalejad, kulud ning seotud failide metaandmed. Saadud andmete põhjal genereeritakse kasutajale PDF-formaadis aruanne.

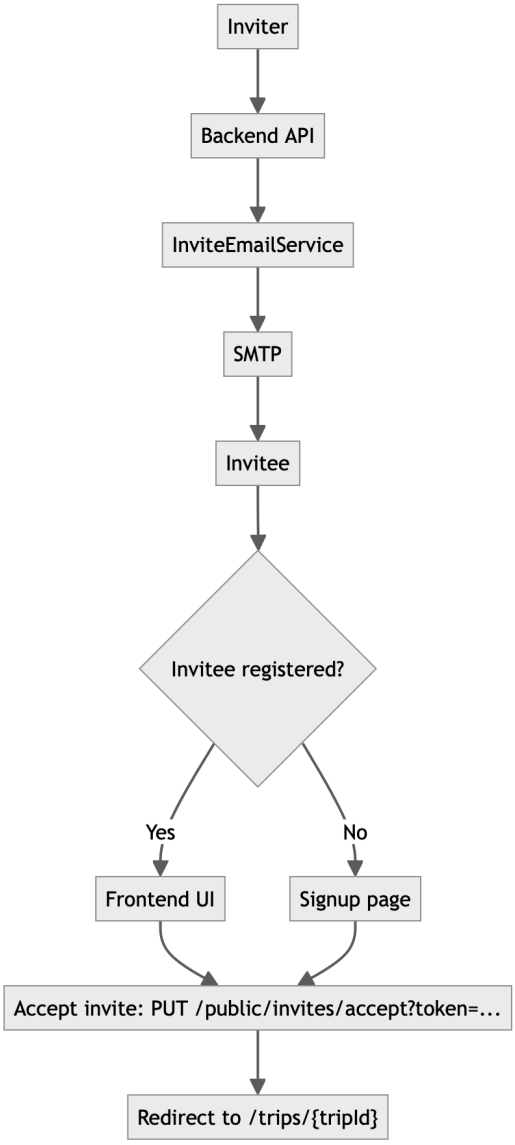
PDF-formaat valiti selle laialdase toe, platvormiülesuse ning dokumentide kujunduse säilitamise tõttu [27]. PDF-failide töötlemiseks ja kuvamiseks kasutatakse `pdfjs-dist` teeki, mis põhineb Mozilla PDF.js projektile. Teek võimaldab PDF-dokumente brauseris kuvada ning pakkuda kasutajale aruande eelvaadet enne faili allalaadimist.[28]

Lisaks aruande põhisisule toetab süsteem reisiga seotud failide kaasamist ekspordiprotsessi. See võimaldab koondada aruandluseks vajalikud dokumendid, näiteks kuludokumendid, kviitungid ja muud tõendusmaterjalid, ühtsesse eksporditavasse kogumisse. Selline lähenemine vähendab vajadust dokumente käsitsi eraldi otsida ja hallata ning toetab projektireiside

aruandluse töövoogu.

5.3.6 E-kirjade saatmine

Joonisel 2 on esitatud e-kirjade saatmise loogika. Käesolevas süsteemis toimub meilide saatmine tagarakenduse sees ning eraldi e-posti mikroteenust ei kasutata. E-kirjade saatmine on realiseeritud teenuses `InviteEmailService`, mis kasutab kirjade edastamiseks `JavaMailSender` liidest ja SMTP-protokolli. SMTP ühenduse parameetrid (host, port, autentimine, kasutajanimi, parool) määratakse keskkonnamuutujatega.



Joonis 2. E-kirjade saatmise loogika

Süsteemis saadavad e-kirju järgmised funktsionaalsused:

- Reisikutse saatmine – kutse loomisel käivitab `InviteService` meetodi `sendInviteEmail(...)`. Kirja sisu sisaldab kutse linki: registreeritud kasutajale suunatakse kutse kinnitamise vaatesse, registreerimata kasutajale konto loomise vaatesse koos kutse tokeniga.
- Projekti info-uuenduste teavitused – projekti infosektsiooni muudatustest teavitamisel käivitab `NotificationService` meetodi `sendProjectUpdateEmail(...)`. Kiri saadetakse osalejatele, kellel e-posti teavitused on lubatud.

5.4 Andmebaasi disain

Süsteemi andmehaldus on realiseeritud relatsioonilise andmebaasi abil, kuna rakenduses hallatakse suurel hulgal omavahel seotud andmeid, nagu kasutajad, reisid, osalejad, kulud, päevaplaanid ja teavitused. Relatsiooniline andmemudel võimaldab kirjeldada tabelite vahelisi seoseid struktureeritud kujul ning tagada andmete terviklikkuse välisvõtmete ja relatsioonide abil.

Andmebaasi keskseks olemiks on reis (*trip*), millega on seotud erinevad süsteemi funktsionaalsused, sealhulgas päevaplaanid, tegevused, kulud, kutsed ja teavitused. Selline ülesehitus võimaldab koondada kogu reisiga seotud informatsiooni ühte terviklikku andmemudelisse. Andmebaasi skeem on esitatud lisas 5.

Andmebaasina kasutati PostgreSQL süsteemi, mis valiti selle töökindluse, ACID-omaduste ning tugeva relatsioonilise andmemudeli toe tõttu. PostgreSQL võimaldab tagada andmete järjepidevuse ja terviklikkuse ka olukordades, kus samaaegselt tehakse mitmeid andmemuudatusi. See on oluline süsteemis, kus mitmed kasutajad võivad sama reisi andmeid paralleelselt muuta. [29]

Lisaks toetab PostgreSQL keerukamaid päringuid, relatsioone ja transaktsioone, mis sobivad hästi süsteemi kasutusjuhtudega, kus erinevad andmeüksused on omavahel tugevalt seotud. Relatsioonilise andmebaasi kasutamine võimaldab vähendada dubleeriva info tekkimist ning lihtsustab andmete struktureeritud haldamist ja pärimist. [29]

5.4.1 Linnade andmestiku import

Reiside ja sihtkohtade sisestamise toetamiseks kasutatakse süsteemis eelnevalt importditud linnade andmestikku. Linnade nimede allikana kasutatakse GeoNames *cities5000*

andmestikku, mis sisaldab üle 5000 elanikuga linnade andmeid [30].

Eeldefineeritud linnade kasutamine võimaldab pakkuda kasutajale struktureeritud ja ühtseid sihtkohtade nimetusi ning vähendab käsitsi sisestamisest tulenevate vigade riski. Lisaks võimaldab selline lähenemine lihtsustada sihtkohtade otsingut ja automaatset soovitamist kasutajaliideses.[31]

Linnade andmete import toimub tagarakenduse esmakäivitusel `CommandLineRunner` komponendi abil. Rakenduse käivitamisel kontrollitakse, kas linnade tabel sisaldab juba andmeid. Kui tabel on tühi, loetakse GeoNames andmefail ning imporditakse linnade kirjed andmebaasi.

Suuremahulise andmeimpordi tõhusamaks teostamiseks kasutatakse *JDBC batch insert* lähenemist, mis võimaldab andmeid sisestada partiidena ning vähendab üksikute andmebaasipäringute arvu. Selline lahendus parandab impordiprotsessi jõudlust ning vähendab süsteemi käivitamisel tekkivat koormust.

Valitud lähenemine tagab, et süsteemil on juba esmakordsel kasutamisel olemas sihtkohtade otsinguks vajalik baasandmestik, vältides samal ajal andmete korduvat importimist järgmistel käivitustel.

5.5 Tehisintellekti komponent

Käesolevas süsteemis kasutatakse tehisintellekti reisiplaneerimise protsessi toetamiseks, eesmärgiga vähendada kasutaja käsitsi tehtavat tööd ning pakkuda automaatseid soovitusi reisiplaani koostamisel. Suured keelemudelid (LLM-id) võimaldavad genereerida loomulikus keeles vastuseid ning töödelda kontekstipõhist sisendit [32]. Antud süsteemis kasutatakse tehisintellekti kasutaja sisendi põhjal personaliseeritud reisisoovituste genereerimiseks ning päevaplaani automaatseks täitmiseks.

5.5.1 Tehisintellekti teenuse eraldiseisev arhitektuur

Tehisintellekti funktsionaalsus on realiseeritud eraldiseisva teenusena, mis töötab sõltumatult põhirakendusest. Antud otsuse peamiseks põhjuseks oli süsteemi töökindluse ja hooldatavuse parandamine ning võimalike tõrgete isoleerimine. Kui tehisintellekti tee-

nus ei ole kättesaadav, jääb põhisüsteemi funktsionaalsus endiselt täielikult toimivaks. Seeläbi välditakse olukorda, kus ühe välise komponendi rike mõjutaks kogu rakenduse põhifunktsionaalsust. [33]

Arhitektuurselt on tehisintellekti teenuse ja Spring Boot rakendusel selge vastutuse jaotus. Spring Boot toimib keskse äriloogika kihina, mis võtab vastu kasutaja päringud, valideerib need ning pärib vajalikud andmed andmebaasist. Seejärel koondab Spring Boot vajaliku konteksti ning edastab selle koos kasutaja sisendiga eraldiseisvale tehisintellekti teenusele HTTP-põhise päringu kaudu.

Tehisintellekti teenus vastutab ainult keelemudeli töövoo eest, sealhulgas *prompt*'i koostamise, konteksti struktureerimise ning mudelipäringu haldamise eest. Selline lahendus võimaldab hoida AI-spetsiifilise loogika täielikult eraldatuna rakenduse põhikihtidest ning vähendab süsteemi sidusust.

Selle arhitektuuri tulemusena on süsteem paremini skaleeritav, kuna tehisintellekti teenust saab vajadusel iseseisvalt skaleerida vastavalt päringute mahule, ilma et oleks vaja muuta või ümber käivitada põhirakendust. Samuti lihtsustab eraldatus erinevate komponentide sõltumatut arendamist ja testimist, kuna tagarakendust ja tehisintellekti teenust saab arendada eraldi paralleelselt.

5.5.2 Keelemudeli valik ja analüüs

Süsteemi arendamisel hinnati erinevaid kommertslikke suurte keelemudelite lahendusi. Keelemudelite võrdlemisel lähtuti vastuste kvaliteedist, kuluefektiivsusest, ning struktureeritud väljundi usaldusväärsusest [34, 35].

Erinevate lahenduste analüüsi tulemusel valiti keelemudeliks GPT-5.4-mini [36]. Varasemad uuringud on näidanud, et väiksemad ja optimeeritud mudelid suudavad pakkuda hea kompromissi kvaliteedi ja jõudluse vahel, eriti rakendustes, kus oluline on madal latentsus ja suur päringute arv [37]. Antud süsteemi kontekstis osutus GPT-5.4-mini sobivaks, kuna see võimaldab kiiret vastuse genereerimist ning piisavat täpsust reisisoovituste loomiseks.

Oluline kriteerium oli ka struktureeritud väljundi stabiilsus, mida on käsitletud ka LLM-ide usaldusväärsuse ja determinismi probleemina teaduskirjanduses [38]. Süsteemis kasutatav

mudel osutus piisavalt järjepidevaks, et vähendada vajadust ulatusliku järeltöötluse järele.

5.5.3 LangChain

LangChain'i kasutatakse keelemudeliga suhtlemise tehnilise kihina [39]. Tegemist on raamistikuga, mis võimaldab luua rakendusi suurte keelemudelite baasil ning ühendada mudelid, *prompt*'id ja andmeallikad ühtseks töövooks. Antud süsteemis kasutatakse LangChain'i eelkõige selleks, et eraldada äriloogika ja mudelispetsiifiline implementatsioon.

5.5.4 Langfuse

Prompt'ide haldamiseks ja tehisintellekti päringute jälgimiseks kasutatakse Langfuse keskkonda [40]. Erinevalt traditsioonilisest lähenemisest ei ole *prompt*'id defineeritud rakenduse lähtekoodis, vaid neid hallatakse eraldi süsteemis, mis võimaldab nende muutmist ilma rakenduse uuesti juurutamiseta. Käesolevas töös kasutatud *prompt*'id on esitatud lisades 6 ja 7.

LLM-põhiste süsteemide puhul on jälgitavus kriitilise tähtsusega komponent süsteemi töökindluse ja kvaliteedi tagamisel [35]. Langfuse võimaldab salvestada päringute meta-andmeid ning analüüsida mudeli käitumist reaalsetes kasutusolukordades.

5.5.5 *Prompt*'ide disain ja optimeerimine

Prompt'ide loomine viidi läbi iteratiivse protsessina, mis põhines keelemudeli sisendite disaini ja optimeerimise põhimõtetel [41]. Esialgsed *prompt*'id koostati lähtuvalt soovitud väljundi struktuurist ning neid täiendati järk-järgult vastavalt mudeli käitumise analüüsile.

5.5.6 Vigade käsitlemine ja töökindlus

LLM-põhised süsteemid sõltuvad välisest teenusepakkujast, mistõttu ei ole nende töökindlus täielikult kontrollitav [38]. Süsteemis logitakse kõik vead ning kasutajale kuvatakse informatiivsed veateated. Lisaks kontrollitakse mudeli väljundi struktuuri enne selle kasutamist rakenduse järgmistes kihtides, et vältida vigaste vastuste levikut süsteemis.

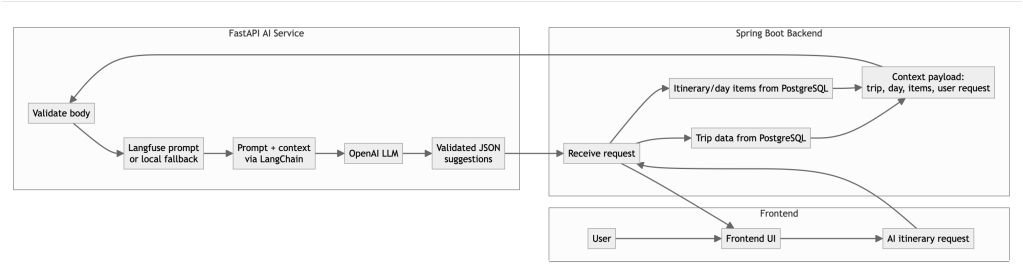
5.5.7 Tehisintellekti piirangud

Suured keelemudelid on tuntud oma piirangute poolest, sealhulgas hallutsinatsioonide genereerimine ja faktuaalne ebakindlus [42]. Reisiplaneerimise kontekstis võib see väljenduda ebatäpsetes soovitustes või ajaliselt ebarealistlikes plaanides.

Lisaks sõltub mudeli väljund tugevalt *prompt*'i sõnastusest ning sisendi detailsusest, mis on laialdaselt dokumenteeritud *prompt*'ide tundlikkuse probleemina [41]. Seetõttu tuleb süsteemi väljundit käsitleda kui soovituslikku, mitte faktitäpset planeerimisvahendit.

5.5.8 Töövoog

Joonisel 3 on esitatud tehisintellekti komponendi töövoog, mis kirjeldab kasutaja päringu töötlemist automaatse päevaplaani genereerimise näitel.



Joonis 3. Tehisintellekti komponendi töövoog päevaplaani automaatse genereerimise protsessis

Protsess algab kasutaja päringuga, mis jõuab Spring Boot tagarakendusse. Spring Boot valideerib päringu ning teostab vajalikud andmebaasipäringud, et koguda reisiplaneerimiseks vajalik kontekst (näiteks kasutaja eelistused ja olemasolevad andmed). Seejärel koondatakse saadud info ühtseks struktureeritud sisendiks.

Järgnevalt edastab Spring Boot selle konteksti koos kasutaja päringuga eraldiseisvale tehisintellekti teenusele HTTP-päringu kaudu. Tehisintellekti teenus ei suhtle otse andmebaasiga, vaid tugineb täielikult Spring Booti poolt ette valmistatud sisendile.

Seejärel kombineeritakse Langfuse'ist saadud *prompt* kasutaja sisendiga ning päring edastatakse LangChain'i kaudu GPT-5.4-mini mudelile. Lõppvastus saadetakse tagasi Spring Booti rakendusele, mis omakorda edastab selle kasutajale.

Selline arhitektuur toetab modulaarset disaini ning on kooskõlas kaasaegsete LLM-põhiste

süsteemide arhitektuursete praktikatega, kus rõhutatakse komponentide eraldatust ning jälgitavust [35].

6 Loodud lahenduse analüüs

Analüüsi eesmärk oli hinnata arendatud süsteemi vastavust püstitatud eesmärkidele ning uurida lahenduse kasutatavust ja praktilist väärtust erinevates kasutusstsenaariumides. Selleks viidi läbi valideerimine nii tellija kui ka tavakasutajatega, et koguda tagasisidet süsteemi funktsionaalsuse, kasutajakogemuse ja tööprotsesside toetamise kohta.

Analüüsi käigus keskenduti eelkõige sellele, kuidas arendatud lahendus vastab algselt defineeritud nõuetele, millised olid süsteemi peamised tugevused ja puudused ning milliseid edasiarendussuundi oleks võimalik tulevikus rakendada. Lisaks hinnati, kas süsteem on piisavalt üldistatav ja skaleeritav, et seda oleks võimalik kasutada ka väljaspool konkreetse tellija tööprotsessi.

6.1 Valideerimine tellijaga

Tellijaga läbi viidud valideerimisprotsess koosnes mitmest järjestikusest etapist, mille eesmärk oli tagada arendatava süsteemi vastavus tellija vajadustele ning võimaldada kasutajapõhise tagasiside põhjal süsteemi järjepidevat täiustamist. Valideerimine viidi läbi iteratiivse protsessina kogu arendustsükli vältel, et tagada süsteemi vastavus funktsionaalsetele ja kasutatavusnõuetele, hõlmates nii nõuete täpsustamist, vahepealseid arutelusid kui ka süsteemi funktsionaalsuse testimist erinevates arendusetappides. Valideerimisprotsessi võib jagada neljaks peamiseks etapiks: nõuete valideerimine, iteratiivne arendusprotsess, minimaalse toimiva toote testimine, lõpplahenduse valideerimine.

6.1.1 Nõuete valideerimine

Esimeses etapis toimus koostöös tellijaga süsteemi nõuete kaardistamine ja täpsustamine, mis on kirjeldatud käesoleva töö varasemates peatükkides. Selle etapi käigus selgitati välja tellija peamised vajadused, probleemkohad ning ootused loodavale süsteemile. Nõuete määratlemine oli oluline alus edasisele arendustööle, kuna see võimaldas luua selge arusaama süsteemi eesmärgist ja funktsionaalsusest. Täielik süsteemile esitatud nõuete

loetelu ning nende realiseerimise tulemused on toodud lisas 8.

6.1.2 Iteratiivne arendusprotsess

Teises etapis toimus arendusprotsessi käigus regulaarne suhtlus tellijaga, sealhulgas vahekoosolekute pidamine ning tekkinud küsimuste ja probleemide arutamine. Selline iteratiivne lähenemine võimaldas saada jooksvalt tagasisidet arendatava süsteemi kohta ning teha vajadusel muudatusi enne järgmiste arendusosade valmimist. Regulaarne suhtlus aitas vähendada riski, et lõplik lahendus ei vasta tellija ootustele, ning toetas paindlikku arendusprotsessi.

6.1.3 Minimaalse toimiva toote testimine

Esimese testimisetapi käigus tutvustati tellijale arendatud rakenduse esialgset versiooni, mille eesmärk oli hinnata süsteemi kasutusmugavust, funktsionaalsust ning vastavust eelnevalt kaardistatud nõuetele. Valideerimine viidi läbi koostöös tellija esindajaga, kes sai rakendust praktiliselt kasutada ning anda otsest tagasisidet selle toimivuse ja kasutuskogemuse kohta.

Selles etapis sai loodud rakendus valdavalt positiivset tagasisidet. Tellija tõi peamiste tugevustena esile süsteemi kasutusmugavuse ning intuitiivse kasutajaliidese, mis võimaldas erinevaid funktsioone kiiresti omandada ka ilma põhjaliku juhendamiseta. Samuti kinnitati, et rakenduses realiseeritud põhifunktsionaalsused vastasid tellija varasemalt määratletud nõuetele ning võimaldasid täita peamisi tööülesandeid, nagu reiside haldamine, osalejate info koondamine ja kulude jälgimine.

Positiivse üllatusena toodi esile ka tehisintellektil põhinevad funktsionaalsused, mis pakkusid kasutajatele täiendavat tuge reiside planeerimisel. Kuigi nimetatud funktsionaalsused ei ole tellija igapäevases tööprotsessis otseselt hädavajalikud, leiti, et need võivad teatud olukordades aidata tööprotsessi lihtsustada ning vähendada käsitsi tehtava töö hulka. Seetõttu hinnati nende olemasolu süsteemis väärtuslikuks lisavõimaluseks, mis suurendab rakenduse paindlikkust ja kasutusvõimalusi tulevikus.

Testimise käigus tuvastati ka mitmeid väiksemaid parandamist vajavaid kohti, mis ei mõjutanud süsteemi põhifunktsionaalsust, kuid mille täiendamine aitaks parandada kasutuskogemust. Näiteks märgiti vajadust korrigeerida mõningaid tekstilisi elemente ning

muuta teatud kasutajaliidese komponentide paigutust loogilisemaks.

Lisaks ilmnes valideerimise käigus mitmeid ettepanekuid uute funktsionaalsuste lisamiseks, mis aitaksid tellija tööprotsessi veelgi efektiivsemaks muuta. Üheks oluliseks sooviks oli võimalus saata kasutajatele teavitusi rakenduse siseselt olukordades, kus reisiga seotud andmed muutuvad, näiteks kui ajakava või olulist infot uuendatakse. Samuti peeti kasulikuks võimalust saata teavitusi e-posti teel, et tagada oluline info jõudmine kasutajateni ka siis, kui nad ei kasuta rakendust aktiivselt.

Kõige olulisema muudatusena toodi välja vajadus täiustada aruandluse funktsionaalsust. Tellija hinnangul peaks aruannete lõpus olema kuvatud ka kuludega seotud üles laaditud failid, näiteks arved või kviitungid. Selline lahendus võimaldaks koondada kogu vajalik dokumentatsioon ühte aruandesse ning vähendaks vajadust hoida faile eraldi keskkondades või saata neid täiendavalt e-posti teel. See muudaks aruandluse protsessi terviklikumaks ning lihtsustaks dokumentide haldamist.

Samuti arutati võimalust lisada täiendavaid autentimisviise, näiteks sisselogimine kolmanda osapoole teenuse kaudu. Näiteks peeti kasulikuks võimalust kasutada sisselogimiseks olemasolevat kontot Google'i teenuses, mis lihtsustaks kasutajate ligipääsu süsteemile ning vähendaks vajadust uute paroolide loomise ja meeldejätmise järele.

Esimese valideerimisetapi tulemused kinnitasid arendatud süsteemi põhifunktsionaalsuse sobivust tellija vajadustele. Samal ajal andis see etapp väärtuslikku tagasisidet süsteemi edasiseks arendamiseks ning aitas tuvastada täiendavaid funktsionaalsusi, mis võivad parandada kasutusmugavust ja tööprotsessi efektiivsust.

6.1.4 Lõpptoote valideerimine

Lõpptoote valideerimine viidi läbi pärast süsteemi kõigi planeeritud põhifunktsionaalsuste realiseerimist. Selle etapi eesmärk oli hinnata rakenduse vastavust tellija algsetele nõuetele, süsteemi kasutusmugavust ning üldist sobivust igapäevasesse tööprotsessi.

Lõpplahenduse valideerimine viidi läbi koostöös tellija esindajaga reaalse tööprotsessi kontekstis. Erinevalt varasemast MVP valideerimisest toimus testimine asünkroonselt pikema perioodi vältel, mille jooksul kasutati rakendust aktiivselt tegeliku projektireisi

planeerimisel ja haldamisel. Selline lähenemine võimaldas hinnata süsteemi toimivust realistlikes kasutusolukordades ning tuvastada võimalikke kitsaskohti, mis lühiajalise testimise käigus ei pruugi ilmneda. Valideerimise käigus keskenduti süsteemi terviklikkusele, sealhulgas reiside haldamise, osalejate lisamise, kulude jälgimise ja aruannete genereerimise funktsionaalsustele. Lisaks testis tellija rakendust erinevate kasutajarollide vaates, hinnates süsteemi kasutatavust nii reisikorraldajate kui ka reisil osalejate perspektiivist. Lisaks võimaldati ka tegelikel osalejatel süsteemiga liituda ja rakendust kasutada, mis andis täiendavat tagasisidet kasutajaliidese intuitiivsuse, töövoogude loogilisuse ning funktsioonide omavahelise sidususe kohta.

Valideerimise tulemusena kinnitati, et arendatud süsteem vastab tellija poolt seatud põhieesmärkidele ning võimaldab oluliselt lihtsustada varasemat tööprotsessi. Eriti toodi esile asjaolu, et kogu reisiga seotud informatsioon on koondatud ühte keskkonda, mis vähendab vajadust kasutada mitut erinevat tööriista ning parandab andmete kättesaadavust ja ülevaatlikkust.

Samuti hinnati positiivselt süsteemi rollipõhist kasutajahaldust, mis võimaldab määrata kasutajatele erinevaid õiguseid ja vastutusalasid vastavalt nende rollile projektireisi kontekstis. Tellija hinnangul vastas realiseeritud lahendus hästi nende olemasolevatele tööprotsessidele ning võimaldas süsteemi kasutada kõikides peamistes kasutusstsenaariumides ilma täiendavate kohandusteta. Valideerimise käigus arutati ka võimalust laiendada süsteemi tulevikus dünaamilisema rollihaldusega, kus administraatoritel oleks võimalik rolle ja õiguseid ise defineerida. Kuigi sellist lähenemist peeti funktsionaalsuse seisukohalt paindlikuks, toodi samas välja, et liiga keerukas rollide seadistamine võib muuta kasutajaliidese vähem intuitiivseks, eriti kasutajate jaoks, kellel puudub tehniline taust. Seetõttu leiti, et olemasolev eeldefineeritud rollidel põhinev lahendus pakub praeguses tööprotsessis paremat tasakaalu paindlikkuse ja kasutusmugavuse vahel.

Aruandluse funktsionaalsus vastas tellija ootustele ning võimaldas genereerida struktureeritud ülevaateid reisidega seotud kuludest ja tegevustest. Eelnevas valideerimisetapis tehtud ettepanekud, nagu kuludokumentide kaasamine aruannetesse, realiseeriti ning parandasid oluliselt aruannete funktsiooni kasutatavust.

Tehisintellektil põhinevaid funktsionaalsusi hinnati jätkuvalt positiivselt kui toetavaid

tööriistu, mis aitavad vähendada käsitsi tehtava töö hulka ning toetavad reiside planeerimise protsessi. Eriti toodi esile funktsioon *Autofill Day*, mida peeti kasulikuks olukordades, kus on vaja projekti raames kiiresti koostada ühepäevane reisiplaan. Tellija hinnangul võimaldas see funktsionaalsus genereerida kiiresti esmase päevaplaani ning pakkuda välja tegevusi ja sihtkohti, mis andsid kasutajale planeerimiseks selge lähtepunkti. Selle tulemusena vähenes vajadus teha mahukat eeltööd ja käsitsi informatsiooni otsida, mis muutis reiside planeerimise ajaliselt efektiivsemaks.

Tellija tõi eriti positiivselt esile asjaolu, et esimese testimise käigus välja pakutud täiendavad funktsionaalsused realiseeriti edukalt süsteemi lõppversioonis. Nende hulka kuulusid näiteks e-posti teel saadetavad teavitused, Google'i kontoga autentimine ning aruannete genereerimine koos seotud failide ja kuludokumentidega. Tellija hinnangul parandasid need täiendused märkimisväärselt süsteemi praktilist väärtust ja kasutusmugavust ning muutsid tööprotsessi terviklikumaks.

Lõpptoote valideerimise käigus ei tuvastatud kriitilisi vigu, mis takistaksid süsteemi kasutuselevõttu. Esile toodi vaid üksikuid väiksemaid parenduskohti, mis olid seotud peamiselt kasutajaliidese detailide ja kasutusmugavusega. Need tähelepanekud ei mõjutanud süsteemi põhifunktsionaalsust ega selle vastavust tellija vajadustele.

Lisaks arutati valideerimise käigus võimalikke edasiarendussuundi ning tellija pakkus välja mitmeid täiendavaid funktsionaalsusi, mis võiksid tulevikus süsteemi kasutusvõimalusi laiendada. Näiteks arutati võimalust täiustada otsingufunktsionaalsust, et muuta süsteemi kasutamine mugavamaks. Tellija tõi välja, et igapäevatöös võib aeg-ajalt ette tulla olukordi, kus otsitavat infot ei leita kohe üles, näiteks juhul, kui sisestamisel tehakse väike kirjaviga või kasutatakse veidi erinevat sõnastust. Kuigi tegemist ei olnud kriitilise probleemiga, peeti sellise olukorra lahendamist kasutusmugavuse seisukohalt mõistlikuks täienduseks. Samas rõhutati, et olemasolev lahendus on juba praeguses seisus piisavalt terviklik ja sobiv igapäevaseks kasutamiseks.

Kokkuvõttes võib järeldada, et arendatud lahendus vastab tellija poolt seatud eesmärkidele ning toetab efektiivselt projektireiside planeerimise ja haldamise tööprotsessi.

6.2 Valideerimine tavakasutajatega

Tavakasutajatega läbiviidud valideerimise eesmärk oli hinnata süsteemi kasutatavust, intuitiivsust ja üldist sobivust kasutajatele, kes ei ole seotud tellija spetsiifiliste tööprotsessidega. Lisaks oli valideerimise üheks eesmärgiks hinnata, kas arendatud lahendus on piisavalt üldistatud ja skaleeritav, et seda oleks võimalik kasutada ka teistes organisatsioonides ning individuaalsete kasutajate poolt. See oli oluline, et süsteem ei lahendaks üksnes ühe konkreetse organisatsiooni vajadusi ega kujuneks liiga kitsalt tellimuspõhiseks lahenduseks.

Valideerimine viidi läbi pärast süsteemi põhifunktsionaalsuste realiseerimist ning keskendus eelkõige kasutajakogemuse ja kasutajaliidese arusaadavuse hindamisele esmakordsel kasutamisel.

6.2.1 Valideerimise metoodika

Süsteemi valideerimiseks kasutati kasutatavustestimise (*usability testing*) põhimõtteid, mille eesmärk on hinnata, kui efektiivselt, arusaadavalt ja mugavalt suudavad kasutajad süsteemi funktsionaalsusi kasutada [43]. Valideerimine viidi läbi stsenaariumipõhise testimisena, kus kasutajatele anti ette konkreetsed ülesanded, mida tuli süsteemis iseseisvalt täita.

Selline lähenemine võimaldas hinnata süsteemi intuitiivsust, kasutajaliidese loogilisust ning kasutajate võimet mõista süsteemi töövoogu ilma eelneva juhendamiseta. Lisaks ülesannete täitmise jälgimisele koguti kasutajatelt kvalitatiivset tagasisidet süsteemi tugevuste, puuduste ja võimalike parenduskohtade kohta.

Kasutajakogemuse kvantitatiivseks hindamiseks kasutati viiepallisel Likerti skaalal põhinevat küsimustikku, kus kasutajad hindasid erinevaid süsteemi kasutatavusega seotud väiteid. Likerti skaala on laialdaselt kasutatav meetod hoiakute ja kasutajakogemuse hindamiseks infosüsteemide uurimisel [44].

Valideerimise käigus täitsid kasutajad erinevaid süsteemi põhifunktsionaalsusi hõlmavaid testülesandeid. Täielik ülesannete loetelu on esitatud lisas 2.

6.2.2 Valideerimise tulemused

Valideerimises osales 10 tavakasutajat vanuses 18–35 aastat, kelle varasem kogemus reiside planeerimisega varieerus. Osalejatel paluti täita eeldefineeritud ülesandeid ilma täiendava juhendamiseta, mis võimaldas hinnata süsteemi intuitiivsust ning kasutajate võimet mõista rakenduse tööloogikat iseseisvalt.

Valideerimise käigus täitsid kasutajad erinevaid süsteemi põhifunktsionaalsusi hõlmavaid testülesandeid, sealhulgas reiside loomist, päevaplaani haldamist, kulude sisestamist, failide lisamist ning osalejate kutsumist. Kõik osalejad suutsid ülesanded edukalt lõpule viia ilma kõrvalise abita ning ühegi ülesande täitmisel ei tekkinud olukorda, kus kasutaja oleks töövoos täielikult takerdunud. See viitab sellele, et süsteemi põhifunktsionaalsused ja peamised kasutusvood olid kasutajatele arusaadavad ka esmakordsel kasutamisel.

Pärast teststsenaariumite läbimist koguti kasutajatelt täiendavat kvalitatiivset tagasisidet süsteemi kõige kasulikumate funktsionaalsuste ning võimalike puudujääkide kohta. Lisaks paluti kasutajatel hinnata erinevaid kasutajakogemusega seotud väiteid viiepallisel skaalal, kus 1 tähistas madalat ja 5 täielikku nõustumist väitega. Kasutajatele esitatud küsimustik on lisas 3 ja vastused lisas 4. Täielikud küsimustiku tulemused ja vastuste jaotused on esitatud graafilisel kujul lisas 4.

Küsimustiku tulemused olid üldiselt väga positiivsed. Rakenduse kasutamist ilma juhendita hinnati keskmiselt hindega 4,9 ning ülesannete täitmise lihtsust hindega 4,8. Samuti hindasid kasutajad kõrgelt väidet, et nad ei jäänud ühegi ülesande täitmisel kinni, mille keskmine hinnang oli 4,9. Need tulemused viitavad sellele, et süsteemi põhifunktsionaalsused ja peamised kasutusvood olid kasutajatele arusaadavad ka esmakordsel kasutamisel.

Positiivselt hinnati ka süsteemi ülesehitust ja navigeerimist. Navigeerimise loogilisust hinnati keskmiselt hindega 4,7 ning rakenduse mõju reiside planeerimise lihtsustamisele hindega 4,8. Tulemused näitavad, et kasutajad pidasid süsteemi struktuuri loogiliseks ning tunnetasid rakenduse praktilist väärtust reiside planeerimise protsessis.

Kasutajate hinnangul osutus eriti väärtuslikuks asjaolu, et kogu reisiga seotud informatsioon, sealhulgas ajakava, kulud, tegevused ja osalejate muudatused, on koondatud ühte keskkonda. Selline lähenemine lihtsustas kasutajate hinnangul oluliselt reiside planeerimist ja koostööd,

kuna kõik osalejad said jooksvalt informatsiooni täiendada ja muudatusi reaalajas jälgida.

Olulise tugevusena toodi esile ka kulude jagamise funktsionaalsus, mille kasulikkuse keskmine hinnang oli 4,8. Kasutajate hinnangul vähendas automaatne kulude tasaarvelduste arvutamine märkimisväärselt käsitsi tehtava arvutamise vajadust ning muutis ühiste reisikulude haldamise mugavamaks ja ülevaatlikumaks.

Tehisintellektil põhinevaid funktsionaalsusi hinnati samuti positiivselt. AI-põhiste soovituste kasulikkuse keskmine hinnang oli 4,6 ning AI mõju reisiplaani kiiremale koostamisele hinnati keskmiselt hindega 4,7. Kasutajate hinnangul aitasid AI-funktsionaalsused vähendada käsitsi tehtavat planeerimistööd ning pakkusid head lähtepunkti reisiplaani koostamiseks. Tulemused viitavad sellele, et tehisintellektil põhinevad funktsionaalsused suutsid pakkuda kasutajatele praktilist lisaväärtust ka olukordades, kus kasutajatel puudus varasem detailne reisiplaneerimise kogemus.

Eriti väärtustati võimalust saada automaatselt genereeritud reisisoovitusi ja päevaplaane, mis vähendasid vajadust iseseisvalt sihtkoha kohta informatsiooni otsida. Kasutajad tõid välja, et AI-põhised soovitused andsid planeerimisprotsessile selge lähtepunkti ning võimaldasid koostada tervikliku reisiplaani oluliselt väiksema ajakuluga.

Positiivselt hinnati ka süsteemi ajapõhist päevaplaani organiseerimist, kus tegevused järjestati automaatselt vastavalt määratud kellaaegadele. Kasutajate sõnul parandas see reisiplaani loetavust ning võimaldas saada kiire ülevaate päeva struktuurist ilma täiendava käsitsi korraldamiseta.

Samuti peeti mugavaks võimalust saata reisikutseid e-posti teel, kuna see lihtsustas uute osalejate kaasamist ning vähendas vajadust kasutada eraldi suhtluskanaleid.

Failide lisamise funktsionaalsuse keskmine hinnang oli 4,2, mis oli võrreldes teiste funktsionaalsustega mõnevõrra madalam, kuid jäi siiski positiivsele tasemele. Osa kasutajaid tõi siiski esile failide lisamise võimaluse nii tegevuste kui ka kulude juurde kui olulise praktilise tugevuse. Kasutajate hinnangul muutis see süsteemi märkimisväärselt praktilisemaks, kuna kogu reisiga seotud dokumentatsioon oli kättesaadav ühest keskkonnast kõigile osalejatele.

Eriti kasulikuks peeti võimalust lisada kulude juurde tšekke ja arveid ning tegevuste juurde pileteid, ajakavasid ja muud reisiga seotud informatsiooni. Failide lisamise funktsionaalsuse

mõnevõrra madalam hinnang võib olla seotud asjaoluga, et see on eelkõige suunatud organisatsiooniliste projektireiside kasutusjuhtudele, kus on vajalik säilitada kuludokumente, pileteid ja muud aruandlusega seotud materjali. Individuaalsete reiside kontekstis ei pruugi selline funktsionaalsus olla sama kriitilise tähtsusega, kuid kasutajad hindasid seda siiski praktiliseks ja kasulikuks võimaluseks.

Valideerimise käigus ilmnes siiski ka mõningaid kasutajaliidesega seotud ebaselgusi. Kõige sagedamini tekitas segadust mõistete „projekt“ ja „reis“ eristamine, kuna tavakasutajatele ei olnud nende omavaheline seos esmakordsel kasutamisel täielikult intuitiivne. See viitab vajadusele täiustada süsteemi terminoloogiat ja visuaalset hierarhiat, et muuta erinevate üksuste roll kasutajale selgemaks.

Lisaks ei saanud osa kasutajatest esialgu täielikult aru *Important Info* sektsiooni eesmärgist ega selle erinevusest tavapärasest reisikirjeldusest või päevakavast. Tagasiside põhjal selgus, et funktsionaalsuse tähendus ei olnud kasutajaliideses piisavalt selgelt kommunikeeritud, mistõttu jäi osa kasutajatele arusaamatuks, millist tüüpi informatsiooni sinna lisada tuleks.

Testimise tulemuste põhjal otsustati piirata selle funktsionaalsuse kasutamist ainult projektireisidega, kuna tavareiside kontekstis ei osutunud selline eraldiseisev infohalduse sektsioon vajalikuks. Selline muudatus võimaldas lihtsustada tavareiside kasutajakogemust.

Kokkuvõttes kinnitas tavakasutajatega läbiviidud valideerimine, et süsteem on kasutatav ka ilma eelneva juhendamiseta ning sobib lisaks organisatsioonilisele kasutusele ka individuaalseks ja grupipõhiseks reisiplaneerimiseks. Testimise tulemused näitasid, et süsteemi põhifunktsionaalsused on intuitiivsed, toetavad efektiivset koostööd ning pakuvad kasutajatele praktilist lisaväärtust reiside planeerimisel ja haldamisel.

6.3 Edasiarenduse plaanid

Käesoleva bakalaureusetöö raames arendatud süsteem täidab püstitatud eesmärgid ning vastab tellija peamistele vajadustele. Samas ilmnes nii arendusprotsessi kui ka valideerimise käigus mitmeid võimalusi süsteemi edasiseks täiustamiseks ja laiendamiseks.

Ühe võimaliku edasiarendussuuna moodustab failihalduse täiustamine. Tulevikus võiks süsteem toetada failide paremat struktureerimist, kategoriseerimist ja otsingut, mis muudaks

suuremahulise dokumentatsiooni haldamise efektiivsemaks. Näiteks võiks kasutajatel olla võimalik näha kõiki faile ühes kohas ja filtreerida faile tüübi, üleslaadimise aja või seotud tegevuse põhjal.

Valideerimise käigus tõi tellija välja ka võimaluse täiustada otsingufunktsionaalsust. Praegust lahendust oleks võimalik laiendada hägusotsingu algoritmidega, mis võimaldaksid leida tulemusi ka kirjavigade või erinevate sõnavormide korral. See parandaks süsteemi kasutatavust olukordades, kus kasutaja ei mäleta täpset nimetust või teeb sisestamisel väikese vea. Lisaks oleks võimalik laiendada otsingut failidele ja kasutajatele, võimaldades kiiremat ligipääsu süsteemis olevale informatsioonile. Samas kaasnevad selliste funktsionaalsustega ka täiendavad turvariskid, näiteks kasutajate enumereerimise võimalus, mistõttu tuleks otsingumehhanismide edasiarendamisel pöörata suuremat tähelepanu ligipääsukontrollile ja päringute piirangutele.

Tavakasutajatega läbiviidud valideerimise käigus ilmnes, et mõiste „projekt“ ei olnud kõikidele kasutajatele täielikult arusaadav ning selle seos reisidega tekitas esmakordsel kasutamisel segadust. Selle põhjal võiks tulevikus süsteemi kasutusvooge paremini eraldada erinevate kasutusstsenaariumide vahel, näiteks organisatsioonilise projektireiside halduse ja tavapärase individuaalse reisiplaneerimise jaoks. Alternatiivse lahendusena võiks süsteem sisaldada onboarding- või juhendamisfunktsionaalsust, mis tutvustab kasutajale esmakordsel kasutamisel süsteemi põhilisi töövooge ja mõisteid.

Täiendavaid edasiarendusvõimalusi pakuvad ka süsteemi tehisintellektil põhinevad funktsionaalsused. Tulevikus võiks AI-komponente laiendada kontekstiteadlikumate soovitussüsteemidega, mis arvestavad kasutaja varasemaid reisiplaane, eelistusi ja kulumustreid. Samuti võiks süsteem toetada automaatset reisiplaani optimeerimist, kus tegevuste järjekorda kohandatakse vastavalt asukohale, ajakulule või transpordivõimalustele.

Lisaks oleks võimalik integreerida süsteem väliste reisiplaneerimise teenustega. Näiteks võiks rakendus kasutada kolmandate osapoolte API-sid lendude, hotellide või transpordivõimaluste otsimiseks ja kuvamiseks otse rakenduse kasutajaliideses. Ühe võimaliku lahendusena võiks AI-funktsionaalsusi siduda reisiplatvormidega nagu Skyscanner, et pakkuda kasutajatele automaatselt sobivaid lennu- ja transpordisoovitusi vastavalt planeeritava reisi parameetritele.

Arhitektuurilisest vaatenurgast oleks süsteemi võimalik edasi arendada suurema skaleeritavuse ja modulaarsuse suunas. Näiteks võiks tulevikus kaaluda mikroteenustel põhinevat arhitektuuri, mis võimaldaks erinevaid süsteemi komponente sõltumatumalt hallata ja edasi arendada.

Arendatud süsteem pakub tugevat alust edasiseks arenduseks ning võimaldab tulevikus lisada uusi funktsionaalsusi vastavalt kasutajate vajadustele ja tehnoloogia arengule.

7 Kokkuvõte

Bakalaureusetöö käigus valmis veebipõhine reisihaldusrakendus, mille eesmärk on võimaldada kasutajatel planeerida ja hallata reise ühtses keskkonnas. Töö keskendus eelkõige organisatsiooniliste projektireiside haldamise kasutusjuhtudele ning lähtus tellija tööprotsessidest ja vajadustest. Lahendus toetab reiside loomist, päevapõhiste ajakavade koostamist, kulude haldamist, osalejate lisamist ning aruannete genereerimist. Lisaks sisaldab süsteem tehisintellektil põhinevaid funktsioone reisiplaanide genereerimiseks ja soovituste andmiseks.

Töö käigus analüüsiti olemasolevaid lahendusi ning kaardistati koostöös tellijaga süsteemi nõuded. Nende põhjal arendati terviklik rakendus, kasutades Reacti ja TypeScripti klientrakenduse jaoks, Java Spring Booti tagarakenduse jaoks ning PostgreSQL andmebaasi. Süsteem järgib mitmekihilist arhitektuuri ning tehisintellekti funktsionaalsus on realiseeritud eraldiseisva teenusena.

Valminud süsteemi valideeriti nii tellija kui ka tavakasutajatega. Tulemused näitasid, et rakendus vastab tellija vajadustele ning on kasutajatele arusaadav ja kasutatav ka ilma eelneva juhendamiseta. Eriti positiivselt hinnati võimalust koondada reisidega seotud info, kulud, dokumendid ja ajakavad ühte keskkonda.

Kokkuvõttes täidab arendatud lahendus oma eesmärgi, pakkudes ühtset ja terviklikku keskkonda reiside planeerimiseks ning lihtsustades varasemat killustatud tööprotsessi. Edaspidi on võimalik süsteemi edasi arendada näiteks täiustatud otsingufunktsionaalsuse, väliste reisiteenuste integratsioonide ning täpsemate keelemudelite põhiste soovitussüsteemide suunas.

Kasutatud kirjandus

[1] UN Tourism. *UNWTO 2022: A Year in Review*. [Kasutatud 03.02.2026]. 2025. URL: https://www.untourism.int/unwto-2022-a-year-review.

[2] European Commission. *Erasmus+ Programme Guide*. [Kasutatud 03.02.2026]. 2024. URL: https://erasmus-plus.ec.europa.eu/programme-guide.

[3] Mary Czerwinski, Eric Horvitz ja Susan Wilhite. „A Diary Study of Task Switching and Interruptions“. Teoses: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [Kasutatud 03.05.2026]. 2004, lk. 175–182. URL: https://erichorvitz.com/taskdiary.pdf.

[4] Haridus- ja Noorteamet. *Mis on Erasmus+?* [Kasutatud 03.02.2026]. 2025. URL: https://eeagentuur.ee/programmist/mis-on-erasmus/.

[5] Alan R. Hevner *et al.* „Design Science in Information Systems Research“. *MIS Quarterly* 28.1 (2004). [Kasutatud 17.05.2026], lk. 75–105.

[6] Ken Peffers *et al.* „A Design Science Research Methodology for Information Systems Research“. *Journal of Management Information Systems* 24.3 (2007). [Kasutatud 17.05.2026], lk. 45–77.

[7] Wanderlog. [Kasutatud 05.02.2026]. URL: https://wanderlog.com/.

[8] Stippl. [Kasutatud 05.02.2026]. URL: https://www.stippl.io/.

[9] Trip Planner AI. [Kasutatud 05.02.2026]. URL: https://tripplanner.ai/.

[10] Perk. https://www.perk.com/. [Kasutatud 13.05.2026]. 2025.

[11] Perk Support. *Assign roles and permissions*. https://support.perk.com/hc/articles/21925999102492-Assign-roles-and-permissions. [Kasutatud 13.05.2026]. 2025.

[12] Booking.com. *Booking.com for Business*. https://business.booking.com/. [Kasutatud 13.05.2026]. 2025.

[13] Booking.com. *Travel Managers - Booking.com for Business*. https://business.booking.com/travel-managers/. [Kasutatud 13.05.2026]. 2025.

[14] Tripadvisor. [Kasutatud 05.02.2026]. URL: https://www.tripadvisor.com/.

[15] Skyscanner. [Kasutatud 05.02.2026]. URL: https://www.skyscanner.net/.

[16] Martin Fowler. *Presentation Domain Data Layering*. [Kasutatud 17.05.2026]. 2005. URL: https://martinfowler.com/bliki/PresentationDomainDataLayering.html.

[17] Roy Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. [Kasutatud 17.05.2026]. 2000. URL: https://ics.uci.edu/~fielding/pubs/dissertation/top.htm.

[18] Martin Fowler ja James Lewis. *Microservices*. [Kasutatud 17.05.2026]. 2014. URL: https://martinfowler.com/articles/microservices.html.

[19] Meta Platforms, Inc. *React Documentation*. [Kasutatud 15.05.2026]. 2025. URL: https://react.dev/.

[20] Spring Framework Team. *Spring Boot Reference Documentation*. [Kasutatud 15.05.2026]. 2025. URL: https://docs.spring.io/spring-boot/index.html.

[21] Spring Security Team. *Spring Security Reference Documentation*. [Kasutatud 15.05.2026]. 2025. URL: https://docs.spring.io/spring-security/reference/.

[22] Internet Engineering Task Force (IETF). *RFC 7519: JSON Web Token (JWT)*. [Kasutatud 15.05.2026]. 2015. URL: https://www.rfc-editor.org/rfc/rfc7519.

[23] Len Bass, Paul Clements ja Rick Kazman. „Software Architecture in Practice“. *Addison-Wesley* (2021). [Kasutatud 15.05.2026].

[24] Ravi Sandhu *et al.* „Role-Based Access Control Models“. *IEEE Computer* 29.2 (1996). [Kasutatud 10.05.2026], lk. 38–47. URL: https://csrc.nist.gov/csrc/media/projects/role-based-access-control/documents/sandhu96.pdf.

[25] David Ferraiolo, D. Richard Kuhn ja Ramaswamy Chandramouli. *Role-Based Access Control*. 2. väljaanne. [Kasutatud 10.05.2026]. Artech House, 2007. URL: https://csrc.nist.gov/pubs/book/2007/01/rolebased-access-control/final.

[26] D. Hardt. *The OAuth 2.0 Authorization Framework*. [Kasutatud 17.05.2026]. 2012. URL: https://datatracker.ietf.org/doc/html/rfc6749.

[27] Adobe. *PDF Reference and Adobe PDF Resources*. [Kasutatud 17.05.2026]. 2026. URL: https://opensource.adobe.com/dc-acrobat-sdk-docs/.

[28] Mozilla. *PDF.js*. [Kasutatud 17.05.2026]. 2026. URL: https://mozilla.github.io/pdf.js/.

[29] PostgreSQL Global Development Group. *PostgreSQL Documentation*. [Kasutatud 17.05.2026]. 2026. URL: https://www.postgresql.org/docs/.

[30] GeoNames. *GeoNames Geographical Database*. [Kasutatud 17.05.2026]. 2026. URL: https://www.geonames.org/.

[31] IBM. *Reference data*. [Kasutatud 17.05.2026]. 2026. URL: https://www.ibm.com/docs/en/cloud-paks/cp-data/5.3.x?topic=artifacts-reference-data.

[32] Amazon Web Services. *What is a Large Language Model (LLM)?* [Kasutatud 23.04.2026]. 2026. URL: https://aws.amazon.com/what-is/large-language-model/.

[33] Chris Richardson. *Microservice Architecture Pattern*. [Kasutatud 01.06.2026]. URL: https://microservices.io/patterns/microservices.html.

[34] Zichao Lin *et al.* „Towards Trustworthy LLMs: A Review on Debiasing and Dehallucinating in Large Language Models“. *Artificial Intelligence Review* (2024). [Kasutatud 15.05.2026]. URL: https://link.springer.com/article/10.1007/s10462-024-10896-y.

[35] Yang Liu, Yuanshun Yao ja Jean-Francois Ton. „Trustworthy LLMs: A Survey and Guideline for Evaluating Large Language Models' Alignment“. *arXiv* (2023). [Kasutatud 15.05.2026]. URL: https://arxiv.org/abs/2308.05374.

[36] OpenAI. *GPT-5.4-mini model specification and documentation*. [Kasutatud 17.05.2026]. 2026.

[37] Jiho Shin *et al.* „Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models“. *arXiv* (2023). [Kasutatud 15.05.2026]. URL: https://arxiv.org/abs/2310.10508.

[38] Jiayi Ma Salvador García-Lopez. „Survey of Uncertainty Estimation in LLMs: Sources, Methods, Applications, and Challenges“. *Information Fusion* (2026). [Kasutatud 15.05.2026]. URL: https://www.sciencedirect.com/science/article/pii/S1566253525011194.

[39] LangChain. *LangChain Overview*. [Kasutatud 23.04.2026]. 2026. URL: https://docs.langchain.com/oss/python/langchain/overview.

[40] Langfuse. *Langfuse Documentation*. [Kasutatud 23.04.2026]. 2026. URL: https://langfuse.com/docs.

[41] Vu Tran, Le-Minh Nguyen ja Dang Anh-Hoang. „Survey and Analysis of Hallucinations in Large Language Models: Attribution to Prompting Strategies or Model Behavior“. *Frontiers in Artificial Intelligence* (2025). [Kasutatud 15.05.2026]. URL: https://www.frontiersin.org/articles/10.3389/frai.2025.1622292/full.

[42] Aisha Alansari ja Hamzah Luqman. „Large Language Models Hallucination: A Comprehensive Survey“. *Computer Science Review* (2026). [Kasutatud 15.05.2026]. URL: https://www.sciencedirect.com/science/article/pii/S157401372600078X.

[43] Jakob Nielsen. *Usability Engineering*. [Kasutatud 17.05.2026]. Morgan Kaufmann, 1993.

[44] Rensis Likert. „A Technique for the Measurement of Attitudes“. *Archives of Psychology* 140 (1932). [Kasutatud 17.05.2026], lk. 1–55.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Meie, Eliise Karafin ja Eva Kazakovskaia

1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "**Veebipõhise reisihaldusrakenduse kavandamine ja arendamine**", mille juhendaja on Ago Luberg
    1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
    1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

01.06.2026

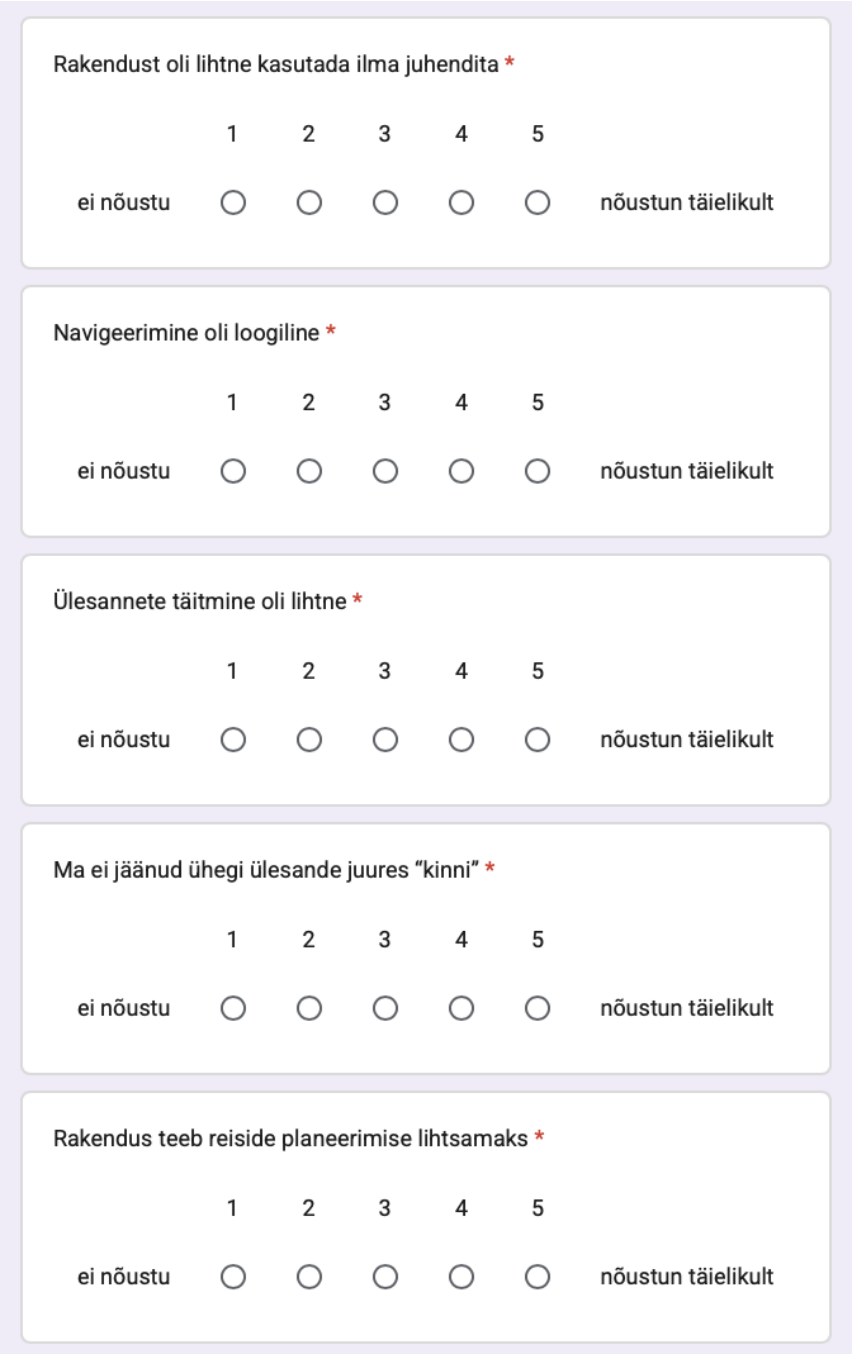
[1] Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Tavakasutajate testülesanded

Tabel 3. Tavakasutajate testülesanded

Testülesanne
Loo uus kasutajakonto.
Logi süsteemist välja ning logi uuesti sisse.
Muuda kasutajakonto parooli.
Vaheta profiilipilti kasutaja seadetes.
Loo uus reis ning määra sellele pealkiri, kirjeldus ja sihtkoht.
Vaheta olemasoleva reisi pilt.
Muuda olemasoleva reisi pealkirja ja kirjeldust.
Lisa reisile uus päev ning loo päevaplaani tegevus koos asukoha ja kellajaga.
Muuda olemasoleva tegevuse kellaaega ning kontrolli päevaplaani ümberjärjestumist.
Lisa tegevuse juurde fail (nt pilet või ajakava dokument).
Kustuta tegevuse juurest eelnevalt lisatud fail.
Sisesta uus reisikulu ja määra kuluga seotud osalejad.
Lisa reisile eelarve (nt 100 eurot söögi jaoks).
Lisa kulule fail (nt tšekk või arve).
Kustuta kuluga seotud fail.
Lisa reisile mitu osalejat ning kontrolli nende kuvamist osalejate vaates.
Kutsu uus osaleja reisile e-posti teel.
Kasuta AI-põhist reisisoovituste funktsionaalsust kasutaja seadetes.
Filtreeri reise ajaperioodi järgi.
Vaata üle reisi üldinfo, päevaplaan ja kulude vaade.

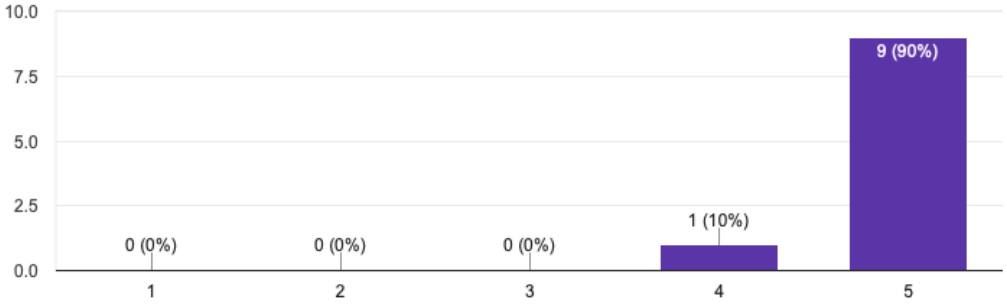
Lisa 3 – Tavakasutajate rakenduse kasutajakogemuse küsimustik



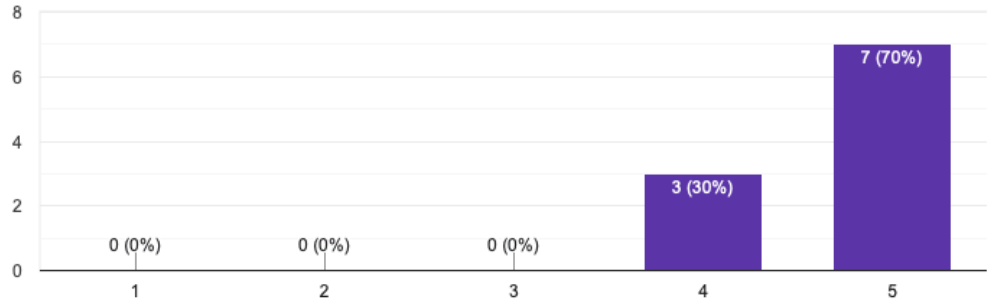
Joonis 4. Kasutajakogemuse küsimustiku vaade

Joonis 5. Kasutajakogemuse küsimustiku jätk

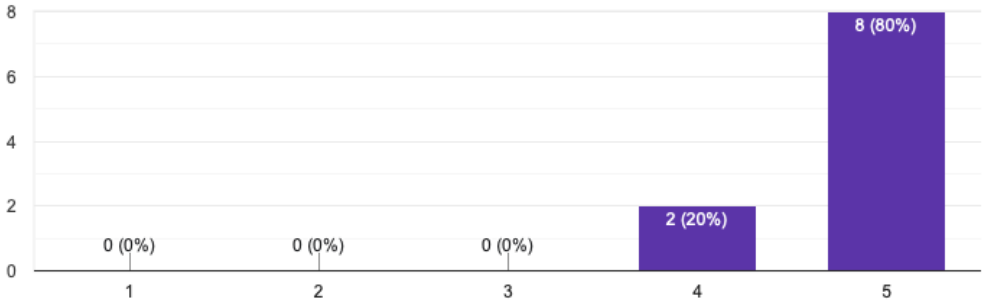
Lisa 4 – Tavakasutajate rakenduse kasutajakogemuse küsimustiku vastused



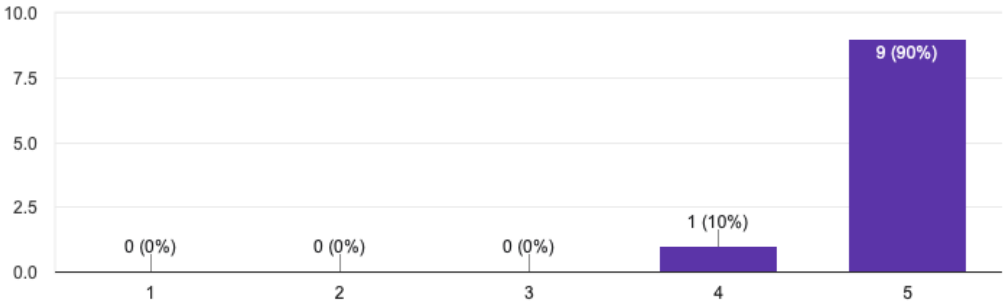
Joonis 6. Rakendust oli lihtne kasutada ilma juhendita



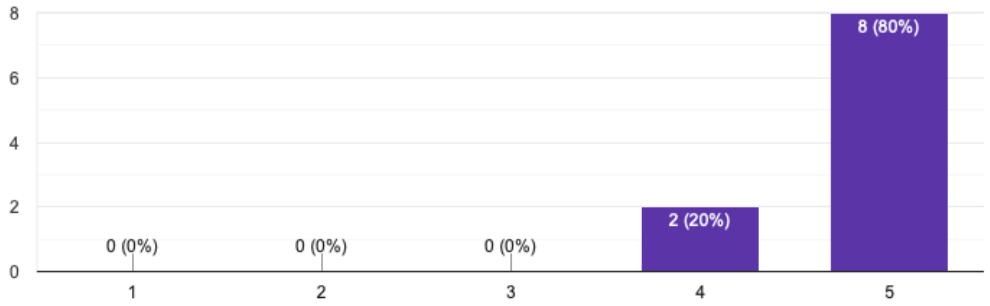
Joonis 7. Navigeerimine oli loogiline



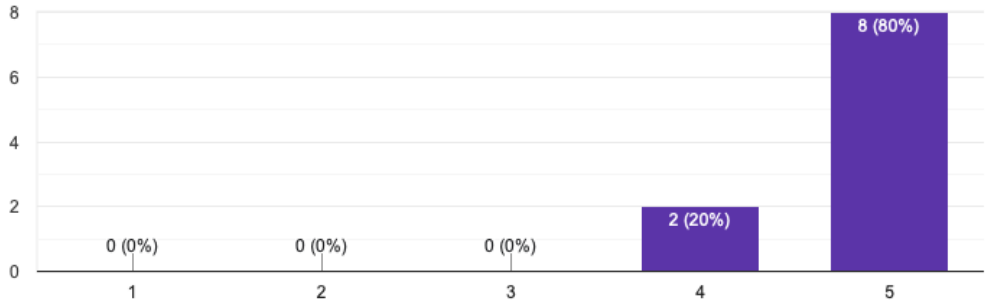
Joonis 8. Ülesannete täitmine oli lihtne



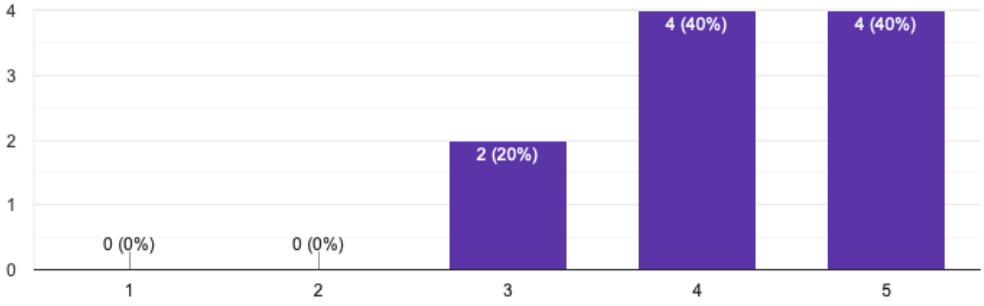
Joonis 9. Kasutajad ei jäänud ülesannete täitmisel kinni



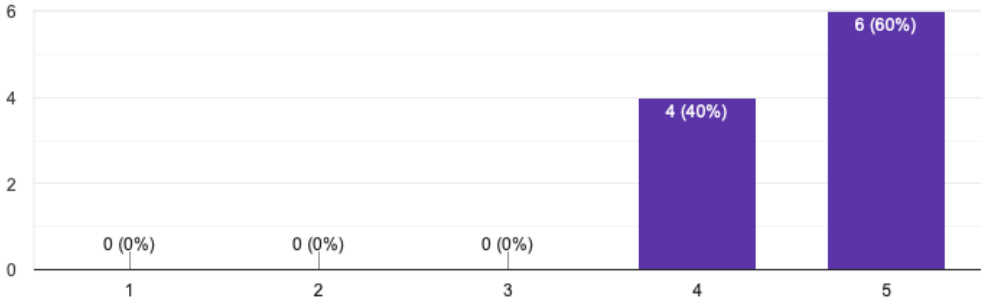
Joonis 10. Rakendus lihtsustas reiside planeerimist



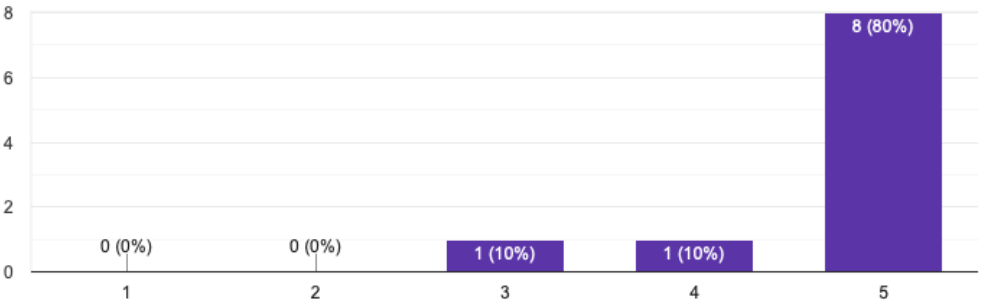
Joonis 11. Kulude jagamise funktsionaalsuse kasulikkus



Joonis 12. Failide lisamise funktsionaalsuse kasulikkus

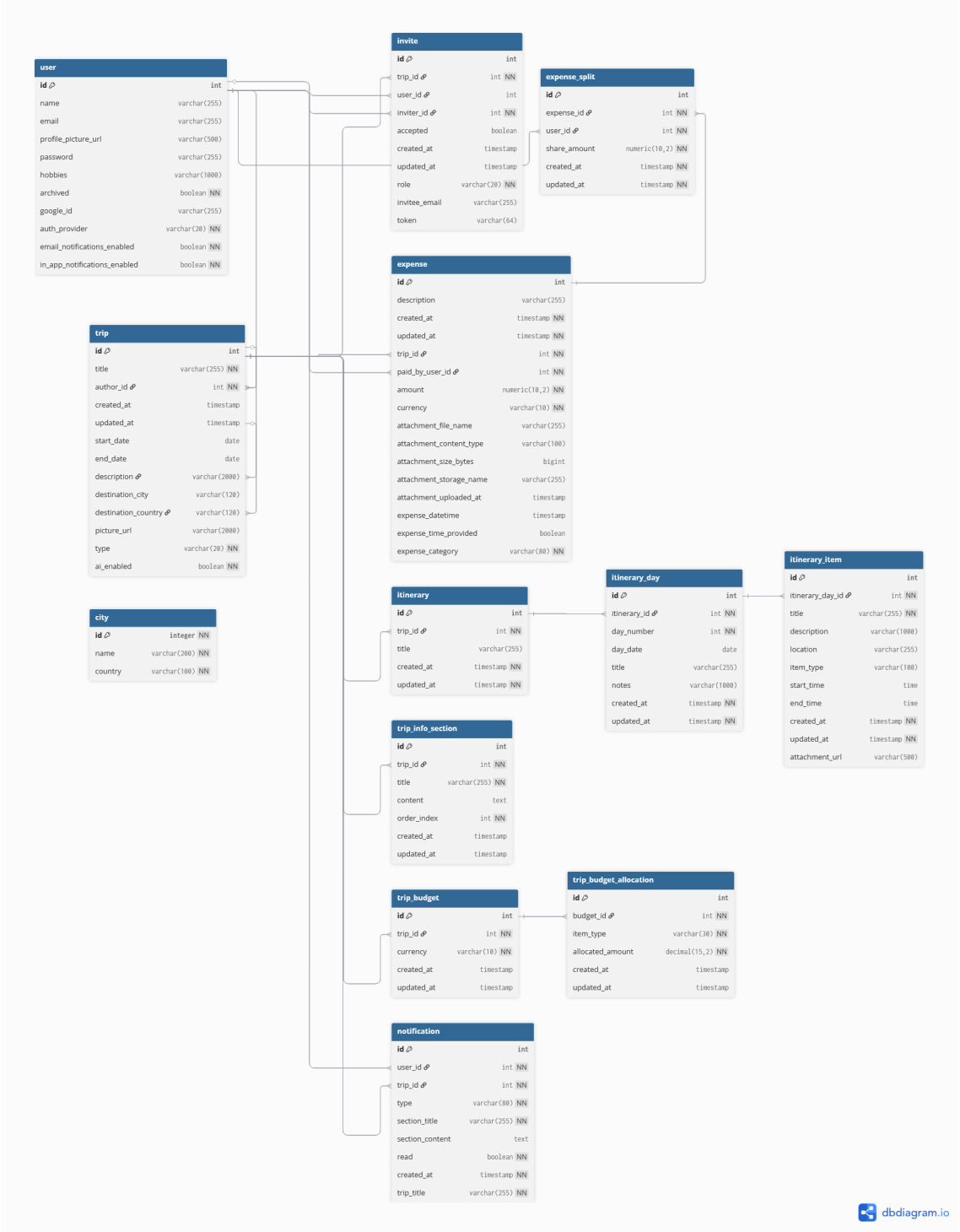


Joonis 13. AI-põhiste soovituste kasulikkus



Joonis 14. AI mõju reisiplaani kiiremale koostamisele

Lisa 5 – Andmebaasi skeem



Joonis 15. Süsteemi andmebaasi skeem

Lisa 6 – *Prompt*: päevaplaani automaatne täitmine

```
You are an expert travel planner.
Generate itinerary items for ONE specific trip day.

Return ONLY valid JSON with this schema:
{
"items": [
{
"title": "string",
"itemType": "TRANSPORT | ACCOMMODATION | FOOD | ACTIVITY | OTHER",
"description": "string or null",
"location": "string or null",
"startTime": "HH:mm or null",
"endTime": "HH:mm or null"
}
],
"explanation": "short plain-language summary"
}

Rules:
- Only valid JSON, no markdown or extra text.
- 3-8 items unless context suggests fewer.
- Respect trip and day context.
- Prioritize additionalRequest if provided.
- If fillMode is APPEND, avoid duplicating existingItems and generate only items
that complement them.
- If fillMode is APPEND, new items must not conflict with existingItems. Do not
create overlapping time ranges, duplicate reservations, duplicate activities, contradictory
```

```
locations, or impossible logistics. Ensure all generated items fit coherently around
the existing schedule for the day
- Use realistic times when possible, otherwise null.
- itemType must match allowed values.
- Titles must be short and specific.
- explanation must be 1-2 short sentences written for the traveler. Briefly explain
what was added and what was optimized for (for example budget, pace, sightseeing,
food, or logistics). Do not include technical details, system reasoning, prompt
references, scheduling logic, fillMode, existingItems, or any implementation-specific
information.

Input:
tripTitle: {tripTitle}
tripDescription: {tripDescription}
destinationCity: {destinationCity}
destinationCountry: {destinationCountry}
tripStartDate: {tripStartDate}
tripEndDate: {tripEndDate}
totalTripDays: {totalTripDays}
dayNumber: {dayNumber}
dayDate: {dayDate}
dayTitle: {dayTitle}
dayNotes: {dayNotes}
fillMode: {fillMode}
additionalRequest: {additionalRequest}
existingItems: {existingItems}
```

Lisa 7 – *Prompt*: soovitatud reisisihtkoha genereerimine

Alljärgnev *prompt* kirjeldab tehisintellekti komponenti, mida kasutatakse kasutaja sisendi põhjal ühe sobiva reisisihtkoha soovitamiseks. Mudelile antakse sisendina kasutaja huvid ja eelistused ning väljundina tagastatakse üks soovitatud sihtkoht koos lühikese põhjendusega.

```
You're a professional trip organizer in a trip planning agency with 25 years of
work experience.
You will be given the user's description about their hobbies and interests. Your
task is to recommend a trip destination based on the context and explain the reasoning
behind it.
Always recommend only one destination. Do not include links. Keep reasoning precise.

Output should follow this example:
{
"recommended_destination": "Tokyo, Japan",
"reasoning": "User likes anime and asian food."
}

{{description}}
```

Lisa 8 – Rakenduse funktsionaalsed ja mittefunktsionaalsed nõued

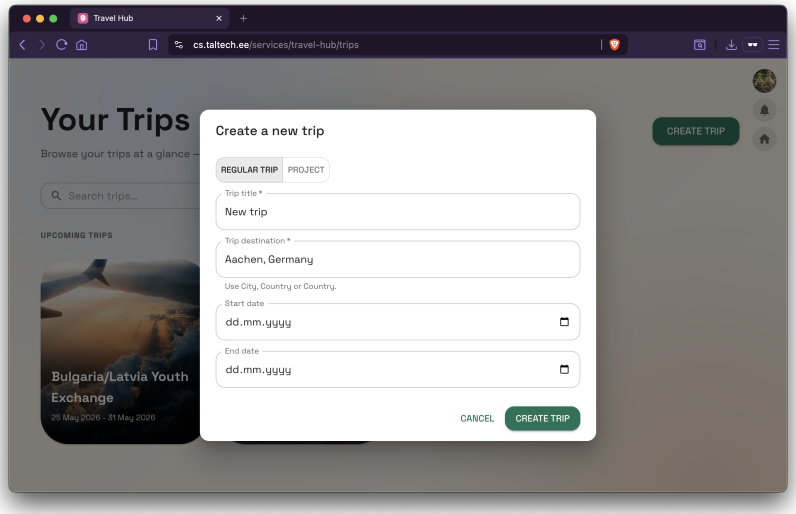
Tabel 4. Funktsionaalsed nõuded ja nende täitmine

Nõue	**Täitmine**
Süsteem peab võimaldama kasutajatel registreerida konto ning sisse logida autentitud kasutajana.	Täidetud ja valideeritud
Süsteem peab võimaldama kasutajatel luua, vaadata, muuta ja kustutada kahte tüüpi reisi - tavareis ja projektireis.	Täidetud ja valideeritud
Süsteem peab võimaldama koostada reiside päevapõhiseid ajakavasid ning lisada tegevusi.	Täidetud ja valideeritud
Süsteem peab võimaldama kasutajatel kutsuda teisi osalejaid reisile e-posti teel ning rakenduse siseselt.	Täidetud ja valideeritud
Süsteem peab toetama rollipõhist ligipääsuhaldust, kus erinevatel kasutajatel on erinevad õigused.	Täidetud ja valideeritud, olemas 5 erinevat rolli
Süsteem peab võimaldama hallata reisiga seotud kulusid ning siduda need konkreetsete osalejatega.	Täidetud ja valideeritud
Süsteem peab võimaldama üles laadida kuludega seotud dokumente (nt arved ja kviitungid).	Täidetud ja valideeritud
Süsteem peab võimaldama genereerida reisiga seotud aruandeid struktureeritud kujul.	Täidetud ja valideeritud
Süsteem peab võimaldama eksportida aruandeid PDF-formaadis.	Täidetud ja valideeritud
Süsteem peab võimaldama kasutajatel vaadata ja hallata oma profiili ning seadeid.	Täidetud ja valideeritud
Süsteem peab võimaldama saata teavitusi kasutajatele oluliste muudatuste korral (nt ajakava uuendused).	Täidetud ja valideeritud
Süsteem peab pakkuma tehisintellektil põhinevat funktsionaalsust reisiplaanide ja soovituste genereerimiseks.	Täidetud (boonus funktsionaalsus), valideeritud demo- ja teststsenaariumites

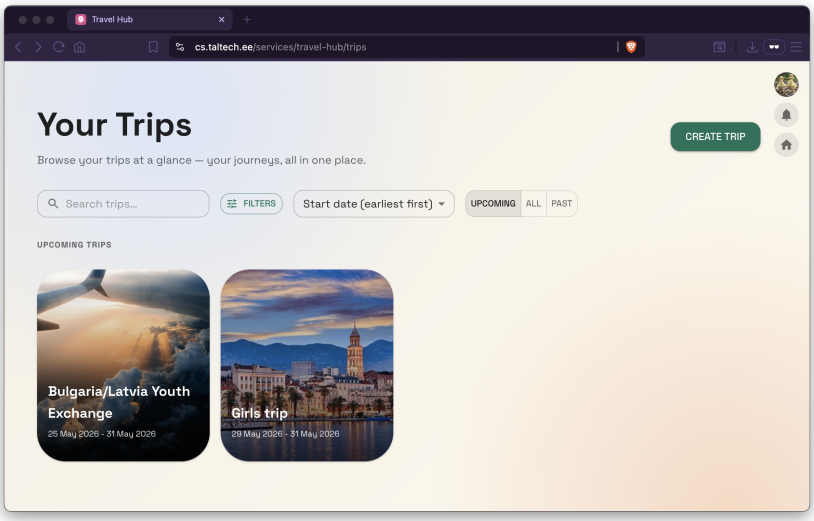
Tabel 5. Mittefunktsionaalsed nõuded ja nende täitmine

Nõue	**Täitmine**
Süsteem peab tagama piisava jõudluse, võimaldades kasutajatel sooritada põhilisi toiminguid ilma märgatava viivituseta.	Täidetud
Süsteem peab olema turvaline, kaitstes kasutajate andmeid autentimise ja autoriseerimise mehhanismidega.	Täidetud ja turvakihtide kaudu valideeritud
Süsteem peab olema kasutajasõbralik ning intuitiivne, võimaldades kasutajatel funktsioone kasutada ilma põhjaliku juhendamiseta.	Täidetud ja kasutajate tagasiside põhjal hinnatud
Süsteem peab olema skaleeritav, et toetada kasutajate arvu ja andmemahtude kasvu tulevikus.	Täidetud (mikroteenuste arhitektuur)
Süsteem peab olema töökindel ning tagama andmete säilimise ka võimalike vigade või katkestuste korral.	Täidetud ja arhitektuurselt tagatud (PostgreSQL)
Süsteem peab olema hooldatav, võimaldades arendajatel koodi lihtsasti täiendada ja muuta.	Täidetud ja arhitektuuriliselt disainitud (kihiline arhitektuur)
Süsteem peab tagama ühilduvuse erinevate seadmete ja veebilehitsejatega, et kasutajad saaksid rakendust kasutada erinevates keskkondades.	Täidetud ja UI tasemel testitud

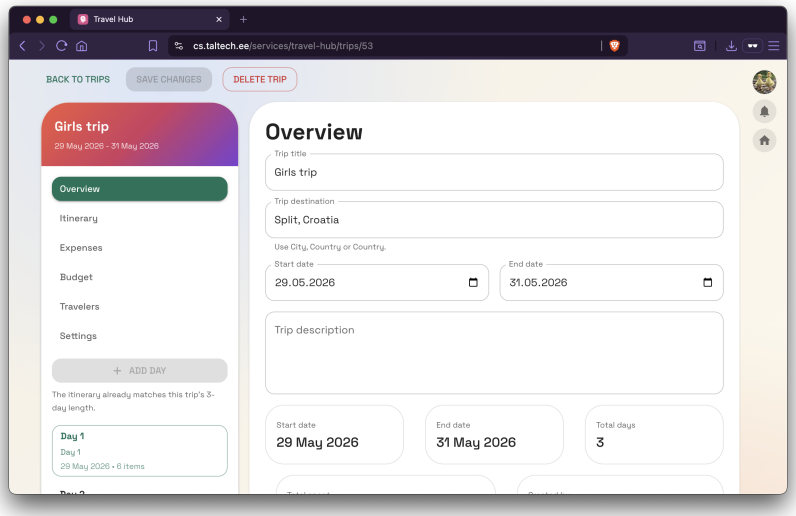
Lisa 9 – Ekraanipildid kasutajaliidesest



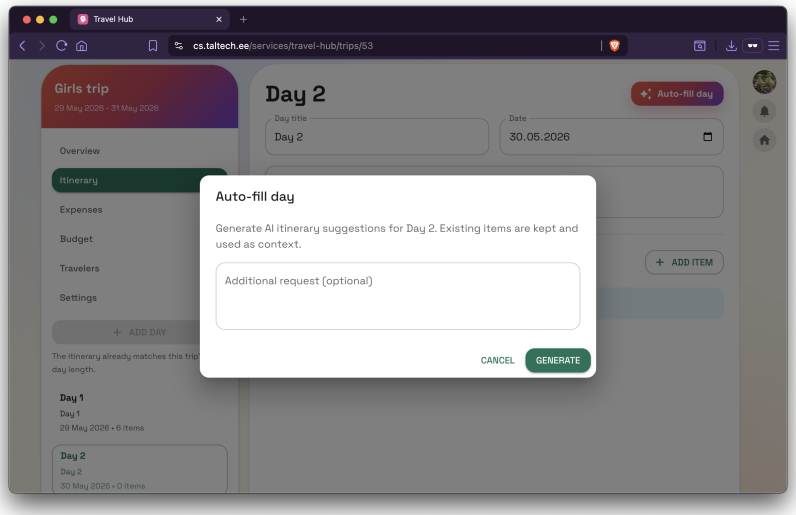
Joonis 16. Uue reisi loomise vaade



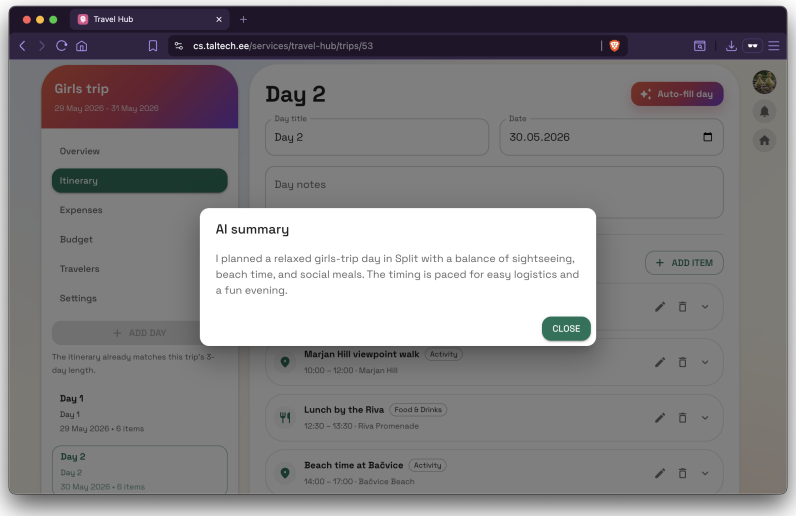
Joonis 17. Reiside ülevaade



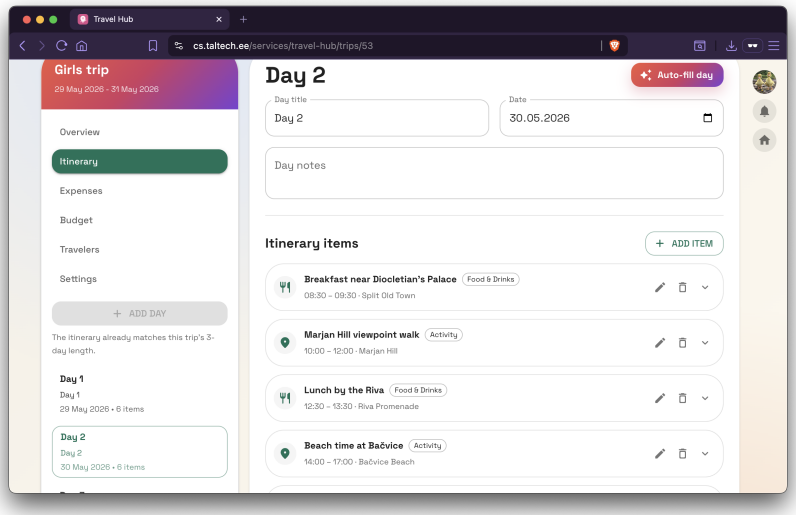
Joonis 18. Reisi detailne ülevaade



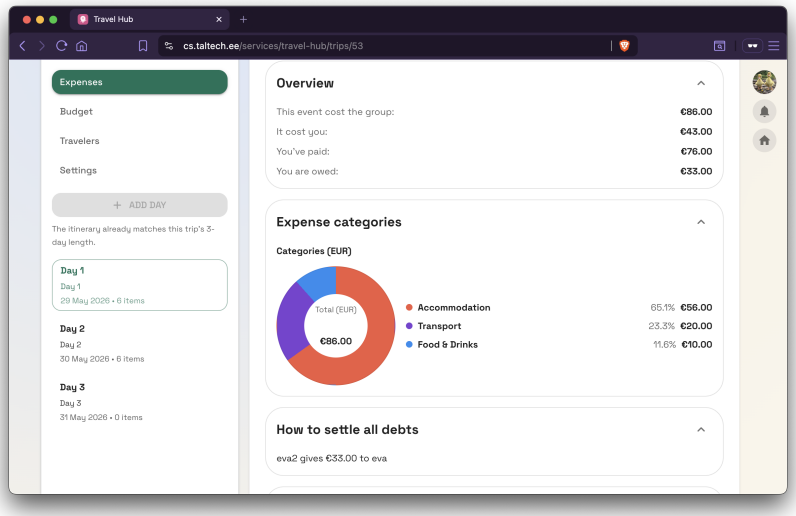
Joonis 19. Tehisintellekti abil reisi automaatne täitmine



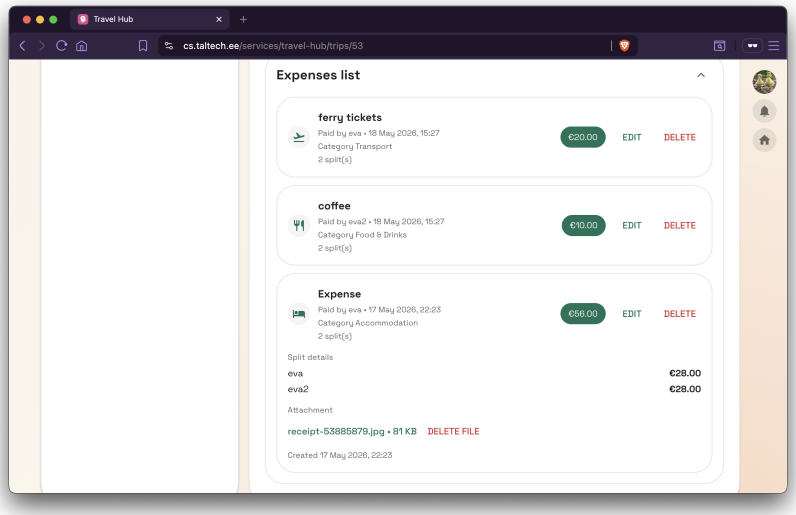
Joonis 20. Automaatse täitmise tulemuste kuvamine



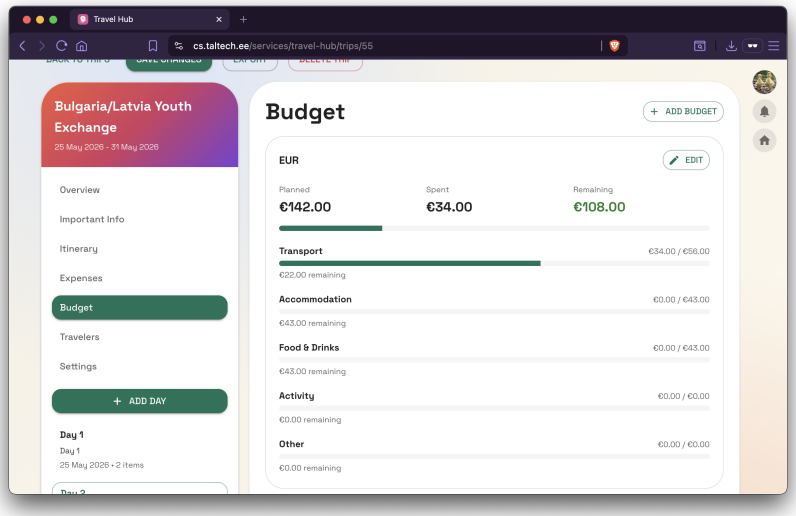
Joonis 21. Päevaplaani vaade



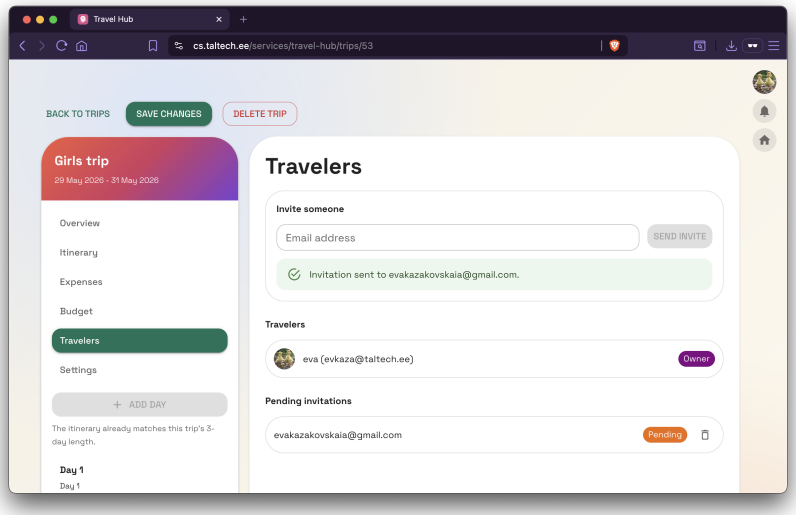
Joonis 22. Kulude ülevaade



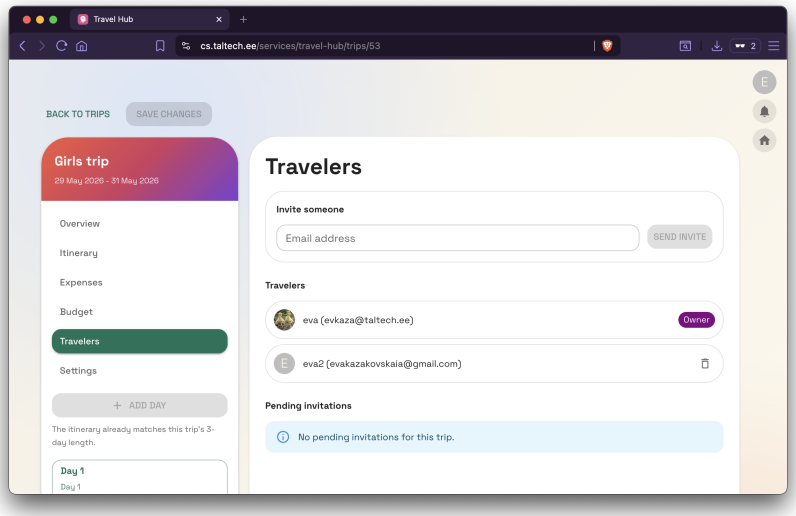
Joonis 23. Kulude lisamise ja haldamise vaade



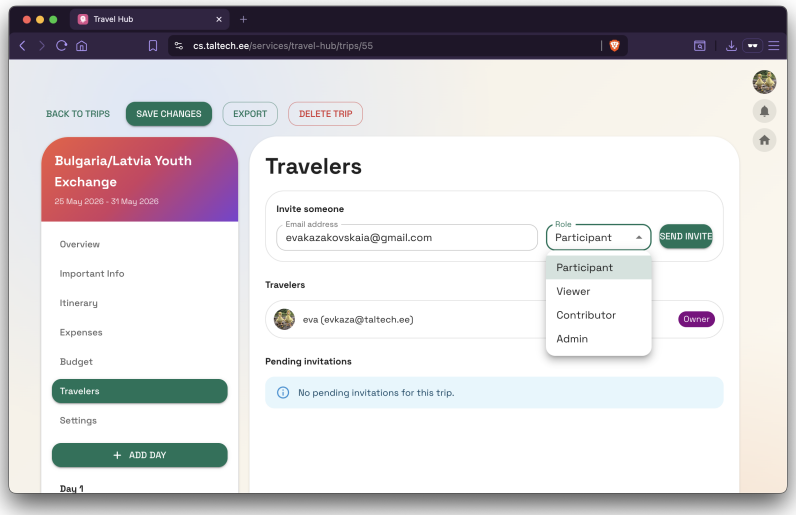
Joonis 24. Eelarve ülevaade ja haldus



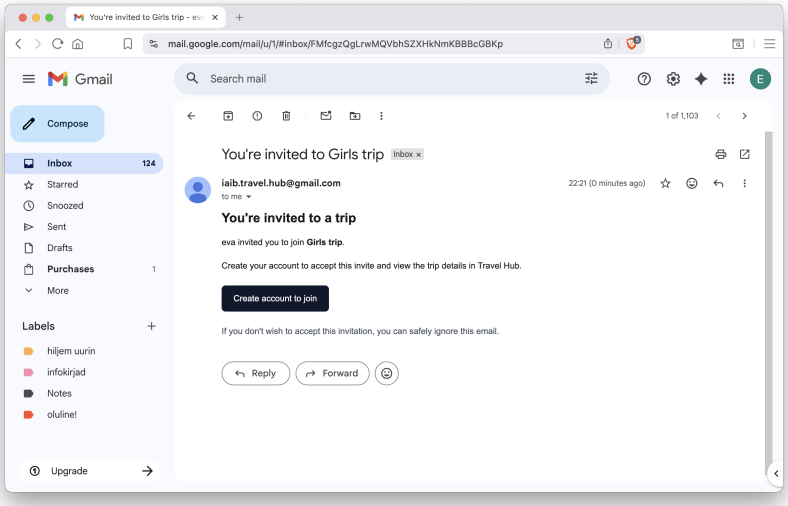
Joonis 25. Reisikaaslaste haldusvaade



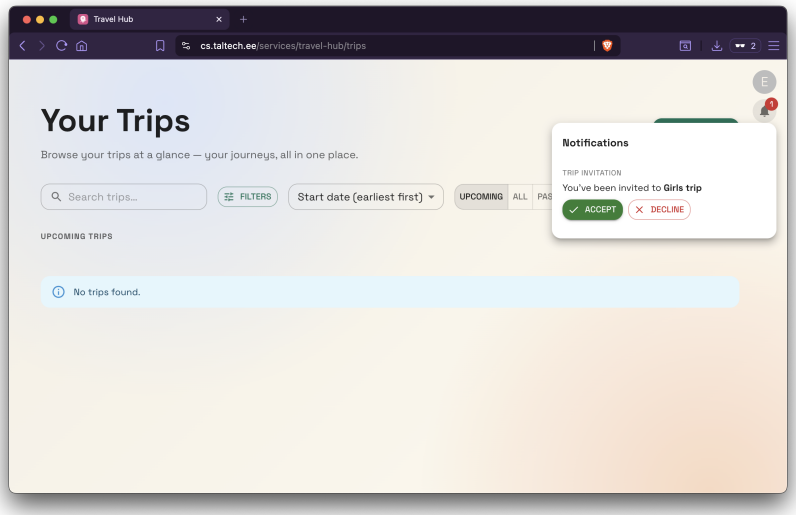
Joonis 26. Aktsepteeritud reisikaaslaste loend



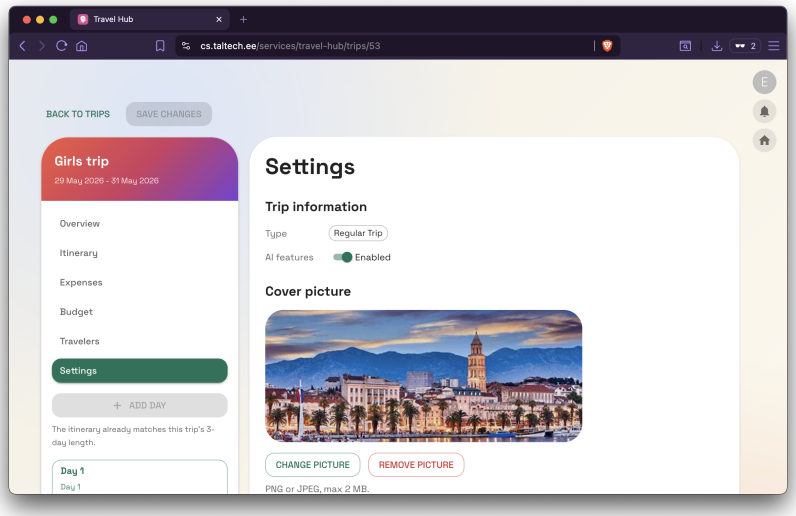
Joonis 27. Reisikaaslaste rollide määramine



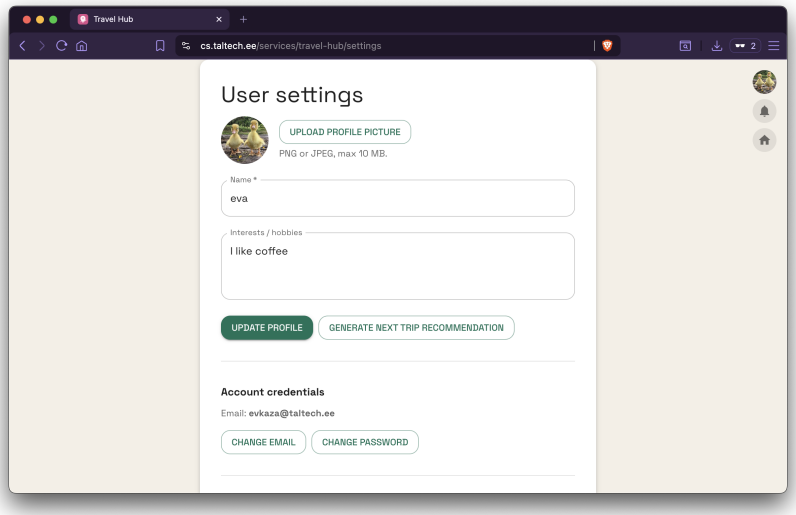
Joonis 28. Kutse saatmine e-posti teel



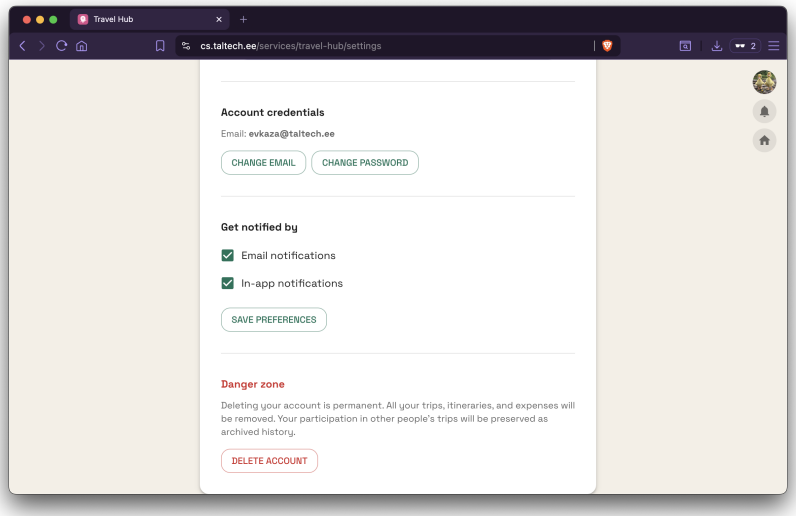
Joonis 29. Rakendusesisene kutse teavitus



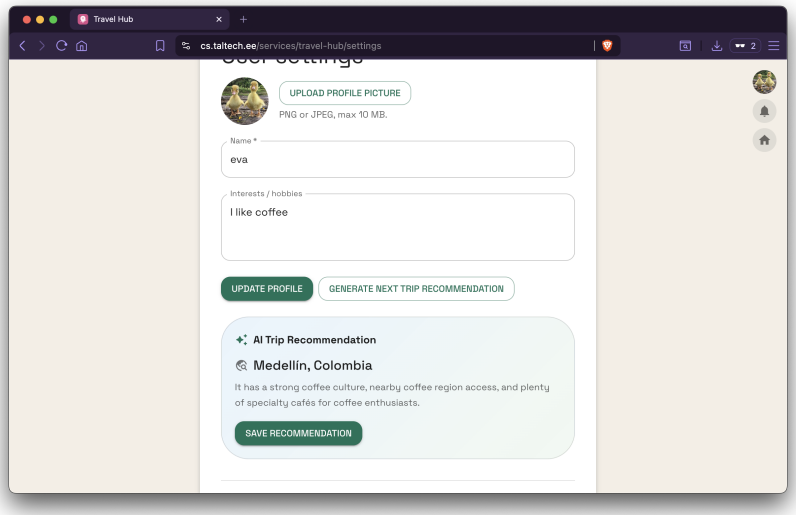
Joonis 30. Reisi üldseadete vaade



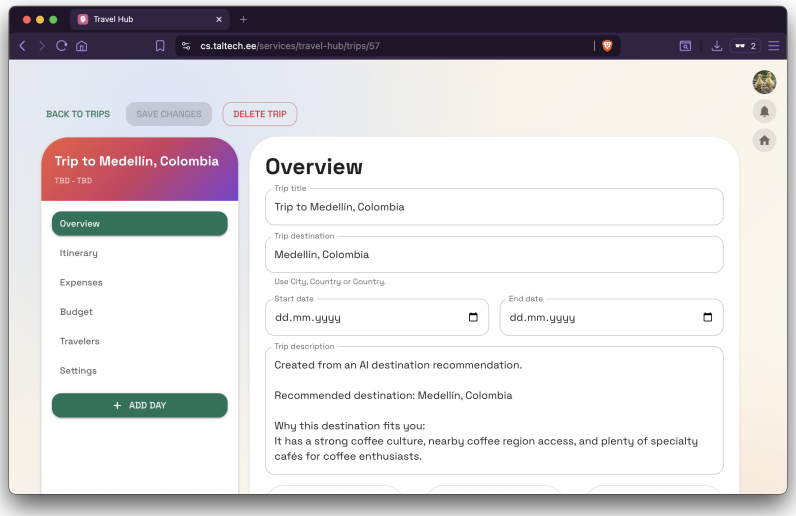
Joonis 31. Kasutaja seaded (vaade 1)



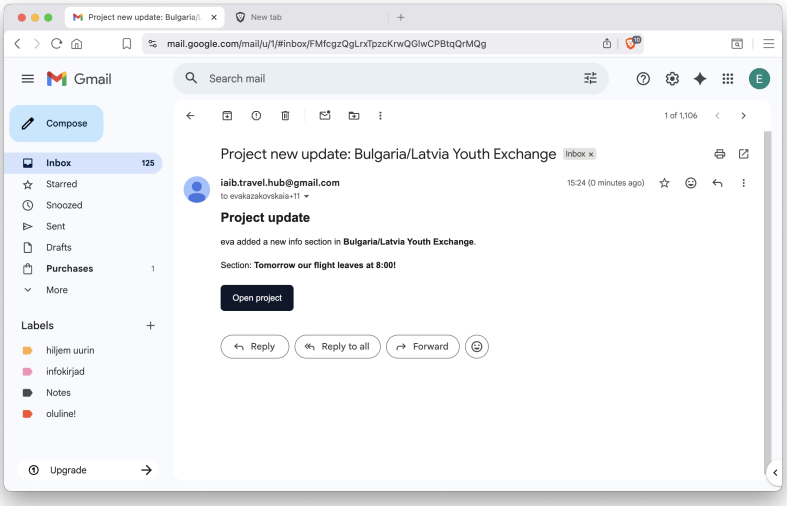
Joonis 32. Kasutaja seaded (vaade 2)



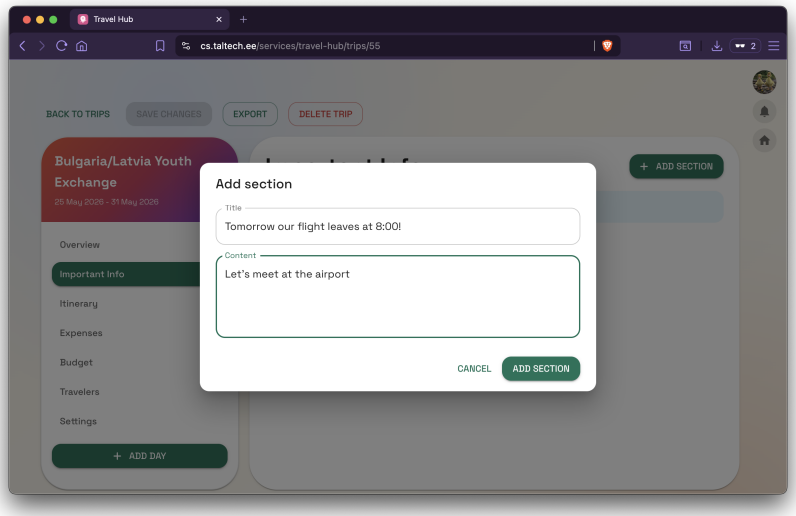
Joonis 33. Reisisoovituste sisendvaade



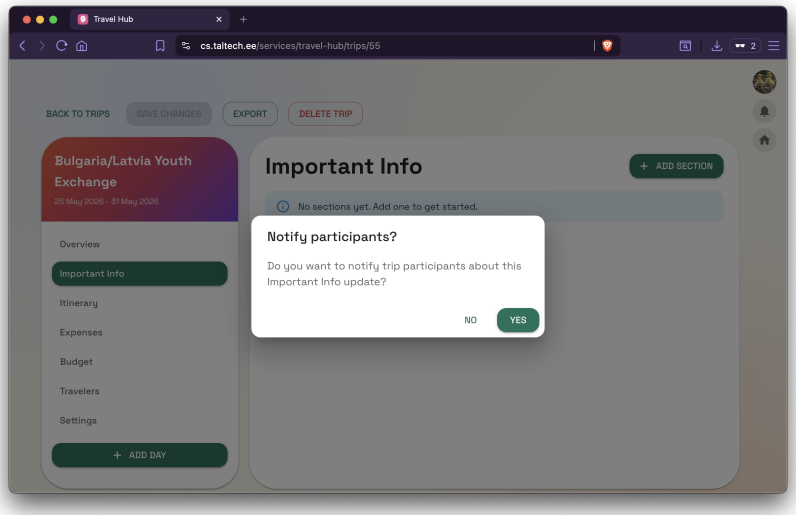
Joonis 34. Reisisoovituste genereeritud tulemused



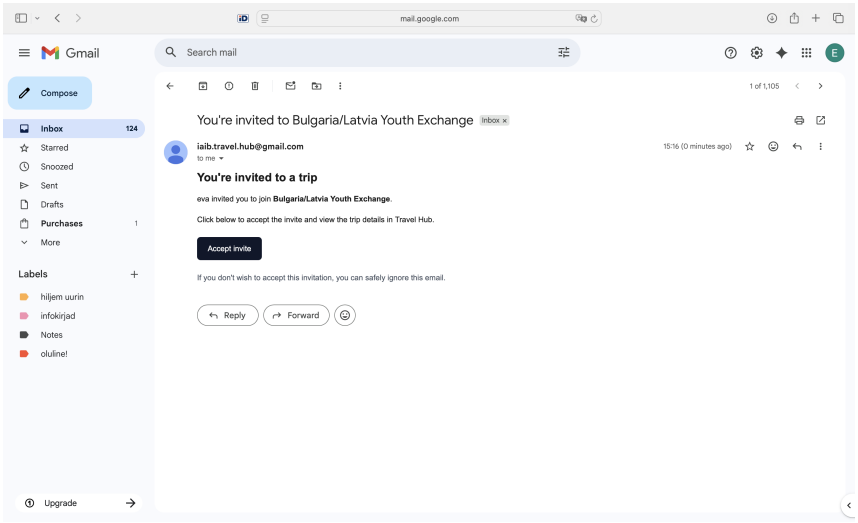
Joonis 35. E-posti teavitus süsteemist



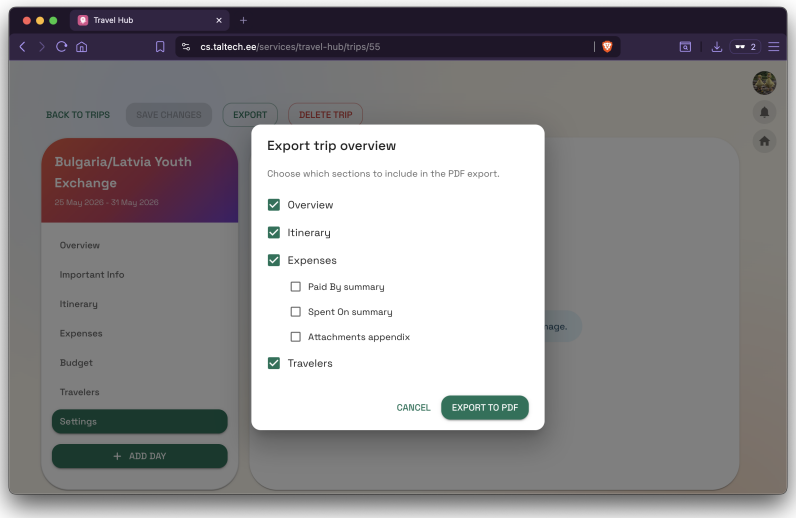
Joonis 36. Projekti teavitus



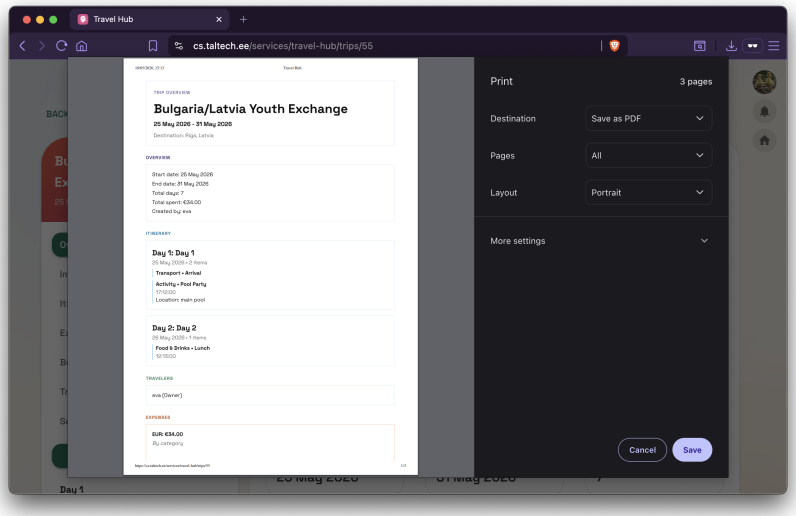
Joonis 37. Projekti teavituse kinnitus



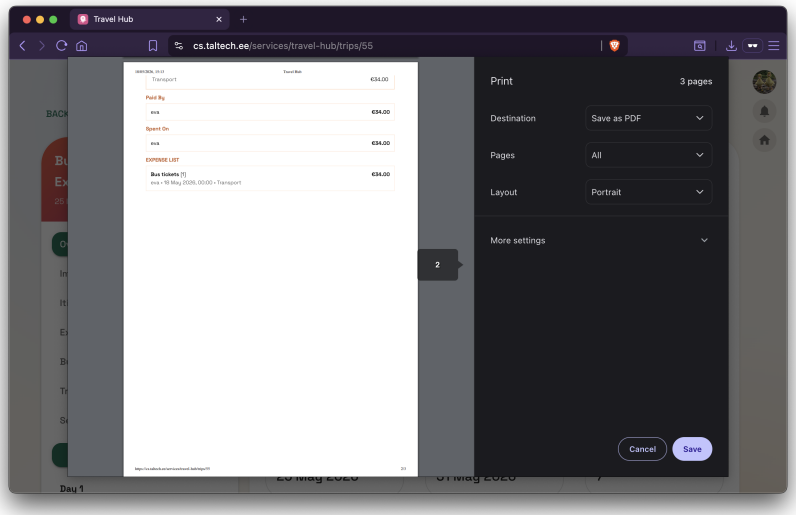
Joonis 38. E-posti teavitus reisiga seotud muudatustest



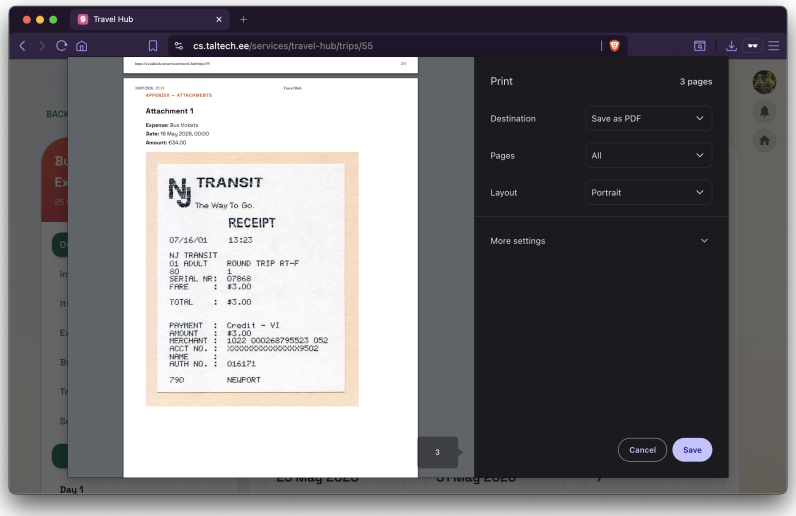
Joonis 39. Reisi ekspordi seaded



Joonis 40. Ekspordi eelvaade



Joonis 41. Ekspordi eelvaade jätk



Joonis 42. Ekspordi failide eelvaade