

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Mari-Nicole Berezovski 222994IAIB

# **Veebipõhine rakendus isikliku eelarve koostamiseks**

Bakalaureusetöö

Juhendaja: Meelis Antoi

MSc

Tallinn 2025

## **Autorideklaratsioon**

Kinnitan, et olen 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.

Autor: Mari-Nicole Berezovski

04.06.2025

## **Annotatsioon**

Käesoleva bakalaureuse töö raames loodi tasuta veebipõhist eelarvestamise rakendust, mis aitab täita seniste lahenduste puudujääke ning toetab noori rahaplaneerimisel. Peamiseks rakenduse funktsionaalsusteks said eelarvete planeerimine, kulude jälgimine ning säästmiseesmärkide seadmine.

Eesmärgi saavutamiseks viidi läbi turu-uuring, analüüsid olemasolevaid tööriistu ning kaardistades sihtrühma vajadusi, mille põhjal koostati nõusid antud rakenduse jaoks.

Rakenduse elluviimiseks uuriti erinevaid tehnoloogiaid ning valiti välja need, mis sobisid kõige paremini seatud eesmärkide saavutamiseks. Valideerimiseks kasutati nii automaatkui ka kasutajatestimist. Kasutajatestimise põhjal koostati testianalüüs ning parandati ilmnunud vead. Tulemuste kinnitamiseks viidi läbi teine kasutajatestimise voor, mis osutus edukaks.

Tulemuseks koostatakse tervislikku rakenduse prototüübi, mis vastab kõikidele püstitatud nõudele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 7 peatükki, 65 joonist.

## **Abstract**

### **Web Application for Personal Budget Planning**

The aim of this bachelor's thesis was to create a web-based application for managing personal finances. The goal was to fill a gap in the current market by offering a solution that supports young users at the beginning of their financial journey. The application is designed to be simple, user-friendly, and accessible. Its main functionalities include budgeting through expense tracking and setting savings goals.

To support the development process, a thorough analysis of existing popular financial applications was carried out. This helped to identify common flaws and essential features that users expect. After, a user survey was conducted to collect insights of the users preferences, and expectations for the selected application. This ensured the new application would meet the real needs.

When developing the desired application, various technologies were analyzed, and the best suited to the project's goals were selected to ensure efficient and effective implementation.

To validate the outcome, user testing was conducted, during which the main issues were identified, analyzed, and resolved. To confirm the improvements, a second round of user testing was carried out, which was successful and confirmed that the application met the defined requirements and user expectations.

As a result, a full-stack application was created that met all the established requirements.

The thesis is in Estonian and contains 33 pages of text, 7 chapters, 65 figures.

## Lühendite ja mõistete sõnastik

Agiilne arendus	<i>Agile development</i> , tarkvaraarenduse meetod, mis põhineb iteratiivsel ja järk-järgulisel lähenemisel
API	<i>Application Programming Interface</i> , rakendusliides serveriga suhtluseks
Axios	HTTP klienditeek JavaScriptis
BCrypt	Paroolide räsi loomise algoritm, mida kasutatakse krüpteerimiseks
<i>Branch</i>	Haru — eraldi arendusliin versioonihalduses
<i>Bug</i>	Viga programmis, mis põhjustab soovimatut käitumist
<i>Changelog</i>	Fail või dokument, mis loetleb muudatused projektis
<i>Changelog-master</i>	Changelog'i põhiharus hoitav peamine muudatuste loend
CI/CD	<i>Continuous Integration/Continuous Deployment</i> , tarkvaraarenduse protsess, mis automatiseerib testimise ja väljalaske
ColorPicker	Kasutajaliideses värvi valimise komponent
<i>Controller</i>	Komponent, mis võtab vastu HTTP-päringud, suunab need teenusele ning tagastab vastuse
<i>Criteria repository</i>	Spetsiifiline Spring Data JPA päringute koostamise viis
DevOps	Arendus- ja operatsiooniprotsesside ühendamise filosoofia ja tööriistakogum
DTO	<i>Data Transfer Object</i> , objekt, mida kasutatakse andmete edastamiseks erinevate kihtide vahel
EmojiPicker	Komponent, mis võimaldab kasutajatel valida emotikone
<i>Endpoint</i>	Lõpp-punkt ehk URL, mille kaudu saab teha päringuid veebira- kendusele
<i>Entity</i>	Andmebaasiobjekti esitus Java klassina, mida kasutatakse JPA-s
Esirakendus	<i>Frontend</i> , rakenduse kasutajaliidese osa
<i>Foreign key</i>	Viitvõti, seob andmebaasitabelid omavahel
<i>Getter</i>	Meetod, mis tagastab objekti välja väärtuse
Git	Versioonihaldussüsteem lähtekoodi muutuste jälgimiseks
GitLab	Veebipõhine Git hoidlahaldusteenus CI/CD tuge
Google	Otsingumootor ja teenusepakkuja
Gradle	Automatiseerimis- ja ehitustööriist
HMAC-SHA256	Krüptograafiline räsi algoritm andmete autentimiseks
HttpOnly	Küpsise atribuut, mis takistab JavaScripti kaudu ligipääsu

I18n	<i>Internationalization</i> , tarkvara lokaliseerimise ettevalmistus
Issue	Probleem või ülesanne projekti halduskeskkonnas
JAR	<i>Java ARchive</i> , Java klasside ja ressursside pakettifail
JPA	<i>Java Persistence API</i> , Java standard andmeobjektide püsivaks salvestamiseks
JSON	JavaScript <i>Object Notation</i> , kergekaaluline andmeformaad
JWT	<i>JSON Web Token</i> , autentimisviis veebisessioonide haldamiseks
Kanban	Visuaalne projektijuhtimise meetod töövoogude haldamiseks
Label	Silt või märgistus, mida kasutatakse näiteks Git'is või UI elementidel
Let's Encrypt	Tasuta SSL sertifikaatide väljaandja
LineChart	Joondiagramm, mis kujutab andmete muutust ajas
Liquibase	Tööriist andmebaasi skeemi haldamiseks ja migreerimiseks
Logimine	Rakenduse sündmuste salvestamine logidesse
Lombok	Java teek, mis genereerib boilerplate koodi (nt getterid/setterid)
Main	Git'i vaikimisi põhiharu, kuhu koondatakse stabiilsed muudatused
Mapper	Objekti teisendaja, mis seob andmemudeleid omavahel
MapStruct	Java annotatsioonipõhine mapi generaator
Merge request	Taotlus muudatuste liitmiseks Git hoidlas
Maven	Java projektide ehitamise ja haldamise tööriist
Meta	Ettevõtte, mis omab Facebooki ja muid teenuseid
Mikroteenused	Väikesed sõltumatud teenused, mis koos moodustavad suure süsteemi
Mock	Testide jaoks loodud võltsobjekt, mis jäljendab tegelikku komponenti
MVC arhitektuur	<i>Model-View-Controller</i> , tarkvaraarhitektuuri muster
NGINX	Veebiserver ja pöördproksiteenus
Node.js	JavaScripti käituskeskkond serveripoolsete rakenduste loomiseks
Objektorienteeritud	Programmeerimisparadigma, kus andmed ja käitumine on objektides
OpenJDK	Avatud lähtekoodiga Java arenduskomplekt
PieChart	Sektordiagramm, mis näitab osade suhet tervikusse
Pipeline	Automatiseeritud töövoog tarkvara ehitamiseks, testimiseks ja juurutamiseks
PostgreSQL	Võimas avatud lähtekoodiga relatsiooniline andmebaasisüsteem
Primary key	Unikaalne veerg, mis identifitseerib tabeli kirjeid
Proxy	Vahekihina töötav server
Repository	Koodibaasi või andmete hoidla
REST	<i>Representational State Transfer</i> , veebiteenuste arhitektuur

RESTful API	API, mis järgib REST põhimõtteid
RGB	Värviesitusmudel punase, roheline ja sinisega
<i>RuntimeException</i>	Käitusaegne viga Javas, mida ei pea ilmtingimata käsitlema
<i>Service</i>	Loogikakiht rakenduses, mis töötleb ärioloogikat
<i>Setter</i>	Meetod, mis määrab objekti välja väärtuse
SocketJs	Teek WebSocketi ühenduste varundamiseks HTTP kaudu
SPA	<i>Single Page Application</i> , üheleheküljeline veebirakendus
SQL	<i>Structured Query Language</i> , päringukeel andmebaasidega suhtlemiseks
SSL/TSL	Turvaprotokollid andmeside krüpteerimiseks
STOMP protokoll	Sõnumsideprotokoll, mida kasutatakse WebSocketi kohal
Swagger	Tööriist REST API dokumenteerimiseks ja testimiseks
Tagarakendus	<i>Backend</i> , serveripoolne loogika, mis töötleb andmeid
UI	<i>User interface</i> , kasutajaliides
URL	<i>Uniform Resource Locator</i> , aadress, mis osutab veebiressursile
UTC	<i>Coordinated Universal Time</i> , koordineeritud maailma aeg
UX	<i>User experience</i> , kasutaja kogemus veebilehel
<i>Volume</i>	Andmete salvestusühik konteinerites (nt Docker volume)
WebSocket	Protokoll reaajas kahepoolseks suhtluseks brauseri ja serveri vahel
XML	<i>eXtensible Markup Language</i> , struktureeritud andmete kirjeldamiseks
<i>Zero-based</i>	Eelarvestamise meetod, kus iga kulu peab olema igas perioodis põhjendatud, alustades nullist

## Sisukord

1	Sissejuhatus.....	10
2	Probleem.....	11
3	Taustauuring.....	12
3.1	Olemasolevad lahendused.....	12
3.1.1	YNAB.....	12
3.1.2	GoodBudget.....	13
3.1.3	EveryDollar.....	13
3.1.4	Swedbanki teenus „Minu eelarve“.....	14
3.1.5	Excel tabelipõhine eelarvestamine.....	14
3.2	Olemasolevate lahenduste analüüs.....	15
3.3	Kasutajauuring.....	16
4	Analüüs.....	17
4.1	Nõuete määramine.....	17
4.1.1	Funktsionaalsed nõuded.....	17
4.1.2	Mittefunktsionaalsed nõuded.....	19
4.2	Tehnoloogiate valik.....	20
4.2.1	Projekti haldus.....	20
4.2.2	Tagarakenduse tehnoloogiad.....	20
4.2.3	Esirakenduse tehnoloogiad.....	22
4.2.4	Andmebaasi tehnoloogiad.....	23
4.2.5	Veebirakenduse arhitektuur.....	23
5	Prototüübi arendamine.....	24
5.1	Arenduspõhimõtted.....	24
5.2	Andmebaas.....	25
5.3	Tagarakendus.....	26
5.4	Esirakendus.....	28
5.5	Turvalisus.....	29



5.6	Automaattestimine .....	31
5.7	Serverisse ülespanek .....	31
6	Tulemused ja valideerimine .....	33
6.1	Kasutajatestimine 1. voor ja selle analüüs.....	33
6.2	Parandused.....	35
6.3	Kasutajatestimine 2. voor ja selle analüüs.....	36
6.4	Nõuetele vastavus .....	37
6.5	Analüüs .....	41
6.6	Edasised arnedusvõimalused .....	41
7	Kokkuvõte.....	42
	Kasutatud kirjandus .....	43
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	46
	Lisa 2 – Kasutajauuringu küsimustik.....	47
	Lisa 3 – Kasutajauuringu tulemused.....	51
	Lisa 4 – Kasutajavoog .....	53
	Lisa 5 – Andmebaasimudel.....	54
	Lisa 6 – Kasutajatestimise esimese vooru küsimustik .....	57
	Lisa 7 – Kasutajatestimise esimese vooru tulemused .....	60
	Lisa 8 – Kasutajatestimise teise vooru küsimustik.....	63
	Lisa 9 – Kasutajatestimise teise vooru tulemused.....	65
	Lisa 10 – Rakenduse kuvatõmmised .....	67

# 1 Sissejuhatus

Tänapäeva noored vanuses 18-25 seisavad oma esimest sissetulekut teenides ja iseseisvat elu alustades silmitsi mitmete finantsalaste väljakutsetega. Kuigi see vanusegrupp on astumas oma esimesi samme tõelise majandusliku iseseisvuse suunas, puuduvad sageli vajalikud oskused oma rahaasjade tõhusaks haldamiseks [1]. See finantsoskuste lõhe võib viia impulsiivsete kulutuste ja finantsstressini, mis omakorda mõjutab elukvaliteeti. See rõhutab vajadust juurdepääsetavate tööriistade järele, mis võimaldaksid kasutajatel taastada kontrolli oma finantsseisundi üle.

Käesoleva lõputöö eesmärk on luua veebirakenduse prototüüp isikliku eelarvestamise jaoks, mis täidaks olemasolevate lahenduste poolt jäetud puudujääki ning pakuks tasuta platvormi finantsplaneerimiseks.

Eesmärgi täitmiseks viiakse läbi turuuuring, mis koosneb nii olemasolevate lahenduste põhjalikust analüüsist kui ka sihtrühma vajaduste kaardistamisest kasutajauuringu kaudu. Kogutud info põhjal koostatakse rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded, mida viiakse täies mahus ellu, kasutades Java Spring Boot raamistikku, Reacti põhist kasutajaliidest ning PostgreSQL andmebaasi.

Järgnevalt kirjeldatakse töö kõiki etappe põhjalikumalt. Esmalt püstitatakse probleem, toetades allikatele. Seejärel viiakse läbi taustauuring, mille käigus analüüsitakse alternatiivseid lahendusi ja viiakse läbi kasutajauuring, püstitades konkreetseid nõudeid rakendusele. Valitakse sobivad tehnoloogiad vastavalt seatud eesmärkidele ning antakse ülevaade kogu arendusprotsessist. Edasi korraldatakse kasutajatestimine, mille tulemusi analüüsitakse ning tehakse jooksvalt järelusi. Lõpuks võetakse kogu töö kokku, hinnates loodud lahenduse vastavust püstitatud nõuetele.

## 2 Probleem

Eelarve planeerimine on noorte jaoks oluline, see aitab vältida finantsprobleeme ja saavutada majanduslikku stabiilsust. Elukalliduse kasv suurendab igapäevaseid vajadusi ning seega on oluline rahalist planeerimist võimalikult varakult alustada. Uuringud näitavad, et hea finantsplaneerimine võimaldab noortel paremini hallata sissetulekuid ja väljaminekuid, vältides rahalisi raskusi ja aidates saavutada majanduslikku heaolu [2], [3].

Noorte finantsplaneerimise alused hõlmavad eelkõige eelarvestamist, säästmist ja investeerimist. Eelarve koostamine on tõhus viis kulude jälgimiseks ning aitab vähendada ebavajalikke väljaminekuid. Eelarve koostamise protsess koordineerib rahalisi eesmärke kindla ajaperioodi jooksul [4]. Regulaarne eelarvestamine aitab noortel tagada, et kulutused ei ületaks sissetulekuid ning võimaldab tulevasi finantsotsuseid realistlikult hinnata. Uuringud näitavad, et isegi finantsplaneerimise teadmistega ülikooli tudengitel, näiteks raamatupidamise eriala üliõpilastel, võib rahatarkus olla madal [5]. See tähendab, et kuigi omandatakse teadmisi eelarvestamise ja juhtimisarvestuse kohta, ei pruugi tudeng seda osata rakendada isikliku finantsplaneerimise jaoks [6], [7].

Finantsplaneerimine annab noortele turvatunde ja stabiilsust, luues aluse paremale rahalisele tulevikule ja võimaldades paremat elukvaliteeti [8]. Efektivne rahavoo juhtimine eeldab sissetulekute ja kulude tasakaalustamist ning rakendamist eelarvet kui vahendit isikliku rahalise kontrolli saavutamiseks [9]. Seetõttu on noorte jaoks oluline investeerida aega ja ressursse finantskirjaoskuse arendamisse, et osata edukalt hallata oma rahalisi vahendeid. Sarnane rakendus on juba loodud algklassiõpilaste jaoks [10].

Finantsstress ei piirdu üksnes rahaga – see mõjutab kõiki eluvaldkondi, sealhulgas suhteid ja vaimset tervist. Uuringud näitavad, et rahaline ebastabiilsus võib põhjustada stressi, mis on võrreldav suurte elumuutustega. Rahaline pingeline olukord võib sageli avalduda ärevusena, tuues esile finantsraskustega kaasneva sügava emotsionaalse mõju. Tõhusad eelarvestamise tavad võivad aga taastada kontrollitunde [11].

## **3 Taustauuring**

Käesolevas peatükis analüüsitakse olemasolevaid alternatiivseid lahendusi, tuuakse välja nende puudused ja sarnasused. Samuti viiakse läbi kasutajauuring, mille tulemuste põhjal koostatakse põhjalik analüüs.

### **3.1 Olemasolevad lahendused**

Tänapäeval on olemas mitmeid rakendusi, mis üritavad lahendada püstitatud probleemi, kuid hoolimata tungivast vajadusest tõhusate eelarve koostamise tööriistade järele, ei vasta paljud olemasolevad isikliku eelarve koostamise rakendused sageli kasutajate ootustele.

#### **3.1.1 YNAB**

YNAB (You Need a Budget) on üks kõige tuntumaid ja laialdasemalt kasutatavaid isikliku eelarvestamise ja finantside haldamise rakendusi maailmas.

Kuigi see on saavutanud populaarsuse oma paindliku lähenemise tõttu, on mitmeid põhjuseid, miks see rakendus ei pruugi olla parim valik kõikide kasutajate jaoks.

Rakendust ei ole võimalik kasutada ilma maksmata – funkionaalsuse juurde pääseb ainult prooviperioodi alustades, mis nõuab pangakonto andmete sisestamist. Prooviperiood kestab 34 päeva ning selle möödumisel võetakse tasu pangakontolt automaatselt. Paljude jaoks võib see tekitada ebamugavusi ning kahtlusi turvalisuses.

You Need a Budget võimaldab koostada eelarveid, määrates iga rahatähe kindlasse kategooriasse. Rakendus pakub ka õppevahendeid, mis aitavad eelarvestamises paremini orienteeruda, ning toetab sünkroonimist olemasoleva kaardi ja/või pangakontoga. Siiski on oluline teada, et hetkel on sünkroonimise tugi saadaval ainult Ameerika Ühendriikide pankadele [12].

### 3.1.2 GoodBudget

GoodBudget on finantside jälgimise rakendus, mis aitab kasutajatel hallata oma eelarvet ja jälgida kulutusi. Rakendus pakub mitmeid kasulikke funktsioone, kuid sellel on ka teatavad kitsaskohad, mis võivad mõjutada kasutajakogemust.

Üks GoodBudget'i eripäradest on see, et iga kulu või tulu lisamisel tuleb määrata isik või teenus, kellele raha läheb. Valik tuleb teha piiratud etteantud nimekirjast, mis sisaldab spetsiifilisi valikuid (nt "sõõrikud" või Kanada ettevõtted), mistõttu võib sobiva kategooria leidmine olla keeruline. Samuti rakendus ei ole eriti kasutajasõbralik ka kategooriate haldamisel – uusi kategooriaid ei saa tähistada värvide või emotikonidega, mis lihtsustaksid visuaalset eristamist. See võib muuta rakenduse kasutamise keerukamaks võrreldes teiste sarnaste tööriistadega.

Tasuta versioonis on kasutajatel võimalik luua erinevaid eelarvekategooriaid ning määrata iga kategooria jaoks igakuised kulutuste piirangud. Kuid kategooriate arv on piiratud, mis võib osutada probleemiks, kui on vaja rohkem paindlikkust. Samuti saab sama kontot kasutada erinevatest seadmetest, mis võimaldab sünkroniseerida kulutusi ja sissetulekuid mitme seadme vahel.

Valides tasulise versiooni, saab kasutaja ligipääsu suuremale kategooriate arvule ning täiendavatele funktsioonidele, nagu aruannete loomine ja pikaajaliste säästude jälgimine, mis võivad olla eriti kasulikud neile, kes soovivad põhjalikku finantsanalüüsi [13].

### 3.1.3 EveryDollar

EveryDollar on Ramsey Solutions poolt loodud veebirakendus, mis põhineb *zero-based* ehk nullpõhisel eelarvestamisel. See tähendab, et iga sissetulekust saadud eurole määratakse kindel eesmärk, nii et kuu lõpuks on kogu raha jaotatud erinevatesse kulukategooriatesse. Selline lähenemine võib aga olla piirav neile, kes otsivad paindlikumat ja vähem struktureeritud viisi oma rahanduse haldamiseks.

Eelarve koostamine algab sissetulekute sisestamisest, millele järgneb summa jaotamine erinevatesse kategooriatesse, kuni kogu raha on ära planeeritud. Igas kategoorias tuleb määrata vastav summa, mida plaanitakse kulutada. See nõuab põhjalikku ettevalmistust ja

kõigi rahaallikate arvestamist.

Tasuta versioonil on piiratud funktsionaalsus, kuid ka tasulises versioonis on piirangud kulukategooriate ja eelarvete arvule. Samuti rakendus on suunatud peamiselt Ameerika Ühendriikide turule, kuna valuutaks on ainult USA dollar. See võib olla takistuseks välismaa kasutajatele, kuna teised valuutad ei ole toetatud [14].

### **3.1.4 Swedbanki teenus „Minu eelarve“**

Swedbanki teenus „Minu eelarve“ jälgib automaatselt kõiki konto peal tehtuid tehinguid ja koostab neist visuaalseid diagramme, mis annavad selge ülevaade kuu kulutustest. Rakendus automaatselt jaotab kulusid erinevatesse kategooriatesse, näiteks toit, transport ja kodukaubad. Kui soetatakse ühe tehinguga nii toitu kui kodutarbeid, on võimalik tehtud kulusid käsitsi eraldi kategooriatesse jaotada, et tagada täpset ülevaadet. Lisaks saab määrata igakuised kululimiite kindlatele kategooriatele, näiteks piirata toidukulude summat või kehtestada limiidid meelelahutusele, et hoida oma eelarve kontrolli all.

Kuid teenusel on ka teatavad piirangud. Antud rakenduse piiranguks on see, et kasutaja peab olema Swedbanki klient. Käsitsi tehingute lisamine või kulude täiendav sisestamine ei ole võimalik, mis võib olla takistuseks nendele, kes soovivad oma kulutusi täiendada või hallata teistes pankades tehtud tehinguid [15].

### **3.1.5 Excel tabelipõhine eelarvestamine**

Excel on kõige kergemini kättesaadav ja paindlik tööriist kulude jälgimiseks, kuid selle kasutamine eelarve haldamiseks võib olla üsna aeganõudev ja keeruline. Isegi lihtsama personaliseeritud eelarve loomine nõuab põhjalikku ettevalmistust ja süsteemi tundmist. Kui plaanitakse luua täpselt enda vajadustele vastav eelarve, tuleb tutvuda Exceli võimalustega, mõista erinevaid funktsioone ja uurida, millised mudelid on sobivad konkreetse ülesande täitmiseks. Samuti peab olema kogemus tabeltöötlemise kasutamises.

Selline lähenemine sobib inimestele, kes on valmis investeerima aega ja vaeva, et luua täpselt nende vajadustele vastavat süsteemi. Samuti peab olema kindel visioon sellest, kuidas kulusid hallata ja milliseid andmeid jälgida. Exceli puhul on väga oluline põhjalik planeerimine, kuna üks väike viga, näiteks vale valem või andmete sisestamine vale

vorminguga, võib kogu süsteemi toimimise rikkuda ja tekitada vigu, mis võivad olla raskesti parandatavad.

Lisaks on oluline arvestada, et selleks, et Exceli abil kulude haldamine toimiks tõhusalt, tuleb pidevalt värskendada ja täiendada loodud mudeleid. See on pidev töö, kus üks väike muutus võib tähendada vajadust kogu süsteemi ülevaatamiseks. Seega, kui eesmärgiks on täpne ja usaldusväärne kulude jälgimine, tuleb olla valmis kulutama nii aega kui ka ressursse, et tagada süsteemi tõrgeteta toimimist [16].

## **3.2 Olemasolevate lahenduste analüüs**

Eespool käsitletud rakenduste analüüsist võib tuua välja mitte ainult sarnaseid puudusi, vaid ka ühiseid jooni ja funktsionaalsusi, mida on oluline arvestada uue lahenduse väljatöötamisel.

### **Puudused**

- Kohandamisvõimaluste puudumine kitsendab rahalise ülevaate täpsust ja mitmekesisust – kasutajad ei saa oma eelarvet vastavalt isiklikele vajadustele või harjumustele täiendada.
- Kohustuslik on tasulise versiooni soetamine, et pääseda juurde olulistele funktsionaalsustele.
- Rakenduse kasutamiseks on vajalik pangakonto või kaardiga ühendamine, mis on omakorda piiratud ainult teatud pankade või teenustega.
- Nullpõhine eelarvestamine ei pruugi oma jäikuse tõttu sobida kõigile, kuna see piirab eelarve isikupärastamist ja paindlikkust.
- Kasutajad ei pruugi end pangakonto ühendamisel mugavalt tunda, mis kujutab endast puudust juhul, kui ühendamine on kohustuslik.
- Valuutavalikute ja teenusepakkujate piiratus raskendab rakenduse kasutamist väljaspool esmast turgu.

### **Ühised jooned**

- Eelarve koostamine igapäevaste väljaminekute jälgimise ja planeerimise kaudu.
- Sihtide seadmine kindla summa ulatuses rahaliste eesmärkide saavutamiseks.

- Visuaalsed ülevaated ja aruanded, mis aitavad edusammude ja kulutuste jälgimisel.
- Mobiil- ja veebiversioonide ühilduvus, tagades sujuva kasutajakogemuse erinevates seadmetes ja platvormidel.

### 3.3 Kasutajauuring

Teema vajalikkuse kinnitamiseks ja sobivate funktsionaalsuste määratlemiseks viidi läbi kasutajauuring, mis tugines olemasolevate rakenduste puudustel ja funktsionaalsustel. Kasutati kvantitatiivseid ja kvalitatiivseid meetodeid, et saada terviklik ülevaade kasutajate vajadustest ja ootustest. Eesmärk oli uurida inimeste finantsplaneerimise oskusi ning ootusi rakenduse suhtes. Uuringust võttis osa 35 inimest. Vt Lisa 2

Enamik vastajaid (85,7%) olid vanuses 18–25, mis viitab, et just noored, kes alles alustavad iseseisva elu teekonda ja õpivad rahaasju haldama, tunnevad suuremat huvi finantsplaneerimise vastu (vt Lisa 3, Joonis 5). See näitab valmisolekut investeerida aega rahaasjade õppimisse ning teadlikkust rahalise iseseisvuse tähtsusest.

Finantsplaneerimise oskusi hindas 41,7% keskmiseks, samas kui üle 55% vastajatest tunnistas, et tunneb end ebakindlalt – hinnates oma oskusi kas madalaks või väga madalaks. See viitab vajadusele pakkuda rohkem tuge ja haridust finantskirjaoskuse arendamiseks (vt Lisa 3, Joonis 6).

Kuigi 74,3% vastanutest hetkel oma eelarvet ei planeeri, tunneb 46,2% neist siiski vajadust spetsiaalse rakenduse järele (vt Lisa 3, Joonis 7). Neist, kes juba kasutavad sarnaseid lahendusi, soovib 90% paremat alternatiivi (vt Lisa 3, Joonis 8), mis näitab, et olemasolevad tööriistad ei vasta kasutajate ootustele.

Rakendust soovitakse peamiselt kasutada igapäevaste kulude planeerimiseks ja jälgimiseks (65%), kuid oluliseks peetakse ka säästmise ja suuremate ostude kavandamise tuge (33%). Väiksem osa (8%) soovib planeerida konkreetseid eesmärke, näiteks reise (vt Lisa 3, Joonis 9).

Kasutussagedus varieerub – 30% kasutaks rakendust ostu sooritamisel, 50% kord nädalas ja 40% kord kuus. See näitab, et rakendus peaks pakkuma paindlikke ja kasutajasõbralikke lahendusi erinevate eelistuste jaoks (vt Lisa 3, Joonis 10).



## 4 Analüüs

Selles peatükis koostatakse funktsionaalsed ja mittefunktsionaalsed nõuded eelnevalt kogutud andmete põhjal ning määratakse eesmärgi saavutamiseks sobivaimad tehnoloogiad, võttes arvesse kõiki asjakohaseid aspekte.

### 4.1 Nõuete määramine

Nõuete koostamisel lähtuti olemasolevate veebipõhiste finantslahenduste analüüsist ja läbi viidud kasutajauuringust. Analüüsi tulemusena toodi esile peamised puudused ja kitsaskohad senistes lahendustes, mis on mõjutanud kasutajakogemuse kvaliteeti negatiivselt. Samuti tuvastati korduvaid funktsionaalsusi, mis on olulised eelarve haldamise rakenduse toimimiseks. Kasutajauuring võimaldas koguda sihtrühma ootusi ja eelistusi, pakkudes selget ülevaadet klientide vajadustest rakenduse funktsionaalsuse, kasutusmugavuse ja tehniliste omaduste osas.

Kogutud sisendi põhjal sõnastati funktsionaalsed ja mittefunktsionaalsed nõuded.

#### 4.1.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad, milliseid tegevusi ja funktsioone süsteem peab täitma, et kasutaja saaks rakendust eesmärgipäraselt kasutada. Prototüübi kasutajavoog Lisa 4, Joonis 11

- Kasutajakonto ja autentimine
  - Kasutaja saab luua profiili, määrates nime, e-posti aadressi ja parooli.
  - Kasutaja saab sisse logida, kasutades oma e-posti ja parooli.
  - Kasutaja saab välja logida.
- Otsing ja filtrid
  - Kasutaja saab otsida ja filtreerida eelarveid järgmiste kriteeriumite alusel:
  - **Ajavahemik** – eelarve algus- ja lõppkuupäev, sealhulgas käesolev kuu jms.

- **Kategooria** – eelarvele määratud kategooria alusel.
- **Kategooriate haldamine**
  - Eelarvete loomisel saab kasutaja valida olemasolevate kategooriate seast (nt toit, transport, üür, meelelahutus jne).
  - Kasutaja saab luua isikupärastatud kategooriaid, määrates nime, lühikirjelduse, värvi ja emotikoni visuaalseks eristamiseks.
- **Eelarvete haldamine**
  - Kasutaja saab koostada eelarveid, määrates eelarve tüübi (kulu või sääst), ajavahemiku, kategooria ja planeeritud summa.
- **Kulude jälgimine**
  - Kasutaja saab luua kulueelarveid ning sisestada käsitsi sissemakseid.
  - Rakendus kuvab reaajas eelarve täitumist protsentuaalselt, näidates, kui suur osa eelarvest on juba kulutatud ning kui palju on alles kulutamiseks.
  - Kui kulueelarve ületatakse, näidatakse ületatud summat.
- **Säästmise eesmärgid**
  - Kasutaja saab seada säästmise eesmärgid ja lisada neile sissemakseid.
  - Iga eesmärk on eraldiseisev ning võimaldab sissemakseid koos kirjelduse ja summaga.
  - Rakendus kuvab eesmärgi täitmise progressi visuaalselt, näidates säästetud summat, puuduvat summat ning protsentuaalset täituvust.
  - Kui säästetakse rohkem kui planeeritud, kuvatakse ületatud summa.
- **Sissemaksed eelarvetesse**
  - Kasutaja saab käsitsi sisestada sissemakseid eelarvetesse.
  - Sissemaksele saab lisada kirjelduse ja kulutatud summa.
  - Kasutaja saab sissemakseid kustutada.
- **Eelarvete jagamine**
  - Kasutaja (vastava loaga ehk autor või redigeerija) saab jagada eelarveid ja/või säästmise eesmärgid teiste kasutajatega.
  - Jagamisel määratakse roll:
    - \* **Vaataja:** saab vaadata eelarve või eesmärgi täitumist.
    - \* **Redigeerija:** saab lisada/kustutada sissemakseid ja kasutajaid.

- Aruanded ja analüüs
  - Rakendus kuvab visuaalseid aruandeid sissemaksete kohta:
    - \* **Joondiagramm:** iga punkt graafikul tähistab ühte päeva ning kajastab eelarve kasutajate sissemakseid antud kuupäeval.
    - \* **Sektordiagramm:** eelarvete jaotus kategooriatesse (kulu/sääst) protsentides valitud perioodil
    - \* **Tulpdiagramm:** võrdlus kahe perioodi keskmiste eelarvete summade vahel
- Reaalajas teavitused
  - Rakendus saadab kasutajatele teavitusi juhul kui eelarves kuhu nad kuuluvad toimusid järgmised muutused:
    - \* Kasutaja lisatakse või eemaldatakse eelarvelt.
    - \* Eelarvele lisatakse või sealt eemaldatakse teine kasutaja.
    - \* Tehakse või kustutatakse sissemakse.
    - \* Kulueelarve ületatakse (kuvatakse ületamise määr).
    - \* Kulueelarve on 90% täis (hoiatus).
    - \* Säästmise eesmärk sai täis (teavitus).
    - \* Säästmise eesmärk on peaaegu täis (ergutus).
  - Teavitused sisaldavad:
    - \* Summa (kui asjakohane).
    - \* Kasutaja e-posti aadress (keda lisati/eemaldati või kes toimingut tegi).
    - \* Tegevuse kellaeg.

#### 4.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded kirjeldavad rakenduse kvaliteedinõudeid – ehk mis omadused peavad süsteemil olema, et tagada kasutusmugavust, töökindlust ja tõhusat toimimist.

- **Kohandatavus** – Rakenduse kasutajaliides peab olema kohanduv erinevate ekraanisuurustega (nt lauaarvuti, sülearvuti, mobiilseadmed).
- **Autentimine** – Kasutaja autentimine peab toimuma turvaliselt, kasutades JWT-d (*JSON Web Token*), mis salvestatakse turvalise küpsise sisse (nt *HttpOnly*), millel on määratud aegumisaeg.

- **Kasutajasõbralikkus** – Rakendus peab olema kasutajasõbralik ja intuitiivne navigeerimisel.
- **Veebiturvaline** – Rakendus peab olema ligipääsetav domeeninime kaudu ning kasutama turvalist HTTPS-ühendust (*Hypertext Transfer Protocol Secure*).
- **Mitmekeelne** – Rakendus vaikumisi eesti keeles, kuid peab olema kättesaadav ka ingliskeelsena.

## 4.2 Tehnoloogiate valik

Tehnoloogiate valikul lähtuti eelnevalt kirjeldatud nõuetest ja süsteemile seatud eesmärkidest. Valitud tehnoloogiad toetavad rakenduse funktsionaalsust, kasutusmugavust ja töökindlust ning vastavad esitatud vajadustele.

Esirakendus loodi JavaScripti põhjal, kasutades React-raamistikku. Tagarakenduse arenduseks valiti Java-põhine Spring Boot raamistik koos PostgreSQL andmebaasiga, mille versioonihalduseks kasutati Liquibase'i ning konteinerihalduseks Dockerit. Projekti versioonihaldus toimus GitLabis. Rakenduse serverisse paigaldamiseks kasutati NGINX veebiserverit.

### 4.2.1 Projektihaldus

Projektihaldustarkvaraks valiti GitLab, kuna see põhineb Git peal, pakkudes tugevat versioonihaldust. GitLab võimaldab luua ja hallata projekte, koodi, töövooge ning probleemide jälgimist, samuti ülesannete ja bugide haldamist. Lisaks toetab pidevat integreerimist ja pidevat juurutamist (CI/CD), mis võimaldab automaatse testide käivitamist, juurutusvoogude haldamist, koodi stiili kontrollimist ja muid kvaliteedikontrolle [17].

### 4.2.2 Tagarakenduse tehnoloogiad

Tagarakendus on tarkvarasüsteemi osa, mis vastutab andmete töötlemise, äriloogika ning suhtluse eest andmebaasi ja kasutajaliidese vahel, hallates sealjuures päringuid, autentimist ja andmete valideerimist. Tehnoloogia valikul lähtuti seadistamise lihtsusest, põhjalikust dokumentatsioonist ja RESTful API toe olemasolust.

RESTful API arendus järgib REST-arhitektuuri põhimõtteid ja võimaldab erinevatel süsteemidel (nt veebirakendustel, mobiilirakendustel ja serveritel) omavahel andmeid vahetada,

kasutades standardset HTTP-protokolli. Selle eelisteks on teenuspõhine lähenemine ning äri loogika ja kasutajaliidese eraldamine, mis muudab süsteemi paindlikumaks ja kergemini hooldatavaks [18].

Tehnoloogiate valikul kaaluti C# ja Pythoni kasutamist, kuna mõlemad on võimekad ja laialdaselt levinud programmeerimiskeeled. C# pakub tugevat integratsiooni veebiraamistikega (nt ASP.NET), kuid keskendub eelkõige MVC-arhitektuurile, mis ei sobinud antud ülesande RESTful API-põhise lähenemisega [19]. Python on tuntud oma lihtsuse ja paindlikkuse poolest, kuid kuna autoril puudus varasem kogemus veebirakenduste arendamisel Pythoni abil, otsustati kasutada mõnda muud, paremini tuttavat tehnoloogiat [20].

Java on mitmeplatvormiline objektorienteeritud programmeerimiskeel, mille tugevusteks on hea skaleeritavus, turvalisus, stabiilsus ja suur arendajate kogukond, mis toetab keele arengut ja pakub rohkelt tuge. Lisaks on Java jaoks olemas mitmeid tööriistu ja raamistikke, mis muudavad arendamise lihtsamaks ja kiiremaks [21].

Üks juhtivaid Java ökosüsteemi raamistikke on Spring Boot. Uuringute kohaselt kasutab seda ligikaudu 40% arendajatest, mis peegeldab selle populaarsuse kasvu. Selle edu aluseks on lihtsus ja võimsad funktsioonid, mis võimaldavad kiiresti arendada mikroteenuseid ja suuremahulisi rakendusi. Suurim eelis on minimaalne konfiguratsioonivajadus, mis kiirendab arendusprotsessi ja muudab rakenduste loomise sujuvamaks [22].

Spring Boot on ideaalne valik veebirakenduse arendamiseks, kuna see pakub automaatset konfigureerimist, mis vähendab manuaalset seadistamist ja kiirendab arenduse alustamist. Raamistik lihtsustab REST API-de loomist ning võimaldab kiiresti luua iseseisvaid rakendusi, mida saab lihtsalt käivitada [23].

API dokumenteerimiseks kasutati Swaggerit – avatud lähtekoodiga tööriista, mis on mõeldud REST API-de dokumenteerimiseks. Swagger võimaldab luua arusaadavat ja süsteemset dokumentatsiooni iga API-lõpp-punkti (*endpoint*) kohta, kirjeldades põhjalikult kasutatavaid parameetreid, vastuseid ning muud olulist teavet tagarakenduse liideste kohta [24].

### 4.2.3 Esirakenduse tehnoloogiad

Esirakendus vastutab kasutajaliidese ja kasutajakogemuse eest. See on süsteemi osa, mida lõppkasutajad näevad ja millega nad suhtlevad. Antud töö raames valiti kolme juhtiva esirakenduse raamistiku vahel: Angular, Vue.js ja React.

Angular on Google'i arendatav TypeScriptil põhinev raamistik, mis sobib suuremahuliste ja keerukate veebirakenduste loomiseks, kuid oma keerukuse tõttu ei sobinud see antud projekti vajadustega. Vue.js on kerge ja paindlik JavaScripti raamistik, mille lihtne ülesehitus sobivad hästi ühe lehe rakenduste (SPA) loomiseks, kuid piiratud kogemuse tõttu otsustati siiski teise raamistiku kasuks [25].

React on Meta poolt loodud avatud lähtekoodiga JavaScripti teek, mida kasutatakse komponentidel põhinevate kasutajaliideste arendamiseks. Tänu oma kiirusele, paindlikkusele ja laiale ökosüsteemile on React üks populaarsemaid valikuid veebirakenduste loomiseks. Ühesuunaline andmevoog parandab andmehaldust ja rakenduse jõudlust, muutes veaotsingu protsessi kiiremaks [25].

Reacti populaarsust kinnitab ka tööjõuturg – ligikaudu 250 000 tööpakkumise analüüsi põhjal on see enim nõutud esiotsa raamistik, moodustades umbes 52% kõigist vastava valdkonna tööpakkumistest. Lisaks kuulub see kõrgemalt tasustatud tehnoloogiate hulka, mille keskmine aastapalk läheneb 113 000 dollarini. Prognooside kohaselt säilitab React ka 2024. aastal juhtpositsiooni, jäädes stabiilselt 50–60% vahemikku [26], [27].

Käesolevas projektis valiti React kasutajaliidese loomiseks, arvestades selle põhjalikku dokumentatsiooni, aktiivset kogukonda ja varasemat kogemust raamistikuga. React võimaldab komponentide korduvkasutust, tagades arenduse modulaarse ülesehituse ja lihtsustades koodi hooldamist.

Esirakenduse stiliseerimiseks kasutati React Bootstrapit, mis võimaldab kiiresti luua stiliseeritud kasutajaliideseid, kasutades valmis komponente, vorme ja menüüsid. See lihtsustab disaini loomist ja pakub suurepäraselt kohandatavust, et komponente saaks vajadusel muuta. Samuti tagab React Bootstrap, et kasutajaliides kohandub automaatselt erinevate ekraanisuurustega, pakkudes seeläbi paindlikkust ja kasutusmugavust erinevates seadmetes [28].

Graafikute loomiseks ja stiliseerimiseks kasutati Recharts-teeki, mis on mõeldud React rakenduste jaoks. See võimaldab luua interaktiivseid diagramme ja graafikuid, pakkudes laia valikut erinevaid diagrammitüüpe. Recharts'i eeliseks on selle lihtsus, paindlikkus ja suurepärase kohandatavus [29].

#### **4.2.4 Andmebaasi tehnoloogiad**

Andmebaasisüsteem pidi olema kiire, mugav ja võimekas. Lähtuvalt nendest nõuetest otsustati, et kõige sobivam lahendus on SQL-põhine andmebaasisüsteem.

Valikuks osutus PostgreSQL, objekt-relatsiooniline andmebaasisüsteem, mis pakub suurepäraselt jõudlust ja paindlikkust. PostgreSQL on tasuta ning sobib keerukate andmebaasimudelite elluviimiseks. Samuti tagab see andmete turvalisuse, terviklikkuse ja skaleeritavuse, võimaldades hallata suuri andmemahtusid, toetades indekseerimist ning olles tuntud oma stabiilsuse ja usaldusväärsuse poolest [30].

Andmebaasi muudatuste haldamiseks kasutati Liquibase'i, mis lihtsustab andmebaasi versioonihalduse ja muudatuste rakendamise automatiseerimist. Changelog XML- ja SQL-failide abil saab mitte ainult muudatusi rakendada, vaid ka neid vajadusel tagasi võtta. Liquibase võimaldab hallata kõiki andmebaasi struktuurimuutusi ühtse ja järjepideva meetodiga [31].

#### **4.2.5 Veebirakenduse arhitektuur**

Docker on tarkvaraplattform, mis võimaldab rakendusi käivitada isoleeritud keskkonnas, mida nimetatakse konteineriks. Konteinerid on kergekaalulised, kiiresti käivitatavad ja hallatavad. Isoleeritus võimaldab korraga käivitada mitu konteinerit, mis teeb võimalikuks näiteks andmebaasi, tagarakenduse ja esirakenduse eraldi, kuid koordineeritud toimimise [32].

Veebiserverina kasutati NGINX-i – avatud lähtekoodiga tarkvara, mis on tuntud oma suure jõudluse ja madala ressursikasutuse poolest. NGINX toimib puhverserverina, suunates kliendi päringud vastavatele serveritele. Seejuures muudetakse vajadusel HTTP-päiseid, et suurendada turvalisust või parandada süsteemi töökindlust. Samuti kasutatakse NGINX-i SSL/TLS protokolle, mis võimaldavad hallata krüpteeritud HTTPS-ühendusi [33].

## 5 Prototüübi arendamine

Järgnevas peatükis kirjeldatakse arendusprotsessi, kasutades eelnevalt valitud tehnoloogiaid. Samuti käsitletakse rakenduse arhitektuuri ning süsteemikomponentide omavahelist suhtust ja funktsioneerimist.

### 5.1 Arenduspõhimõtted

Prototüübi arendamisel kasutati agiilset arendusmetoodikat Kanban. See on üks DevOps-lähenedamise olulisi tööriistu, mis aitab suurendada töökorralduse tõhusust, toetades sujuvat ülesannete liikumist töövoos. Üheks peamiseks eeliseks on korraga töös olevate ülesannete arvu piiramine, mis aitab vältida ületootamist ja vähendab konfliktide tekkimise võimalust tööülesannete vahel – tänu sellele saavad asjad tehtud loogilises ja prioriteetidepõhises järjekorras. Antud töö raames oli Kanban eriti sobilik, kuna autor tegeles arendamisega üksi [34].

Ülesanded koostati funktsionaalsete nõuete alusel loogilises järjestuses. Iga ülesande jaoks loodi GitLabi keskkonnas eraldi *issue* ehk pilet, millele lisati selge kirjeldus – see võimaldas mõista, milline funktsionaalsus tuleb konkreetse ülesande raames ellu viia. Lisaks määrati igale piletile *label* ehk silt, mis aitas visuaalselt mõista, mida antud ülesanne sisaldab. Samuti määrati hinnanguline ajakulu ning reaalselt kulunud aeg.

Iga ülesande jaoks loodi eraldi *branch* ehk haru, et isoleerida muudatused põhikoodist (*main* harust). Selline lähenemine võimaldas arendustööd teha turvaliselt ja ilma, et see mõjutaks stabiilset koodibaasi.

Kui funktsionaalsus oli valmis, märgiti *issue* kirjelduses vastav ülesanne tehtuks. Seejärel esitati *merge request* (MR), mille käivitamisel toimusid automaatselt *pipeline*'id, mis kontrollisid koodi stiili, kompileeritavust ja testide läbimist. See aitas ennetada vigade sattumist põhikoodi. Kui kõik kontrollid läbisid edukalt, liideti muudatused *main* harusse ja pilet suleti automaatselt. Kogu projekti haldus kättesaadav siin: <https://gitlab.cs.taltech.ee/mabere/iaib>.



## 5.2 Andmebaas

Andmebaasis hoitakse kõiki süsteemi toimimiseks vajalikke andmeid – kasutajate andmeid, kategooriaid, loodud eelarveid koos sissemakssetega, kasutajatele määratud ligipääse ning rakenduse poolt genereeritud teavitusi. Vt Lisa 5, Joonis 12

Andmebaasimudeli loomisel arvestati objektidevaheliste seostega, kasutades välisvõtmeid (*foreign key*). Kõigil objektidel oli ID, mis suurenes automaatselt iga uue kirje lisamisel, see lihtsustas andmete otsingut ja seoste haldamist. ID-sid kasutati nii primaar- (*primary key*) kui ka välis välisvõtmetena.

Iga tabel loodi eraldi changelog-failina, et vähendada vigade tekkimise riski ja lihtsustada edasiste muudatuste haldamist. Changelog'id olid SQL-põhised ning neid hallati master-faili abil (*changelog-master*). Master-fail lisab automaatselt kõik muudatused, mis on määratletud changelog-kataloogi failides, võimaldades andmebaasi struktuuri järjepidevat ja kontrollitud arendamist.

- **User** – Hoiab kasutajate andmeid. Igal kasutajal on unikaalne ID ja e-posti aadress. Lisaks sisaldab tabel kasutaja nime ja parooli, mis hoitakse räsitud kujul.
- **Category** – Hoiab kõiki salvestatud kategooriaid, sealhulgas nii autorite loodud kui ka vaikimisi kategooriaid. Igal kategoorial on unikaalne ID, nimi ja tüüp, mis võib olla kas “kulu” (*expense*) või “sääst” (*saving*). Väljendid nagu emotikon ja värv on valikulised, samuti autor ID, kuna vaikimisi kategooriatel autor puudub.
- **Budget** – Tabel sisaldab eelarvete andmeid. Igal eelarvel on unikaalne ID, nimi, tüüp (mis võib olla kas "kulu" ehk *expense* või "sääst" ehk *saving*), hinnanguline summa, mis määrab eelarve limiidi, ja algus- ning lõppkuupäevad, määrates eelarve kehtivuse perioodi. Kirjeldus on valikuline väli, mis võib sisaldada täiendavat teavet eelarve kohta.

Iga eelarve kuulub kindlale kasutajale ja kategooriale. Seos kasutaja ja eelarve vahel on üks-mitmele: üks kasutaja võib omada mitut eelarvet. Samuti on seos kategooria ja eelarve vahel üks-mitmele: üks kategooria võib sisaldada mitut eelarvet. Vt Lisa 5, Joonis 13

- **Transaction** – Hoiab tehingute ehk sissemakssete andmeid. Igal tehingul on unikaalne ID, summa ja valikuline kirjeldus. Samuti määratakse igale tehingule automaatselt

salvestamisaeg, mis sisaldab kuupäeva ja kellaaega UTC formaadis.

Iga tehing kuulub kindlale kasutajale ja eelarvele. Seos tehingu ja kasutaja vahel on üks-mitmele: ühele kasutajale võib kuuluda mitu tehingut. Samuti on seos tehingu ja eelarve vahel üks-mitmele: üks eelarve võib sisaldada mitut tehingut. Vt Lisa 5, Joonis 14

- **User Budget Access** – Tabel hoiab infot loodud ligipääsude kohta, kus üks kasutaja on andnud teisele kasutajale õiguse pääseda ligi kindlale eelarvele. Igal kirjel on unikaalne ID ning roll, mis määrab kasutaja õigused: vaataja (*viewer*) saab eelarvet ainult vaadata, redigeerija (*editor*) saab lisada ja eemaldada tehinguid ning hallata ka teisi kasutajaid.

Iga ligipääs on seotud konkreetse eelarve ja kasutajaga. Seos eelarvega on üks-mitmele: ühele eelarvele saab määrata mitu ligipääsu. Seos kasutajaga on samuti üks-mitmele: üks kasutaja võib omada ligipääsu mitmele eelarvele. Vt Lisa 5, Joonis 15

- **Notification** – Hoiab andmeid kõigi rakendusesiseste teavituste kohta. Igal teavitusel on unikaalne ID ja kohustuslik tüüp, mis kirjeldab, millise sündmusega on tegemist — näiteks teavitus sissemakse tegemise kohta. Teavituse loomisel määratakse sellele automaatselt salvestamise aeg (kuupäev ja kellaaeg UTC formaadis) ning vaikimisi on teavituse olek „lugemata“ (*is\_read = false*). Olenevalt teavituse tüübist võib see sisaldada lisainfot, näiteks summat või seotud kasutajat.

Iga teavitus on seotud kolme osapoolega: saaja (*recipient*) – kasutaja, kellele teavitus on suunatud. Seos on üks-mitmele: üks kasutaja võib olla mitme teavituse saajaks. Toimingu tegija (*action\_user*) – kasutaja, kes põhjustas teavituse (nt lisas tehingu). Seos on samuti üks-mitmele: üks kasutaja võib olla mitme teavituse põhjustaja. Eelarve, mille kohta käis teavitus. Seos on üks-mitmele: üks eelarve võib olla seotud mitme teavitusega. Vt Lisa 5, Joonis 16

### 5.3 Tagarakendus

Projekti ehitamiseks kasutati Gradle'it, mis on kaasaegne tööriist ehitamise ja automatiseerimise jaoks. See võimaldab hallata kogu projekti elutsüklit, sealhulgas kompileerimist, testimist ja sõltuvuste haldamist. Antud töö raames eelistati Gradle'it Mavenile, kuna Gradle'i ehituse vahemälu salvestab eelmiste ehituste tulemused ning Gradle'i deemon

töötab taustal, hoides ehitusprotsessi aktiivsena, mis kiirendab järgnevate ehituste täitmist.

Rakendus on üles ehitatud funktsioonipõhise paketi struktuuri järgi, kus tarkvarakomponendid on rühmitatud vastavalt ärifunktsioonidele või omadustele. Iga funktsioon asub omaette pakettis, mis sisaldab kõiki selle funktsiooni täitmiseks vajalikke komponente, nagu *controller*, *service*, *repository*, *DTO*, *mapper* ja *entity*. Selle lähenemisviisi valiti kõrge sidususe ja modulaarsuse tõttu. Kõik konkreetse funktsiooniga seotud komponendid paiknevad ühes pakettis, mis parandab kapseldatust ning muudab koodi lihtsamini hallatavaks. Lisaks aitab see struktuur hoida igas pakettis vaid konkreetse ärifunktsiooni jaoks vajalikke klasse, toetades loogilist eristamist ja soodustades süsteemi hooldatavust. API dokumentatsioon on kättesaadav lokaalsel käivitamisel <http://localhost:8080/swagger-ui/index.html>

Arenduses rakendati *controller-service-repository* mustrit. Selles struktuuris vastutab *controller* kasutajaliidese ja tagarakenduse vahelise suhtluse eest, kasutades HTTP meetodeid nagu GET, POST, DELETE, PUT jms, mis on iseloomulikud RESTful API-dele. *Controller* suunab kasutaja päringud *service*-kihile, mis tegeleb rakenduse äri loogikaga — töötleb andmeid ja koordineerib suhtlust *repository*-kihiga. *Repository* kiht omakorda vastutab andmete pärimise ja salvestamise eest andmebaasis.

Kuna rakenduse andmekihina kasutati PostgreSQL andmebaasi, võeti kasutusele ka JPA (*Java Persistence API*). JPA võimaldab Java objektide salvestamist, pärimist, uuendamist ja kustutamist andmebaasis ilma SQL-kirjutamiseta. Iga andmeobjekti haldamiseks loodi *repository*-klass, mis laiendab JPA vastavat liidest. See lihtsustab andmebaasi päringute tegemist, andes sisendiks vastava *entity* klassi.

Iga *entity* teisendati vastavaks DTO-ks (*Data Transfer Object*), mida kasutati andmete edastamiseks kasutajale või nende vastuvõtmiseks kasutajalt. DTO võimaldab täpselt määrata, milliseid andmeid väljastatakse või vastu võetakse, tagades andmete turvalisuse ja vältides liigsete andmete kuvamist kasutajaliidese.

*Entity* ja DTO vahelise teisenduse lihtsustamiseks kasutati MapStruct tööriista koos Mapper-klassiga. MapStruct genereerib vajaliku teisenduskoodi automaatselt kompileerimise ajal. Kui *entity* ja DTO väljade nimed kattuvad, seob MapStruct need automaatselt. Erinevate nimede või väljade puhul tuleb kasutada *@Mapping* annotatsiooni.

Koodi loetavuse ja lihtsustamise eesmärgil kasutati Lomboki teeki, mis aitab vältida Java koodis korduvate osade, nagu *getter* ja *setter* kirjutamise ja teiste sarnaste ülesannete kordamist. Lombok võimaldab automaatselt genereerida vajalikke meetodeid ja konstruktoreid, muutes koodi puhtamaks ja lihtsamini hallatavaks.

Süsteemi töö jälgitavuse ja vigade kiireks tuvastamiseks kasutati logimist. Logimise kaudu saab jälgida rakenduse käitumist ning tuvastada võimalikke probleeme. Logimiseks kasutati Lomboki *@Slf4j* annotatsiooni, mis loob automaatselt logimisobjekti, võimaldades lihtsasti logida olulisi sündmusi ja vigu.

Lisaks JPA *repository*le kasutati eelarvete otsinguks kohandatud *Criteria repository*'t, mis võimaldab koostada dünaamilisi päringuid vastavalt kasutaja määratud filtritele. See välistab vajaduse luua mitu eraldi JPA meetodit ning võimaldab paindlikult otsingukriteeriume lisada või eemaldada.

Vigade haldamiseks kasutati kohandatud erandite käsitlejat (*custom exception handler*), mis laiendab *RuntimeException*-i ja määrab sellele vastava sõnumi. See lähenemine võimaldab luua spetsiifilisi erandeid, mis on kohandatud rakenduse konkreetsete vigade jaoks. Erandite käsitleja tagab, et vigade korral edastatakse kasutajaliidesele selged ja arusaadavad veateated, muutes vigade mõistmise kasutajate jaoks lihtsamaks.

## 5.4 Esirakendus

Esirakenduse arendamisel kasutati Reacti raamistikku. Tagarakendusega suhtlemiseks rakendati HTTP-päringute tegemiseks JavaScripti teeki Axios. Korduva koodi vältimiseks loodi iga päringu tüübi (GET, POST, DELETE) jaoks abistavad funktsioonid, kuhu sai sisendina anda URL-i ja vajadusel ka andmeid ning kasutaja tuvastust tokeni validatsiooniga. See parandas koodi loetavust ning aitas vältida korduvaid koodiridu.

Keelevahetuse realiseerimiseks kasutati i18n-lahendust, mis võimaldas keelt mugavalt muuta tõlkefailide süsteemi abil. *Public*-kaustas paiknesid kaks tõlkefaili: üks eesti ja teine inglise keele jaoks. Failid sisaldasid võti-väärtus paare, kus võti viitas konkreetsele tekstile või sõnale ja väärtus on vastav tõlge valitud keeles. Keelevahetus toimub rakenduse päises asuva nupu kaudu. Kui kasutaja valib teise keele, uuendatakse kogu rakenduse tekst vastavalt.

Lisaks staatilistele tõlgetele toetab i18n ka dünaamiliste väärtustega tekstide tõlkimist. Näiteks on võimalik määratleda tõlkefailis lause, kuhu sisestatakse jooksvalt muutuvaid andmeid. See on eriti kasulik teavituste puhul, mis sisaldavad andmeid nagu teavituse tüüp (nt „kasutaja lisatud“), seotud e-posti aadress, muutja nimi ja muudatuse aeg. Tõlkefailis kirjeldatakse, kuidas selline sõnum tuleb tõlkida, ning dünaamilised väärtused lisatakse automaatselt õigesse kohta.

Reaalajas teavituste edastamiseks kasutati STOMP-protokolli (*Simple Text Oriented Messaging Protocol*). Kui kasutaja on sisse logitud ja saab uue teavituse, kuvatakse see reaalajas ekraani paremas servas. Lisaks kasutati SockJS-i, mis toimib WebSocketi laiendusena.

Kõiki saadud teavitusi on võimalik vaadata teavituste lehelt, millele pääseb rakenduse pääsest. Kõik teavitused saab kustutada ühe nupuvajutusega, millega märgitakse need loetuks (teavituse väli “*read*” väärtustatakse tõeseks). Rakenduses olevate diagrammide kuvamiseks kasutati Recharts-teeki ning selle sisseehitatud komponente *BarChart*, *PieChart* ja *LineChart*, eelnevalt töödeldes tagarakendusest saadud andmeid sobivale kujule.

Stiililahendustes kasutatakse Bootstrapi sisseehitatud komponente ning klasside abil kohandatakse kujundust vastavalt ekraanisuurusele.

Kategooriate loomisel emotikonide kuvamiseks kasutati Reacti teeki *EmojiPicker*. Värvivalikut realiseeriti *ColorPicker*-i abil, millega on võimalik sisestada värve HEX- ja RGB-vormingus ning valida värvi paleti pealt.

## 5.5 Turvalisus

Registreerimiseks peab kasutaja määrama salasõna, mis vastab järgmistele nõuetele: peab olema vähemalt 8 tähemärki pikk ning sisaldama vähemalt ühte tähte ja numbrit. Registreerimiseks peab kasutaja sisestama parooli kaks korda — kui mõlemad sisestatud paroolid ühtivad, kasutatakse parooli turvaliseks salvestamiseks *BCryptPasswordEncoder*-it, mis põhineb BCrypt-algoritmil.

BCrypt on spetsiaalselt loodud paroolide turvaliseks räsiarvutuseks — see lisab igale paroolile unikaalse soola ning kasutab mitmeid iteratsioone, et raskendada võimalikke rünnakuid. Tulemuseks on ühesuunaline räsi, mida ei ole võimalik tagasi teisendada.

Sisselogimisel tehakse andmebaasi päring kasutajaandmete leidmiseks sisestatud e-posti aadressi alusel. Kui kasutaja leitakse, võrreldakse sisestatud parooli väärtust andmebaasis talletatud räsiga. Kuna BCrypt salvestab soola räsi sees, saab süsteem kasutada sama soola uuesti, et räsi arvutada ka sisestatud paroolile. Kui uus räsi ühtib andmebaasis salvestatuga, loetakse parool õigeks ja sisselogimine õnnestub.

Sisselogimisel genereeritakse kasutajale unikaalne JWT, mida kasutatakse autentimiseks. Token sisaldab kasutaja andmeid ning sessiooni kehtivusaega. See luuakse kasutades HMAC-SHA256 algoritmiga allkirjastatud võtit, mis genereeritakse iga kord juhuslikult. Tokeni kehtivusajaks on määratud 30 minutit.

Pärast genereerimist salvestatakse JWT turvaliselt küpsisesse, mis on tähistatud kui *HttpOnly* — see tähendab, et JavaScript ei pääse sellele ligi, mis omakorda suurendab turvalisust. Ka küpsise kehtivusaeg on 30 minutit. Kuna unikaalse JWT kehtivusaeg on piiratud, kontrollib rakendus selle kehtivust iga 10 sekundi järel ning iga päringu tegemisel. Esirakendus saadab päringu tagarakendusele, et saada sisselogitud kasutaja JWT, mis on salvestatud *HttpOnly* küpsises ja pole JavaScriptiga otse ligipääsetav.

Kui token saadakse kätte, dekodeeritakse see ning kontrollitakse, kas kehtivusaeg on möödunud. Kui token on aegunud, logitakse kasutaja automaatselt välja ning kuvatakse hüpikaken, mis teavitab sessiooni lõppemisest. Kui token on veel kehtiv, jätkub päring tavapäraselt.

Rakendus piirab juurdepääsu erinevatele lehtedele, kui kasutaja ei ole sisselogitud. Sellisel juhul on ligipääsetavad ainult sisselogimis- ja registreerimisleht ning keeleveliku võimalus. Kui kasutaja sisestab aadressiribale mitteeksisteeriva lehe, suunatakse automaatselt sisselogimislehele.

Pärast edukat sisselogimist suunatakse kasutaja automaatselt avalehele, kust on ligipääs kõikidele funktsioonidele, välja arvatud sisselogimis- ja registreerimisleht, mis muutuvad kättesaamatuks. Kui sisselõgitud kasutaja sisestab otsingusse olematu lehe aadressi, suunatakse ta tagasi avalehele

## 5.6 Automaattestimine

Funktsionaalsuse ning süsteemi korrektsuse kontrollimiseks kirjutati nii *mock*- kui ka integratsiooniteste. Kokku loodi 123 testi, mis katavad 88% klassidest, 94% meetoditest ja 88% koodiridadest.

*Mock*-testimise eesmärk oli asendada tegelikud komponendid või teenused simuleeritud ehk *mock*-objektidega. Neid kasutatakse selleks, et testida ärioloogikat isoleeritult. Testide käigus kontrolliti meetodite väljakutsumist ning nende korrektsust, ilma et oleks ühendatud andmebaasiga.

Lisaks viidi läbi ka integratsioonitestid, mille käigus kontrolliti, kuidas erinevad süsteemi osad – näiteks kontrollid, teenuskiht ja andmebaas koos toimivad. Testide käigus tehti päringuid erinevatele *endpoint*idele, võrreldi saadud staatuskoode ning vastuste sisu, sh nime, ID-d ja muid elemente.

## 5.7 Serverisse ülespanek

Rakendus paigaldati TalTech'i serverisse. Esmalt määrati sellele domeeninimi, milleks sai „SendiSemu“. Pärast domeeni kinnitamist paigaldati serverisse NGINX ning selle konfiguratsioon kohandati vastavalt rakenduse vajadustele.

Seejärel seadistati HTTPS, et tagada krüpteeritud ja turvaline ühendus. Selleks kasutati tasuta SSL-sertifikaati Let's Encrypti kaudu.

Kui HTTPS oli edukalt seadistatud ja server turvaliselt ühendamiseks valmis, alustati rakenduse paigaldamise ja käivitamisega.

Serverisse ülespanekuks loodi *DockerCompose*'i fail, mis käivitab konteinerid andmebaasi, tagarakenduse ja esirakenduse jaoks.

Java tagarakenduse koostamiseks kasutatakse vastavat *Dockerfile*'i, kus on määratud OpenJDK 21 pilt ja Gradle'i paigaldus. Lähtekood kompileeritakse JAR-failiks, mis seejärel konteineris käivitatakse.

Esirakendus, mis on loodud Reacti SPA-na, ehitatakse Node.js abil. Valmisrakendus

kopeeritakse NGINXi kausta, et seda saaks serveerida läbi veebiserveri porti 80 kaudu.

Andmebaasiteenus põhineb PostgreSQL 17 pildil (*Docker image*). Andmebaasi kasutajanimi ja parool saadakse .env failist. Andmed salvestatakse kohalikku kausta ja konteiner töötab porti 5432 kaudu.

Konteinerid käivitatakse kindlas järjekorras: esmalt andmebaas, kuna teised teenused sõltuvad sellest. Seejärel tagarakendus, mis kasutab andmebaasi ja töötab portil 8080. Viimasena käivitatakse esirakendus, mis suunab liikluse lokaalselt portilt 3000 NGINXi kaudu edasi porti 80.

Kõik konteinerid on ühendatud ühtsesse võrku, mis võimaldab neil omavahel suhelda. Igal konteineril on määratud ka oma andmemaht (*volume*), mis tagab andmete säilimise ka pärast konteineri sulgemist või uuesti käivitamist.



## 6 Tulemused ja valideerimine

Järgnevas peatükis keskendutakse projekti tulemustele ning analüüsile. Viiakse läbi kasutajatestimine, mille põhjal hinnatakse süsteemi funktsionaalsust ning analüüsitakse, valminud projekti seatud nõuetele vastavust. Kasutajatestimisest võttis osa 15 inimest.

### 6.1 Kasutajatestimine 1. voor ja selle analüüs

Kasutajatestimine viidi läbi küsimustiku vormis, mis hõlmas nii kasutajaliidese (UI) kui ka kasutuskogemuse (UX) aspekte. Küsimustik oli jaotatud osadeks vastavalt rakenduse erinevatele funktsionaalsustele (vt Lisa 6).

Kulukategooriate loomise lihtsust hinnati viiepallisel skaalal, kus 1 tähistas "väga rasket" ja 5 "väga lihtsat". Kõik vastajad hindasid seda positiivselt: 66% pidas protsessi väga lihtsaks ja 33% lihtsaks (vt Lisa 7, Joonis 20), mis viitab sellele, et kulukategooriate loomine on kasutajatele intuiitivne ja arusaadav.

Seda kinnitas ka tagasiside küsimusele: "Kas sul tekkis segadust või mõni hetk, kus ei teadnud, mida edasi teha?" Avatud vastustes kordusid märksõnad nagu "intuiitivne", "selge", "arusaadav" ja "loogiline", mis näitab, et kasutajad ei kogenud segadust ega ebakindlust kategooria loomise protsessis. Lisaks hinnati vaikimisi kulukategooriate sobivust – 100% vastanutest nõustus väitega, et need on arusaadavad ja täpsed (vt Lisa 7, Joonis 21).

Järgnevalt paluti vastajatel luua konkreetne eelarve ja teha sinna sissemakse. Lisa 7, Joonis 22 põhjal selgub, et 100% vastanutest leidis, et eelarve loomise protsess oli arusaadav. Sissemakse tegemist hindas 93,3% vastanutest väga lihtsaks ning 6,7% (üks vastaja) lihtsalt lihtsaks. (vt Lisa 7, Joonis 23)

Avatud vastustes toodi esile, et meeldivaks peeti eelarve personaliseerimise võimalusi, nagu kuupäevade määramine ja kirjelduse lisamine. Samuti tõsteti esile eelarve täitumise visuaalset esitust, mis muudab kulutatud ja allesjäänud summa selgelt eristatavaks, aidates

paremini jälgida rahakasutust ja teha teadlikke otsuseid.

Lisa 7, Joonis 24 kohaselt, säästmiseesmärkide loomise lihtsust hinnati samuti 5-palli skaalal, kus 1 tähistas väga rasket ja 5 väga lihtsat. Tulemustest selgus, et enamus kasutajaid (80%) pidas eesmärkide seadmist väga lihtsaks. Siiski andsid kolm vastajat (kokku 20%) madalama hinnangu – vastavalt 2, 3 ja 4, mis viitab sellele, et üksikute kasutajate jaoks oli protsess kohati segane. Seda kinnitab ka tagasiside, mille kohaselt pärast eesmärgi loomist oli raske seda üles leida, kuna kasutaja suunati vale lehe peale. Selle probleemi lahendust käsitletakse järgmises peatükis.

Küsimusele säästmise visuaalse arusaadavuse kohta vastas 93,3% jaatavalt (vt Lisa 7, Joonis 25). Avatud vastustes mainiti, et rakenduse visuaalne esitus aitab eesmärkide täitmisest paremini aru saada ning mõjub säästmisele motiveerivalt.

Analüüsidest vastuseid selgub, et protsentuaalselt täituv progressiriba toetab eelarve paremat planeerimist ja jälgimist. Nagu üks kasutaja märkis: “Täpse jälgimiseta ei saada tihtipeale aru, kuhu ja kui palju raha päriselt kulub. Rakendus aitab järge pidada.” Teine lisas: “Mulle meeldib, et saab vaadata diagrammi vormis enda progressi – see aitab raha säästa, kuna saan visuaalse ülevaate hetkeseisust ja planeerin oma tegevusi paremini.”

Järgnevalt paluti jagada loodud eelarvet teise isikuga, kasutades selleks etteantud kasutajatunnust. Eelarve jagamise võimalust hindasid positiivselt kõik vastajad (Lisa 7, Joonis 26). Peamise eelisenähtena toodi välja võimalust pidada ühist eelarvet pereliikmetega.

Lisa 7, Joonis 27 põhjal ilmnes aga, et koos panustamise funktsioon ei olnud kõigile täielikult arusaadav. Avatud vastustes tehti ettepanek eristada selgemalt ühiseid eelarveid, mis muudaks jagatud eelarve jälgimise lihtsamaks ja visuaalselt arusaadavamaks. Selle teema käsitus jätkub järgmises peatükis.

Lisa 7, Joonis 28 illustreerib kasutajate suhtumist teavitustesse. 86,7% vastanutest peab neid väga kasulikeks. 100% nõustub, et märguanded mängivad olulist rolli eelarve planeerimisel ja aitavad kulutusi paremini kontrollida (vt Lisa 7, Joonis 29). See kinnitab, et teavitused ja märguanded toetavad kasutajaid eelarvest kinnipidamisel, muutes rahakasutuse jälgimise lihtsamaks ja teadlikumaks.

Lisa 7, Joonis 30 näitab, et 86,7% vastanutest hindas graafikuid loetavateks ja arusaadavateks. Kasutajate lemmikuks osutus sektordiagramm, mis kujutab eelarve jaotust kategooriate lõikes – see võimaldab mõista, kuhu suurem osa rahast kulub, mille põhjal on lihtne teha järeldusi järgmise kuu eelarve koostamisel.

Lisaks toodi esile tulpdiagramm, mis aitab jälgida kulutuste ja säästmise trende eri ajaperioodidel ning annab väärtuslikku teavet finantskäitumise kohta ajas.

Kõik vastajad nõustusid, et graafikud aitavad rahalist olukorda paremini mõista, kuna visuaalne esitus teeb info mõistmise lihtsamaks ja arusaadavamaks võrreldes numrilise kujutusega.

## 6.2 Parandused

Põhjalikult analüüsid kasutajate tagasisidet rakendusele, toodi selgelt esile korduvaid tähelepanekuid, mida saab jagada kolmeks kategooriaks: bugid ehk vead, UI ehk kasutajaliidese parandused ning uued nõutud funktsionaalsused, mis parandaksid kasutajakogemust kordades.

Bugide hulka kuulus näiteks valele lehele suunamine pärast säästueesmärgi loomist, mis sai kiiresti parandatud. Samuti toodi välja, et teavitus säästueesmärgi täitmisel sisaldas sõnastust: „on jäänud säästa 0 €”. See probleem ilmnis seetõttu, et lisaks õnnitlusele eesmärgi täitmise puhul kuvatakse ka julgustavaid sõnumeid, mis ilmuvad siis, kui säästueesmärk on täidetud rohkem kui 90% ulatuses, sisaldades veel allesjäänud summat. Ka see probleem sai kiiresti lahendatud.

Paljud kasutajatestides osalenud mainisid, et eelarve loomise nupp oli raskesti leitav. Probleem lahendati, tuues nupp esile eristuva värviga ning muutes selle suuremaks. Samuti toodi välja, et olemasolevate eelarvete kuvamise linnuke oli liiga väike ja jäi kergesti märkamata. Selle asemel loodi suurem nupp, mis muutub täidetuks, kui eelarved on nähtaval, ning omab valget tausta, kui need pole kuvatud.

Säästude puhul märgiti, et loogilisem oleks kasutada fraasi „jäänud koguda“ senise „alles“ asemel vaba summa tähistamisel. See muudatus parandas säästueesmärkide mõistetavust ja kasutajakogemust.

Kasutajatestimise käigus küsiti ettepanekuid ning mõtteid, kuidas rahendust paremaks muuta. Enim mainitud pakkumiste hulka kuulusid: jagatud eelarvete eraldi kuvamine, korduvate eelarvete loomise võimalus ning hüpinkaken tehingute kustutamisel.

Varasem loogika nägi ette, et kõiki kasutaja loodud eelarveid kuvatakse isiklike eelarvete all — olenemata sellest, kas need on kellegagi jagatud või mitte. Kasutajatestimise käigus selgus, et enamus vastanutest pidasid seda ebaloogiliseks. Seetõttu muudeti funktsionaalsust, et jagatud eelarved oleks eraldi nähtavad.

Mitmed kasutajad tõid välja probleemi, kus nad olid kogemata kustutanud tehinguid eelarve alt. Selle vältimiseks loodi hüpinkaken, mis palub kasutajal veenduda enne sissemakse kustutamist, kuvades tehingu andmeid.

Samuti toodi välja, et rakendust muudaks mugavamaks võimalus luua korduvaid eelarveid — eelarve loomisel saab nüüd määrata, mitme järjestikuse kuu jooksul soovitakse seda eelarvet korrata.

### **6.3 Kasutajatestimine 2. voor ja selle analüüs**

Pärast esimese kasutajatestimise vooru tagasiside põhjal tehtud parandusi, viidi läbi teine testimisvoor, kus keskenduti tehtud muudatuste toimivusele. Vt Lisa 8

Kuna eelnevalt oli välja toodud, et nupud mis omavad tähtsast funktsionaalsust nagu eelarvete loomine ja kuvamine ei olnud silmapaistvat, küsiti seda kasutajate käest, millele kõik vastasid jaatavalt (vt Lisa 9, Joonis 33).

Suurim toimunud loogika muutus seisnes ühiste eelarvete kuvamises. Küsides kuskohast eeldab kasutaja leida eelarveid, mida ta on jaganud kellegagi teisega, vastas 93.3% ehk märkimisväärne enamus, “ühiste” alt. See kinnitab tehtud muudatuse tähtsust. Vt Lisa 9, Joonis 34

Lisa 9, Joonis 35 kinnitab hüpinkakna vajalikkust kustutamise puhul, kus 100% vastanutest olid nõus, et see aitab olla ettevaatlikum. Põhjendusena toodi esile, et see välistab kogemata kustutamise võimaluse, võimaldab olukorda paremini kontrollida ning annab võimaluse toiming enne lõplikku kinnitamist üle vaadata.

Küsimusele eelarvete korduva loomise kasulikkuse kohta vastas 80% kasutajatest, et see on väga kasulik, mis viitab sellele, et enamikul kasutajatest on vajadus luua sarnase summaga eelarveid iga kuu. See teeb korduvate eelarvete funktsiooni väärtuslikuks. Samas jäid ülejäänud 13,2% uue funktsionaalsuse suhtes ükskõikseks. See võib tuleneda sellest, et mõnede kasutajate finantsolukord on igakuiselt varieeruv ning ei nõua samasuguste eelarvete loomist. Vaatamata sellele, sai korduvate eelarvete loomise funktsionaalsus kinnitatud. Vt Lisa 9, Joonis 36

Seejärel uuriti, kas kasutajad sooviksid tulevikus suuremat kontrolli eelarvete korduvuse üle, arvestades, et praegu saab neid luua ainult kuu algusest kuu lõpuni. Võimalikud täiendused võiksid lubada kordusi ka näiteks nädalate või mitmekuuliste perioodide kaupa. 66% vastanutest ütles, et nad ei soovi rohkem kontrolli. Kuid avatud vastustes ei osanud keegi täpsustada, millist lisafunktsionaalsust läheks veel aja, märkides, et olemasolevast lahendusest piisab. Vt Lisa 9, Joonis 37

Kõik vastajad nõustusid, et tehtud muudatused parandasid kasutajakogemust (vt Lisa 9, Joonis 38). Lisa 9, Joonis 39 pealt selgub, et 86,7% vastanutest hindas kasutajakogemust maksimaalseks, samas kui ülejäänud 13,3% hindasid seda heaks.

Kuigi alati on võimalik arendada uusi funktsionaalsusi ja teha täiendusi, ei esitanud ükski vastaja konkreetseid ettepanekuid võimalike muudatuste osas.

Autor loeb kasutajatestimist õnnestunuks kuna üldine rahulolu jäi märkimisväärselt üle keskmise, Kommentaaridest on selge et vastajatele meeldis antud rakendus ning saaks leida kasutust igapäevaelus.

## **6.4 Nõuetele vastavus**

Lõputöö raames valmiv rakenduse prorotüüp omab järgmisi funktsionaalsusi. Funktsionaalse nõude punkti "Kasutajakonto ja autentimine" kohaselt on kasutajal võimalik luua konto sisestades oma nime, e-posti aadressi ja parooli. Pärast konto loomist saab kasutaja siseneda süsteemi, kasutades e-posti ja salasõna. Samuti on olemas võimalus kontolt välja logida. Lisaks logitakse kasutaja turvapõhjustel automaatselt välja pärast teatud aja möödumist ning kuvatakse hüpikaken, mis teavitab sessiooni lõppemisest. Nimetatud funktsionaalsused on illustreeritud Lisas 10, Joonistel 40-42

Lisas 10, Joonistel 43 - 45 on kujutatud eelarvete otsing vastvalt funktsionaalse nõude punktile "Filtrid ja otsing". Kasutajal on võimalik otsida olemasolevaid eelarveid kategooria ja kuupäevavahemiku alusel. Kategooria saab valida kas vaikimisi olemasolevate hulgast või kasutaja enda loodud kategooriate seast. Eelarvete filtreerimiseks saab määrata ka algus- ja lõppkuupäeva, valides need kalendrist. Mõlemad kuupäevad on kohustuslikud ning alguskuupäev peab olema varasem kui lõppkuupäev. Lisaks on kasutajale kättesaadavad kiirvalikunupud, nagu „Käesolev kuu“, „Eelmine kuu“, „Käesolev aasta“ ja „Eelmine aasta“, mis seadistavad kuupäevi automaatselt.

Eelarve loomisel on kasutajale kättesaadavad kategooriavalikud (funktsionaalne nõue "Kategooriate haldamine"). Valikud põhinevad vaikimisi kategooriatel vastavalt valitud eelarve tüübile ning kasutaja enda varem loodud kategooriatel. Lisaks on võimalik luua uusi. Uue kategooria loomine nõuab ainult nime sisestamist. Soovi korral saab aga kasutaja valida sümboli (*emoji*) pakutud valikust ning määrata värvi etteantud paletist. Lisa 10, Joonis 46-49

Funktsionaalne nõue "Eelarvete haldamine" põhiselt, on eelarve koostamisel on kohustuslik määrata eelarve tüüp (tüüp määratakse vaikimisi automaatselt, kuid kasutajaliides võimaldab seda vajadusel ka käsitsi muuta), kategooria ning planeeritav summa. Lisaks on võimalik soovi korral lisada kirjeldus.

Kuupäevi saab valida kuupõhiselt: vaikimisi määratakse alguskuupäevaks jooksva kuu esimene päev ja lõppkuupäevaks sama kuu viimane päev. Soovi korral on kasutajal võimalik määrata ka täpsemad kuupäevad, valides need kalendrist käsitsi.

Kasutajatestimise tagasiside põhjal lisati eelarve loomise protsessile täiendav funktsionaalsus – kuupõhise valiku korral saab määrata eelarve korduste arvu. See tähendab, et sama eelarve kopeeritakse automaatselt järjestikuste kuude lõikes vastavalt kasutaja määratud korduste arvule. Lisa 10, Joonis 50

Pärast kulueelarve loomist suunatakse kasutaja edasi lehele, mis vastab loodud eelarve tüübile. Vastavalt funktsionaalsele nõudele "Kulude jälgimine", on kulueelarvesse võimalik teha sissemakseid klikkides selle peale, Pärast sissemakse tegemist kuvab kasutajaliides eelarve täitumise seisu: näha on nii olemasolev summa kui ka vaba raha.

Eelarvekaardil kuvatakse kategooria nimi, sellele määratud värv ning ikoon (*emoji*). Lisaks on nähtavad ka eelarve kirjeldus ja kehtivusperioodi kuupäevad. Kui eelarve on täielikult täidetud, kuvatakse vastav summa. Vt Lisa 10, Joonis 51 - 53

Funktsionaalne nõue "Säästmise eesmärgid" ütleb, et säästueesmärgi loomisel suunatakse kasutaja automaatselt säästude lehele. Iga säästueelarve puhul on võimalik teha sissemakseid. Pärast sissemakse tegemist kuvatakse eesmärgi täitumine, koos säästetud summa ning veel koguda vajamineva summaga. Kui säästus ületavad planeeritud eesmärgi, kuvatakse ka ülejääk. Lisa 10, Joonis 54 - 55

Tehingu sisestamisel eelarvesse määratakse summa ning soovi korral ka kirjelduse. Vastavalt funktsionaalsele nõudele "Sissemaksed eelarvetesse". Sissemakse kuupäev määratakse automaatselt. Tehtud sissemakseid on võimalik ka kustutada. Selleks tuleb avada soovitud eelarve, mille järel kuvatakse nimekiri tehtud tehingutest. Konkreetse sissemakse kustutamiseks tuleb vajutada ristikujulisele nupule tehingu juures.

Kasutajatestimise käigus tehti ettepanek lisada tehingu kustutamisel hüpinkaken, mis võimaldab kasutajal oma kustutamissoovi kinnitada. See aitab vältida kogemata kustutamisi ja parandab kasutajakogemust. Vt Lisa 10, Joonised 56 - 57

Pärast eelarve loomist saab kasutaja seda teistega jagada (funktsionaalne nõue "Eelarvete jagamine"). Jagamiseks tuleb vajutada eelarve paremas ülanurgas asuvale ikoonile, mille tähis kuvab, kas eelarve on juba kellegagi jagatud või mitte. Uue kasutaja lisamiseks tuleb sisestada tema e-posti aadress ning määrata sobiv roll. Süsteem kontrollib automaatselt, kas sisestatud e-posti aadressiga kasutaja eksisteerib.

Saadaolevad rollid on vaataja, kellel on õigus ainult eelarve täitumist jälgida, ning redigeerija, kes saab lisada ja eemaldada nii sissemakseid kui ka teisi kasutajaid. Kui vaataja rollis kasutaja üritab sooritada toimingut, milleks tal õigusi pole, kuvatakse talle vastav veateade. Lisa 10, Joonis 58 - 62

Lisa 10, Joonis 63 – avalehelt kättesaadavad graafikud (funktsionaalne nõue "Aruanded ja analüüs") annavad ülevaate kuludest ja säästudest. Sektordiagramm näitab nende protsentuaalset jaotust kategooriate lõikes valitud perioodil. Tulpdiagramm kujutab trende, võimaldades võrrelda perioodi varasematega.

Lisaks on olemas joondiagramm, mis on nähtav iga konkreetse eelarve detailvaates tehingute alt. See kuvab sissemaksete dünaamikat ajas, vastavalt kasutaja õigustele (autor või redigeerija). Vt Lisa 10, Joonis 64

Funktsionaalse nõude "Reaalajas teavitused" vastavalt saadab rakendus automaatseid teavitusi mitmete sündmuste korral. Kui kulueelarve täitumine jõuab 90%ni, kuvatakse vastav teade koos teabega, kui suur summa on veel võimalik kulutada. Sarnane teavitus kehtib ka säästueesmärkide puhul, kuid sel juhul on sõnum ergutava sisuga – kasutajale antakse teada, et eesmärgi saavutamine on lähedal, ning kuvatakse summa, mille võrra tuleb veel säästa.

Kui eelarve jagatakse uue kasutajaga, saadetakse lisatavale kasutajale selle kohta teavitus. Samuti teavitatakse ka olemasolevaid. Teavitused saadetakse ka siis, kui kasutaja jagatud eelarvest eemaldatakse.

Lisaks teavitatakse sissemaksete tegemise ja eemaldamise kohta, et tagada parem ülevaade ja kontroll eelarve muutuste üle. Vt Lisa 10, Joonis 65

Mittefunktsionaalne nõue punkt "Kohandatavus" loetakse täidetuks: rakendus on kättesaadav nii arvuti- kui ka mobiiliseadmetes. Kasutajaliides on kohanduv ja pakub ühtset funktsionaalsust sõltumata platvormist

Kasutaja autentimine vastab turvanõuetele ning on teostatud vastavalt mittefunktsionaalsele nõude punktile "Autentimine ". Täpsem kirjeldus on esitatud alampeatükis 5.5.

Kasutajatestimise tulemused kahest testimisvoorust kinnitavad, et ka mittefunktsionaalne nõue punkt "Kasutajasõbralikkus" on täidetud. Testijad hindasid rakendust intuiitivseks ja kasutajasõbralikuks, tuues esile positiivseid kasutuskogemusi. Vt 6.1 ja 6.3

Rakenduse ligipääsetavus on tagatud domeeninime kaudu ning turvalisus kindlustatud SSL/TLS protokollide abil, kasutades HTTPS-ühendust (mittefunktsionaalne nõue punkt "Veebiturvaline"). Selle tehniline lahendus on kirjeldatud alampeatükis 5.7.

Mittefunktsionaalne nõue punkt "Mitmekeelne": kasutajaliides on saadaval kahes keeles: eesti ja inglise. Lokaliseerimine on ellu viidud i18n-raamistiku abil ning vastav lahendus on kirjeldatud samuti alampeatükis 5.4.



Sellega loetakse kõik funktsionaalsed ja mittefunktsionaalsed nõuded täidetuks ning mitmes osas isegi ületatuks, kuna arendusprotsessi ja kasutajatestimise käigus lisandus mitmeid lisafunktsionaalsusi, mis polnud esialgses nõuete loetelus.

## **6.5 Analüüs**

Autor peab antud tööd õnnestunuks. Arenduse käigus valmis rakenduse prototüüp, mis vastab kõigile funktsionaalsetele ja mittefunktsionaalsetele nõuetele. Tulemuse õigsust kinnitati nii automaatsete kui ka kahevoorulise kasutajatestimisega. Testimisel ilmnis, et potentsiaalsed kasutajad suhtuvad rakendusse positiivselt ja soovivad seda kasutada oma finantside planeerimiseks. Töö käigus kinnistusid ülikoolis omandatud teadmised ning lisaks õpiti uusi tehnoloogiaid, raamistikke ja teeke.

Olulisteks õppetundideks olid põhjalik ajaplaneerimine ning töö jaotamine väiksemateks osadeks, et lihtsustada arendusprotsessi. Samuti kogeti esmakordselt suhtlust "kliendiga" potentsiaalsete kasutajate näol. Selgus, et kliendi soovidest on tihti keeruline aru saada, kuna oma ettekujutusi rakendusest ei osata alati selgelt väljendada. Seetõttu on oluline panustada alternatiivsete lahenduste analüüsi ja toetuda parimatele praktikatele. Samuti on tähtis oskus analüüsida kliendi tagasisidet ja teha selle põhjal asjakohaseid järeldusi. Rakenduse prototüüp kättesaadav lingil: <https://sendisemu.itcollege.ee>

## **6.6 Edasised arnedusvõimalused**

Projektil on palju kasvamis- ja arenguruumi. Eelkõige plaanitakse tulevikus integreerida masinõppel põhinev algoritm, mis kasutaks diagrammidest kogutud andmeid kasutaja käitumis- ja kulutamismustrite analüüsimiseks. Selle põhjal saaks pakkuda personaalseid soovitusi eelarve koostamiseks ja finantsotsuste tegemiseks. Samuti on kavas lisada õppematerjale, mis käsitlevad finantsplaneerimise teoreetilisi aluseid, pakkudes kasutajatele paremaid teadmisi oma rahalise olukorra mõistmiseks ja haldamiseks.

Kuna rakendus on suunatud Eesti turule, plaanitakse see tulevikus ühendada Eesti suuremate pankadega, et kasutajad ei peaks oma kulutusi käsitsi sisestama. Lisaks on plaanis arendada eraldi mobiilirakendus, et pakkuda mugavamalt ja kättesaadavamalt kasutuskogemust.

## 7 Kokkuvõte

Bakalaureuse töö käigus valmis veebirakenduse prototüüp isikliku eelarvestamise jaoks, mis aitaks täita olemasolevate lahenduste puudujäägid ning pakuks tasuta, kasutajasõbralikku ja intuitiivset platvormi isiklikuks finantsplaneerimiseks.

Töö käigus analüüsiti olemasolevaid rakendusi, keskendudes nende põhifunktsionaalsustele ja nõrkadele külgedele. Saadud teadmiste põhjal viidi läbi kasutajauuring, mille eesmärk oli mõista sihtrühma ootusi ning välja selgitada rakenduses vajalikuid funktsionaalsusi. Kogutud andmete alusel koostati nõuete loend, mis realiseeriti täies mahus, kasutades Java Spring Booti raamistikku, Reacti kasutajaliidest ja PostgreSQL andmebaasi.

Prototüübi testimiseks viidi läbi kasutajatestimine, milles osales 15 inimest. Osalejatele anti konkreetset ülesandete erinevate funktsionaalsuste testimiseks ning tagasiside põhjal analüüsiti rakenduse tugevusi ja kitsaskohti. Tulemuste põhjal tehti vajalikud parandused ning viidi läbi teine testimisvoor, keskendudes esialgsete probleemide lahendamisele ja kasutajakogemuse parandamisele. Lõpuks hinnati kasutajate üldist suhtumist rakendusse, mis kinnitas antud teema aktuaalsust ning lahenduse väljatöötamise vajadust.

Projektil on palju arenguruumi ning tulevikus plaanitakse selle edasist arendamist parema kasutajakogemuse ja laiendatud funktsionaalsuste saavutamiseks. Peamiseks eesmärgiks on liidestamine Eesti suuremate pankadega, et automatiseerida kulude sisestamist ja tõsta kasutusmugavust. See eeldab ühtlasi ka kõrgendatud turvalisuse tagamist. Edaspidi nähakse võimalust integreerida masinõppel põhinevaid lahendusi, et analüüsida kasutajate kulutamismustreid ning pakkuda nende põhjal personaalseid soovitusi eelarve koostamiseks. Samuti on plaanis luua õppematerjale finantsplaneerimise alaste teadmiste suurendamiseks ning arendada eraldi mobiilirakendus, et parandada kättesaadavust ja kasutajakogemust.

## Kasutatud kirjandus

- [1] Elizabeth Breitbach ja William B. Walstad. „Financial Literacy and Financial Behavior among Young Adults in the United States“. Teoses: *Economic Competence and Financial Literacy of Young Adults: Status and Challenges*. 1. väljaanne. Köide 3. Verlag Barbara Budrich, 2016, lk. 81–98. URL: <http://www.jstor.org/stable/j.ctvbkk29d.7>.
- [2] Sanie Doda ja Shkelqim Fortuzi. „The Process of Financial Planning in Personal Finance“. *International Journal of Human Resource Studies* 5.4 (2015), lk. 28–35. URL: <https://www.macrothink.org/journal/index.php/ijhrs/article/view/8636>.
- [3] R.R. Winger B.J. ja Frasca. *Personal Finance: An Integrated Planning Approach*. Macmillan Publishing Company, 1993. URL: <https://books.google.ee/books?id=YwRFAAAAYAAJ>.
- [4] Ron Messer. „Budgets and other lies: Evidence of bias in financial planning“. *Business Horizons* 60 (aprill 2017), //todo. URL: [https://www.researchgate.net/publication/316548463\\_Budgets\\_and\\_other\\_lies\\_Evidence\\_of\\_bias\\_in\\_financial\\_planning](https://www.researchgate.net/publication/316548463_Budgets_and_other_lies_Evidence_of_bias_in_financial_planning).
- [5] Thomas Tiboh. „An Analysis of Personal Financial Literacy Among Polytechnic Students: A Case of Kumasi Polytechnics“. Master's thesis. Kwame Nkrumah University of Science ja Technology, 2015. URL: <https://ir.knust.edu.gh/server/api/core/bitstreams/98943ec6-2e4e-427c-96e9-b44f9bf05883/content>.
- [6] Barbara Apostolou *et al.* „Accounting education literature review (2015)“. *Journal of Accounting Education* 35 (2016), lk. 20–55. URL: <https://ideas.repec.org/a/eee/joaced/v35y2016icp20-55.html>.
- [7] T. P. da Silva *et al.* „Financial Education Level of High School Students and Its Economic Reflections“. *Revista de Administração (São Paulo)* 52.3 (2017), lk. 285–303. URL: <https://doi.org/10.1016/j.rausp.2016.12.010>.
- [8] Suzanne Hanna Sherman D. ja Lindamood. „Quantifying the Economic Benefits of Personal Financial Planning“. *Financial Services Review* 19.2 (2010). URL: <https://ssrn.com/abstract=1888285>.
- [9] Lady Angela Herdjiono Irine ja Damanik. „Pengaruh Financial Attitude, Financial Knowledge, Parental Income Terhadap Financial Management Behavior“. *Jurnal Manajemen Teori dan Terapan (Journal of Theoretical and Applied Management)* 9.3 (detsember 2016). URL: <https://e-journal.unair.ac.id/JMTT/article/view/3077>.
- [10] Liis-Marie Kütt. *Algkooliõpilaste rahaliste vahendite haldamise rakenduse arendus*. est. Bachelor's Thesis. Juuni 2021. URL: <https://digikogu.taltech.ee/et/Item/4b07d806-7b62-409f-a200-b33c41de767a>.

- [11] Elizabeth Gilbert. *The surprising psychological effect of tracking your expenses*. Vaadatud: 2024-02-23. September 2022. URL: <https://bigthink.com/sponsored/psychology-of-tracking-finances/>.
- [12] *You Need A Budget (YNAB)*. Vaadatud: 2024-02-23. URL: <https://www.ynab.com>.
- [13] *Goodbudget*. Vaadatud: 2024-02-23. URL: <https://goodbudget.com>.
- [14] Ramsey Solutions. *EveryDollar*. Vaadatud: 2024-02-23. URL: <https://www.ramseysolutions.com/ramseyplus/everydollar/features>.
- [15] Swedbank AS. *Minu eelarve*. Vaadatud: 2024-02-23. URL: <https://www.swedbank.ee/private/d2d/accounts/transactions>.
- [16] Microsoft Corporation. *Microsoft Excel*. Vaadatud: 2024-02-23. URL: <https://www.microsoft.com/en-us/microsoft-365/excel>.
- [17] GitLab. *About GitLab*. Vaadatud: 2024-02-28. 2025. URL: <https://about.gitlab.com>.
- [18] GeeksforGeeks. *REST API Introduction*. Vaadatud: 2025-05-11. Aprill 2025. URL: <https://www.geeksforgeeks.org/rest-api-introduction/>.
- [19] Jesse Liberty. *Programming C#: Building .NET Applications with C#*. O'Reilly Media, 2001. URL: [https://books.google.ee/books?hl=en&lr=&id=Ns4e0ko-S7QC&oi=fnd&pg=PR5&dq=why+c%23&ots=dvpPQbZ5b5&sig=H2Qx\\_2h1FLVumg89oyF6PNRQIiA&redir\\_esc=y#v=onepage&q=why%20c%23&f=false](https://books.google.ee/books?hl=en&lr=&id=Ns4e0ko-S7QC&oi=fnd&pg=PR5&dq=why+c%23&ots=dvpPQbZ5b5&sig=H2Qx_2h1FLVumg89oyF6PNRQIiA&redir_esc=y#v=onepage&q=why%20c%23&f=false).
- [20] Python Software Foundation. *What is Python? Executive Summary*. Vaadatud: 2024-02-28. URL: <https://www.python.org/doc/essays/blurb/>.
- [21] Amazon Web Services. *What is Java? - Java Programming Language Explained*. Vaadatud: 2024-02-28. URL: <https://aws.amazon.com/what-is/java/>.
- [22] Simon Maple ja Andrew Binstock. *JVM Ecosystem Report 2018 – About your Platform and Application*. Vaadatud: 2025-05-11. Oktoober 2018. URL: <https://snyk.io/blog/jvm-ecosystem-report-2018-platform-application/>.
- [23] GeeksforGeeks. *Introduction to Spring Boot*. Vaadatud: 2024-02-28. Märts 2025. URL: <https://www.geeksforgeeks.org/introduction-to-spring-boot/>.
- [24] Swagger. *About Swagger*. Vaadatud: 2025-05-13. 2025. URL: <https://swagger.io/about/>.
- [25] Mohit Joshi. *Angular vs React vs Vue: Core Differences*. Vaadatud: 2024-02-28. Mai 2025. URL: <https://www.browserstack.com/guide/angular-vs-react-vs-vue>.
- [26] Logan dev. *The Most Demanded Frontend Frameworks in 2024*. Vaadatud: 2024-02-28. Veebruar 2025. URL: <https://www.devjobsscanner.com/blog/the-most-demanded-frontend-frameworks/>.
- [27] GeeksforGeeks. *Top 10 Front-End Frameworks in 2025*. Vaadatud: 2024-02-28. Detsember 2024. URL: <https://www.geeksforgeeks.org/top-front-end-frameworks/>.
- [28] React Bootstrap. *React Bootstrap*. Vaadatud: 2025-05-13. 2025. URL: <https://react-bootstrap.netlify.app>.

- [29] Recharts. *Recharts - A composable charting library built on React and D3*. Vaadatud: 2025-05-13. 2025. URL: <https://recharts.org/en-US>.
- [30] PostgreSQL Global Development Group. *About PostgreSQL*. Vaadatud: 2025-05-12. 2025. URL: <https://www.postgresql.org/about/>.
- [31] Liquibase Community. *Introduction to Liquibase*. Vaadatud: 2025-05-12. 2025. URL: <https://docs.liquibase.com/concepts/introduction-to-liquibase.html>.
- [32] Docker Inc. *Docker Overview*. Vaadatud: 2025-05-12. 2025. URL: <https://docs.docker.com/get-started/docker-overview/>.
- [33] Hostinger Tutorials. *What is NGINX and How Does it Work?* Vaadatud: 2025-05-12. 2023. URL: [https://www.hostinger.com/tutorials/what-is-nginx#What\\_is\\_NGINX](https://www.hostinger.com/tutorials/what-is-nginx#What_is_NGINX).
- [34] Atlassian. *Kanban*. Vaadatud: 2025-05-12. 2025. URL: <https://www.atlassian.com/agile/kanban>.

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Mari-Nicole Berezovski

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Veebi-põhine rakendus isikliku eelarve koostamiseks”, mille juhendaja on Meelis Antoi
  - 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. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

04.06.2025

---

<sup>1</sup>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 – Kasutajauuringu küsimustik

### Veebipõhine rakendus isikliku eelarve koostamiseks ja finantside haldamiseks

mn.berezovski@gmail.com [Switch account](#)

Not shared

\* Indicates required question

Kui vana te olete? \*

kuni 17

18-25

26-40

40+

Kuidas hindate enda finantsplaneerimise oskusi? \*

1 2 3 4 5

Väga kehv      Väga hea

Kas teil on vajadus uue veebipõhise rakenduse järel isikliku eelarve koostamiseks ja/või finantside haldamiseks? \*

Jah

Ei

Kas kasutate praegu mõnda veebipõhist või mobiilirakendust oma isiklike finantside haldamiseks? \*


Jah


Ei

[Next](#) [Clear form](#)

Joonis 1. Küsimused kasutajauuringu vastajate profiili ja hoiakute kohta finantsrakenduste suhtes.

## Veebipõhine rakendus isikliku eelarve koostamiseks ja finantside haldamiseks

mn.berezovski@gmail.com [Switch account](#) 

 Not shared

\* Indicates required question

### Varasem kokkupuude finantsrakendustega

Kui tihti te kasutate veebipõhist või mobiilirakendust oma isiklike finantside haldamiseks? \*

ostu sooritamisel

kord päevas

kord nädalas

kord kuus

Other: \_\_\_\_\_

Mis põhjusel te kasutate veebipõhist või mobiilirakendust oma isiklike finantside haldamiseks? \*

suurema ostu tegemiseks

raha kogumiseks

planeerin enda kulusid

reisimiseks

oma äri alustamiseks

igapäevaste väljaminekute jälgimiseks


Other: \_\_\_\_\_


[Back](#) [Next](#) [Clear form](#)

Joonis 2. Kasutajauuringu küsimused varasema finantslahenduste kasutajakogemusega vastanute kasutusmusterite kohta.



## Veebipõhine rakendus isikliku eelarve koostamiseks ja finantside haldamiseks

mn.berezovski@gmail.com [Switch account](#) 

 Not shared

\* Indicates required question

### Potentsiaalne finantsrakenduste kasutamine

Kui tihti kasutaksite veebipõhist või mobiilirakendust oma isiklike finantside haldamiseks? \*

ostu sooritamisel

kord päevas

kord nädalas

kord kuus

Other: \_\_\_\_\_

Mis põhjusel te kasutaksite veebipõhist või mobiilirakendust oma isiklike finantside haldamiseks? \*

suurema ostu tegemiseks

raha kogumiseks

planeerin enda kulusid

reisimiseks

oma äri alustamiseks


igapäevaste väljaminekute jälgimiseks


Other: \_\_\_\_\_

[Back](#) [Next](#) [Clear form](#)

Joonis 3. Kasutajauuringu küsimused varasema finantslahenduste kasutajakogemusega vastanute potentsiaalsete kasutusmusterite kohta.

## Veebipõhine rakendus isikliku eelarve koostamiseks ja finantside haldamiseks

mn.berezovski@gmail.com [Switch account](#) 

 Not shared

\* Indicates required question

### Isikliku eelarve koostamise ja finantside haldamise rakenduse testimine

Kas oleksite huvitatud osalemisest isikliku eelarve ja finantside haldamise rakenduse testimises? \*

Jah

Ei

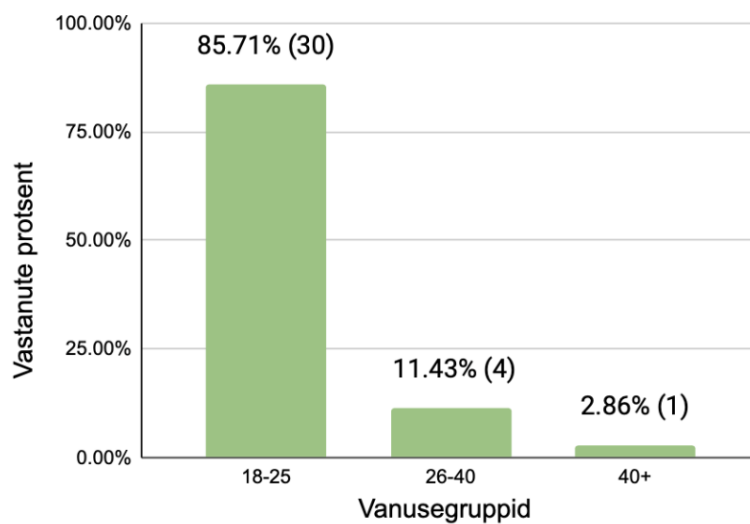
Juhul kui vastasite "Jah", jätkke enda e-maili aadress, et saaksime teiega ühendust võtta

Your answer \_\_\_\_\_

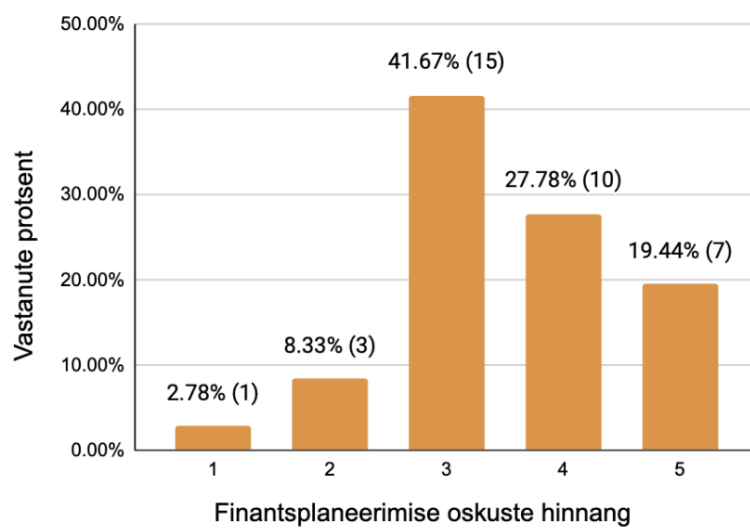
[Back](#) [Submit](#) [Clear form](#)

Joonis 4. Kasutajauuringu küsimused osalejate valmisoleku kohta rakendust testida.

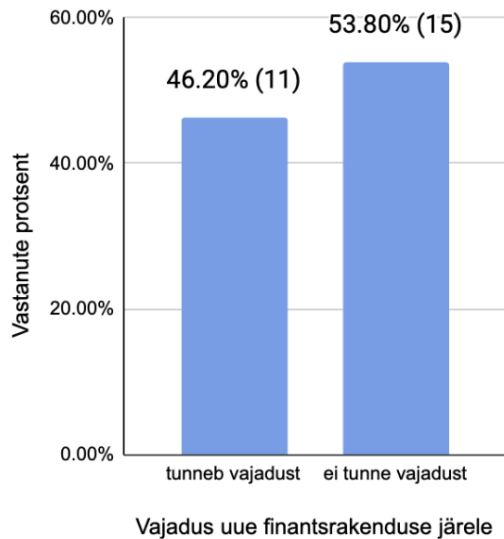
## Lisa 3 – Kasutajauuringu tulemused



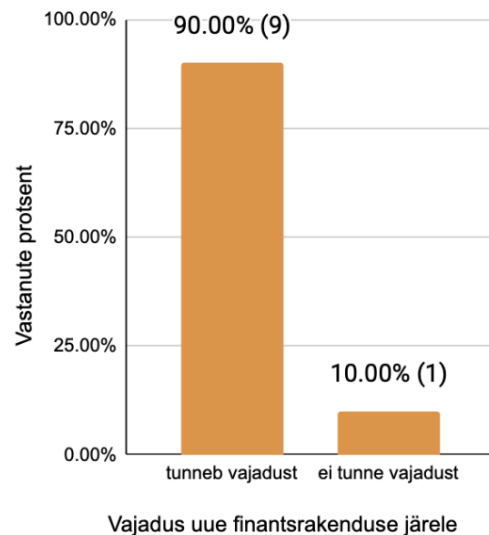
Joonis 5. Kasutajauuringu vastanute vanus



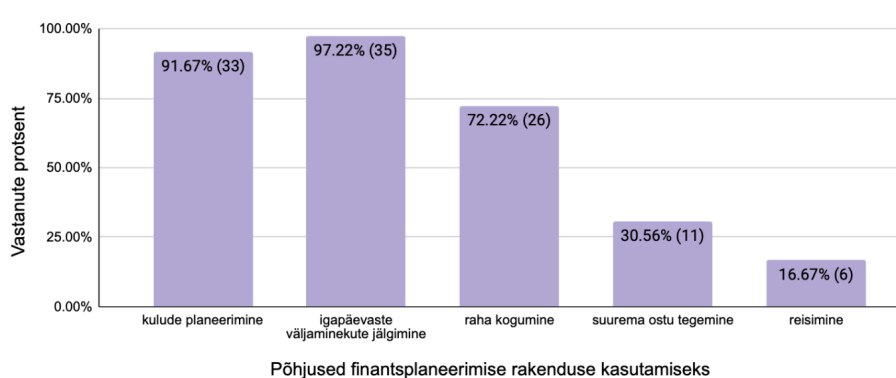
Joonis 6. Kasutajauuringu vastanute finantsplaneerimise oskuste hinnang



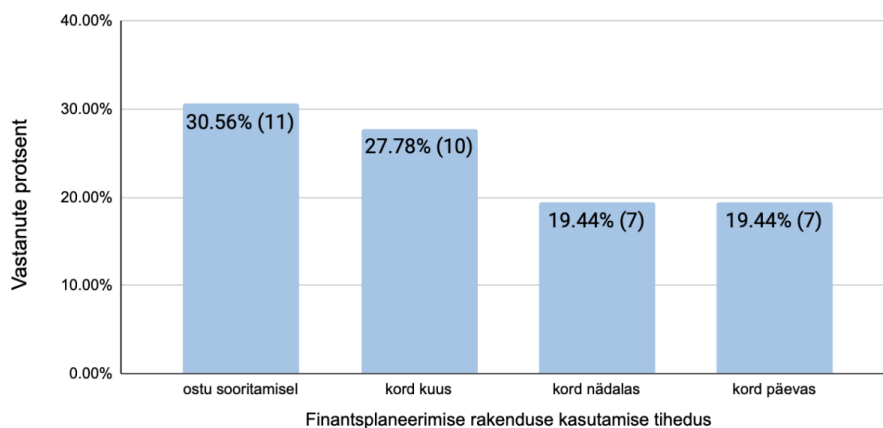
Joonis 7. Varasema finantslahenduste kogemusest vastanute vajadustaju veebipõhiste finantslahenduste järele



Joonis 8. Varasema finantslahenduste kasutajakogemusega vastanute hinnang vajadusele uue finantsrakenduse järele

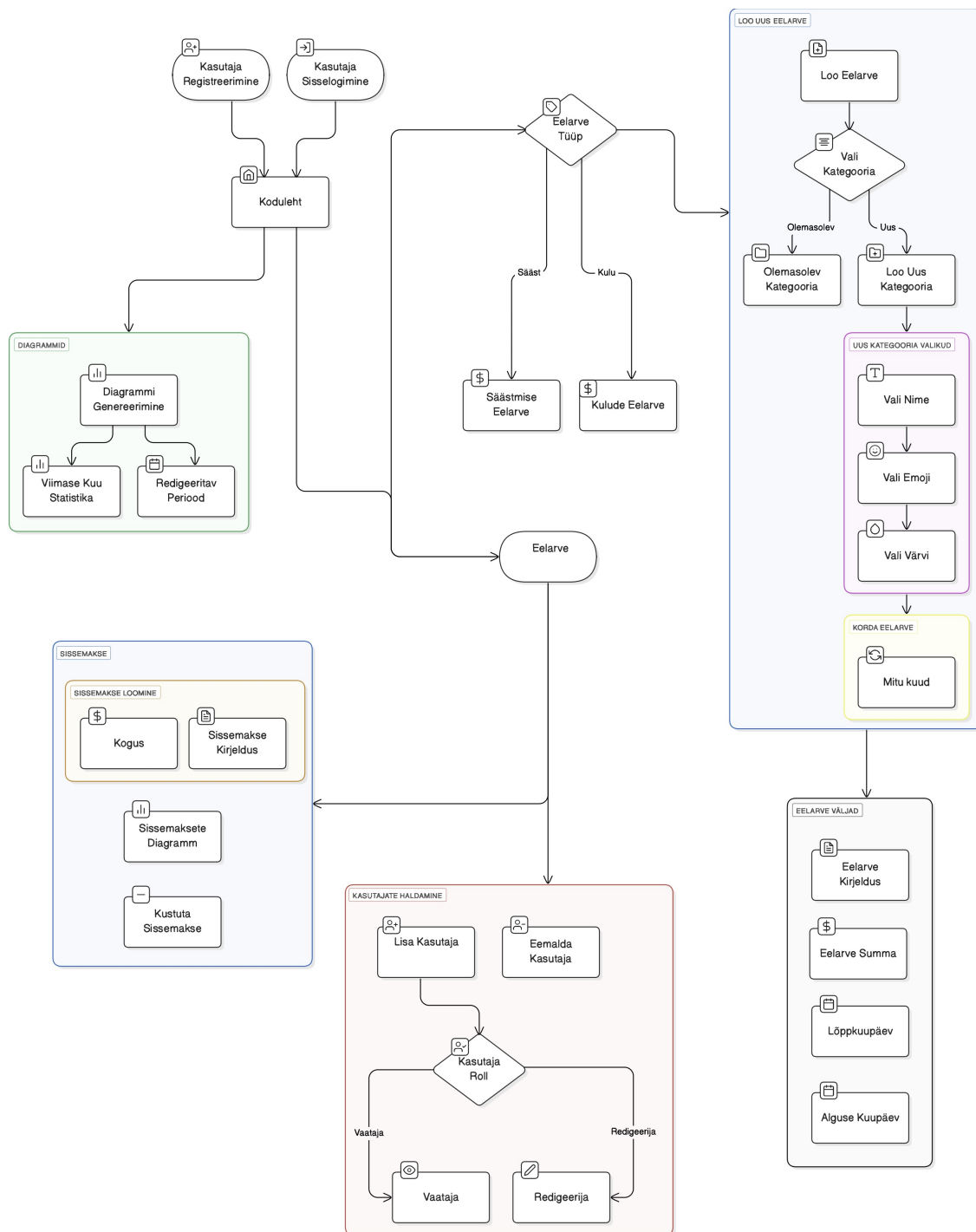


Joonis 9. Kasutajauuringu vastanute finantsplaneerimise rakenduse kasutamise põhjused



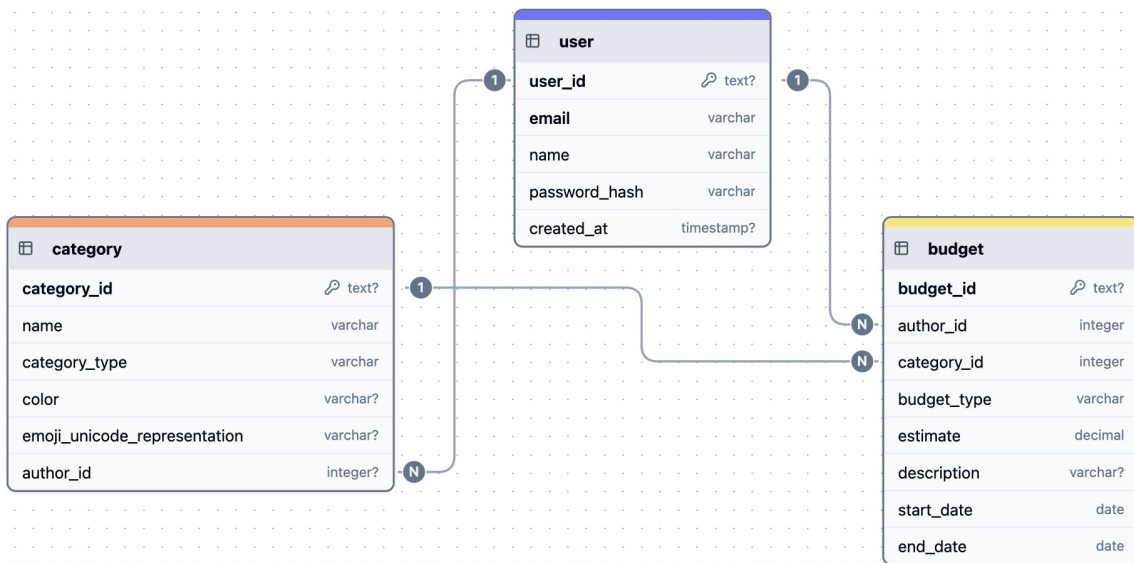
Joonis 10. Kasutajauuringu vastanute finantsplaneerimise rakenduse eelistatud kasutussagedus

## Lisa 4 – Kasutajavoog

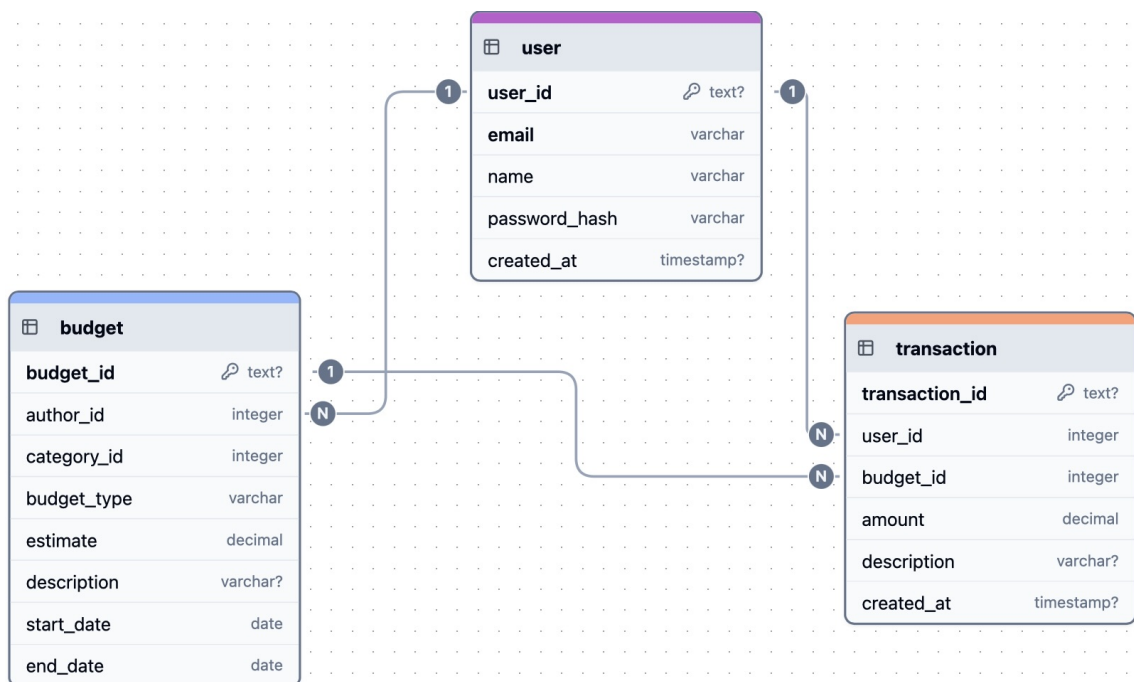


Joonis 11. Rakenduse kasutajavoog

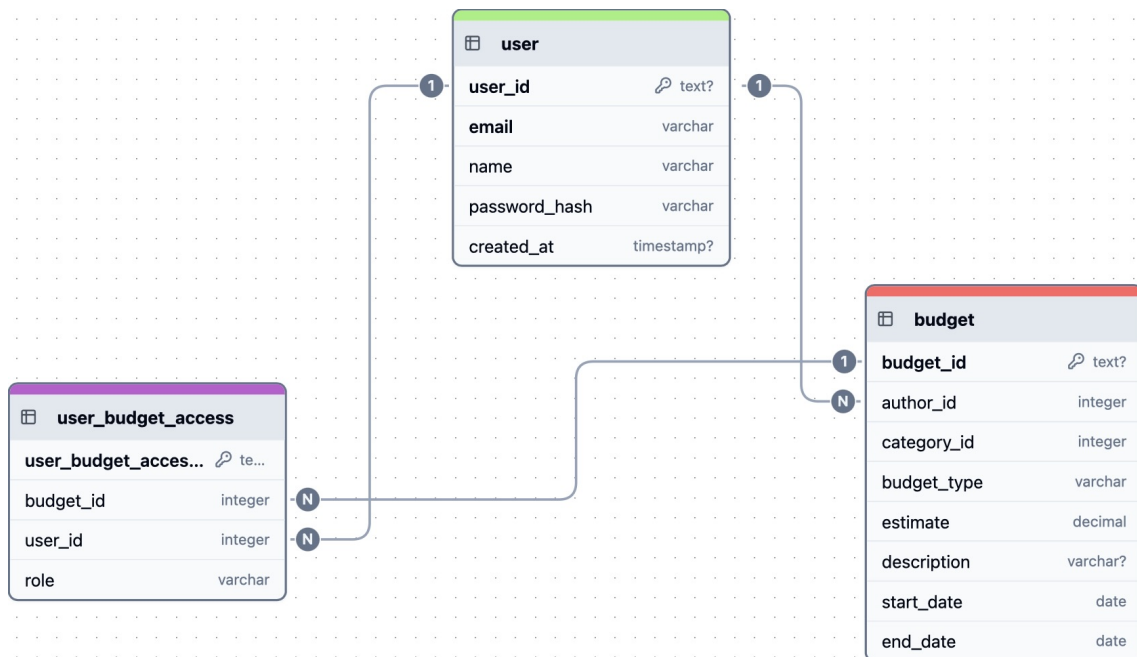




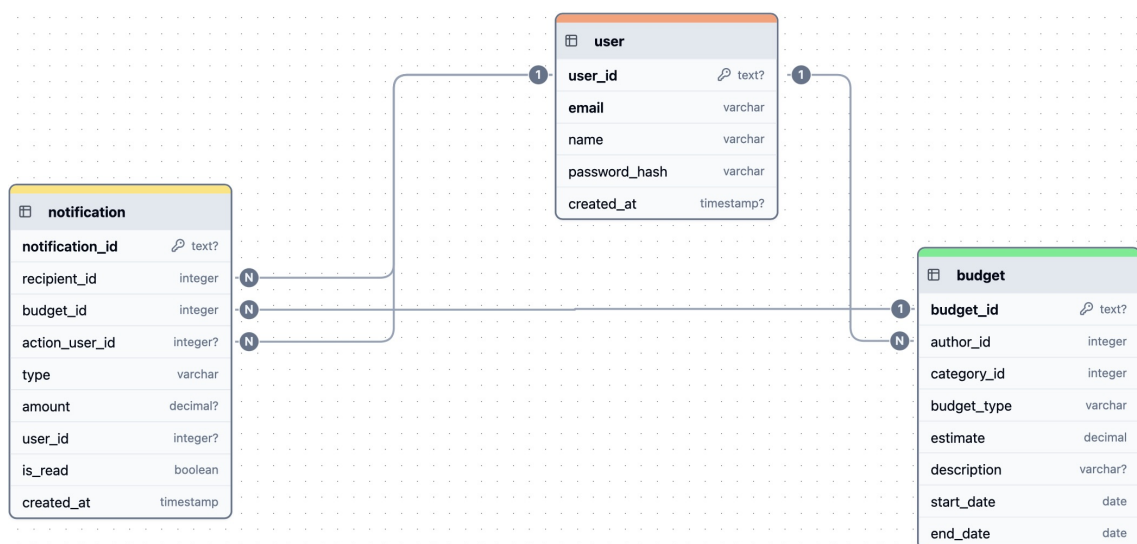
Joonis 13. Kasutaja, eelarve ja kategooria andmestruktuurid ja nende seosed



Joonis 14. Kasutaja, eelarve ja tehingute andmestruktuurid ja nende seosed



Joonis 15. Kasutaja, eelarve ja ligipääsude andmestruktuurid ja nende seosed



Joonis 16. Kasutaja, eelarve ja teavituste andmestruktuurid ja nende seosed



## Lisa 6 – Kasutajatestimise esimese voo küsimustik

### SendiSemu kasutajatestimine

Olen Mari-Nicole Berezovski TalTechi Informaatika 3. kursuse tudeng. Bakalaureusetöö raames viin läbi kasutajatestimist isikliku eelarve ja finantside haldamise veebirakendusele SendiSemu. Tegemist on prototüübiga.

Rakendus on saadaval lingil [SendiSemu](#)  
Registreeri ennast kasutajaks ja logi sisse!

Kasutajatestimine võtab aega alla 15 minuti. Küsimustik on anonüümne.

Teie tagasiside on mulle väga väärtuslik ja suureks abiks – aitäh juba ette!

#### Kategooriate loomine

Palun loo kulu kategooria määrates sellele värvi ja emotikoni ning loo sellest eelarve

Kui lihtne oli kategooria loomine? \*

	1	2	3	4	5	
raske	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lihtne

Kas vaikimisi kategooriad olid arusaadavad ja piisavalt täpsed? \*

ei

jah

Kas sul tekkis segadust või mõni hetk, kus ei teadnud, mida edasi teha? Selgita \*

Your answer

#### Eelarve planeerimine

Loo kulude eelarve toidukaupadele ja tee sinna sissemakse.

Kuidas oli sinu kogemus eelarve määramisel? Kas see oli arusaadav? \*

ei

jah

Kas eelarve sisestamine ja sissemakse salvestamine tundus lihtne? \*

	1	2	3	4	5	
raske	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lihtne

Kas said aru, kuidas jälgida eelarve täitumist? Selgita \*

Your answer

Joonis 17. Kasutajatestimise esimese voo küsimustik: kategooriate loomine ja eelarve planeerimine

**Säästmise eesmärkide seadmine või ostu planeerimine**

Loo uus säästmise eesmärk (nt puhkusereis) summaga 500€. Lisa sellele käsitsi 50€ ja vaata progressi.

Kui lihtne oli eesmärki luua ja sellele raha lisada? \*

1 2 3 4 5

raske      lihtne

Kas säästu jälgimine oli visuaalselt ja sisuliselt arusaadav? \*

ei

jah

Kas sa mõistad, kuidas rakendus aitab sul eesmärki saavutada? Selgita \*

Your answer

**Kulutuste jagamine teistega**

Jaga oma puhkuseisei säästmise eesmärk kellegagi. Nt minuga mari@test.ee

Kui kerge oli jagamise funktsiooni kasutada? \*

1 2 3 4 5

raske      lihtne

Kas said aru, kuidas koostöö toimib (kes kui palju panustab, kes näeb infot)? \*

ei

jah

Kas eelarvete jagamine teistega on kasulik? \*

ei

jah

"Ei" puhul selgita ja/või tee ettepanek \*

Your answer

Joonis 18. Kasutajatestimise esimese vooru küsimustik: säästude ja ligipääsude loomine

**Teavitused ja märguanded**

Teavitused tulevad kui keegi lisab sind oma eelarvesse või teeb jagatud eelarvesse sissemakse. Samuti saadetakse teavitus juhul, kui eelarve saab peaaegu täis või kui summa ületab seatud limiidi.

Kui kasulik on teavituste saatmine kulude ületamisel ja eesmärkide saavutamisel? \*

1 2 3 4 5

pole üldse kasulik      väga kasulik

Kas sõnumid aitavad kulusid vähendada ja säästa? \*

ei

jah

**Kulude visuaalseerimine**

Kaks graafikut asuvad kodulehel ning üks tehingute vaates

Kuidas hindad kulude graafilist esitust? Kas see on lihtsasti loetav ja arusaadav? \*

1 2 3 4 5

raske      lihtne

Milline graafik või visuaal oli kõige ülevaatlikum? Miks? \*

Your answer

Kas tunned, et see aitab sul paremini mõista oma rahalist olukorda? Selgita. \*

Your answer

**Tagasiside**

Millisel viisil aitab antud lahendus teil paremini oma eelarvet ja finantse planeerida? \*

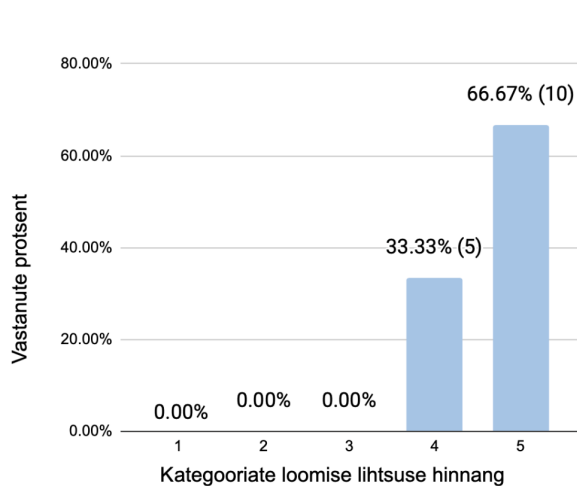
Your answer

Juhul kui on olemas ettepanekuid või mõtteid, kirjuta need siia :)

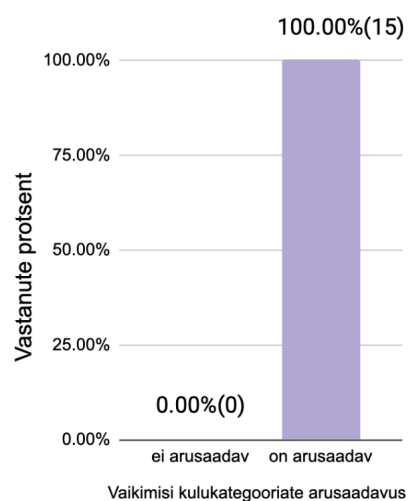
Your answer

Joonis 19. Kasutajatestimise esimese vooru küsimustik: teavitused ja märguanded, diagrammid

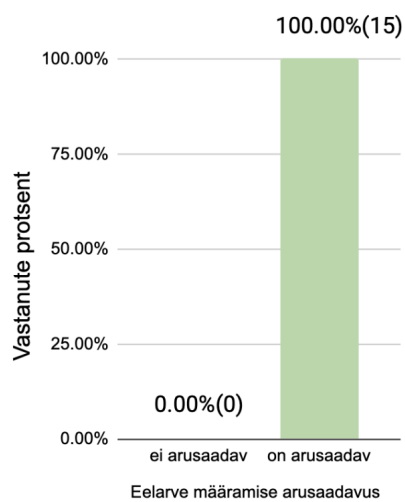
## Lisa 7 – Kasutajatestimise esimese vooru tulemused



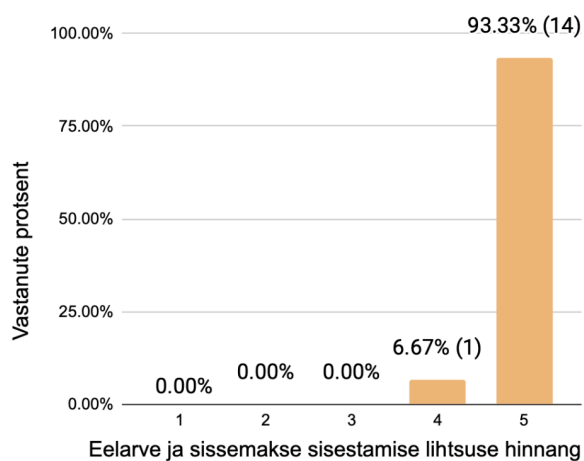
Joonis 20. Kasutajatestimise esimene voor: kategoriate loomise lihtsuse hinnang



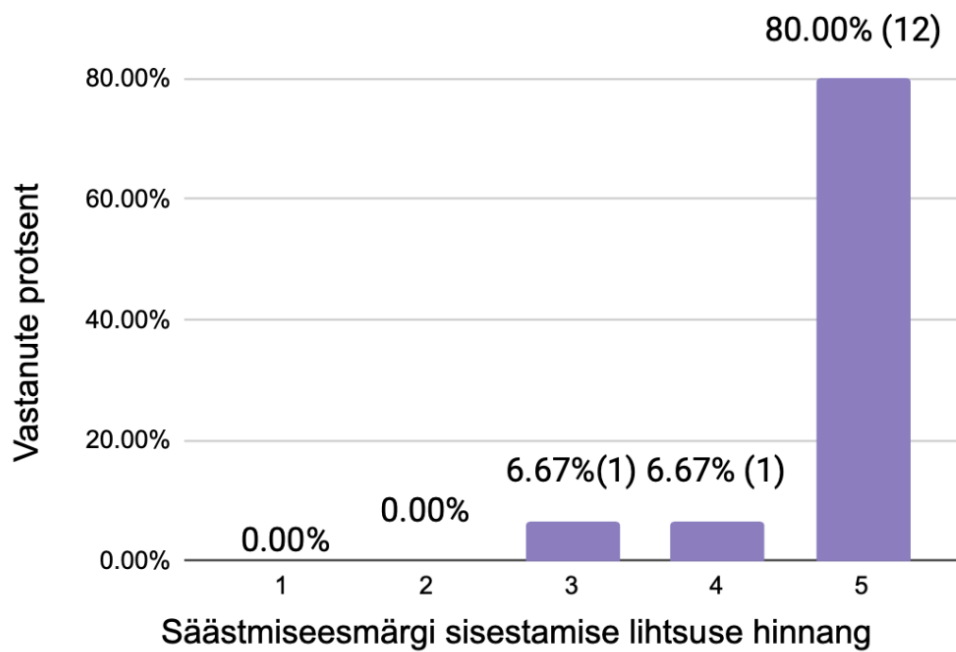
Joonis 21. Kasutajatestimise esimene voor: vaikimisi kulukategoriate arusaadavus



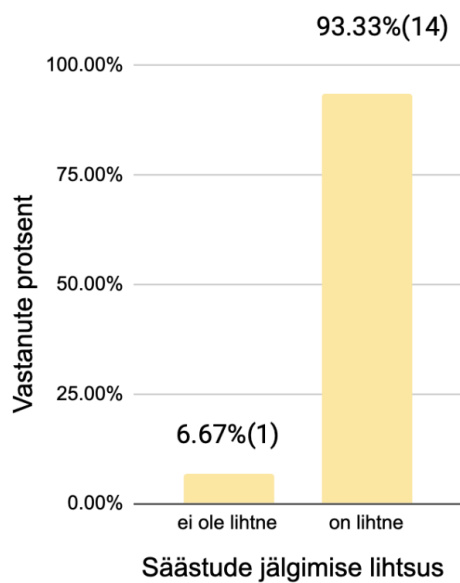
Joonis 22. Kasutajatestimise esimene voor: eelarve määramise arusaadavus



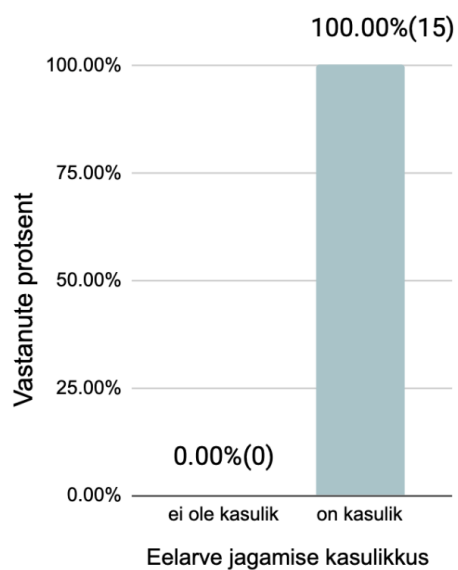
Joonis 23. Kasutajatestimise esimene voor: eelarve ja sissemakse loomise lihtsuse hinnang



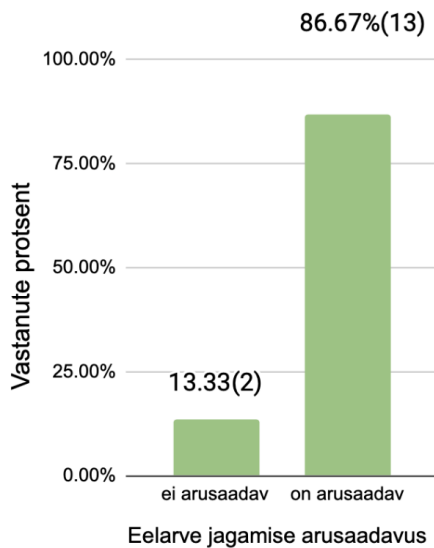
Joonis 24. Kasutajatestimise esimene voor: säästmiseesmärgi sisestamise lihtsuse hinnang



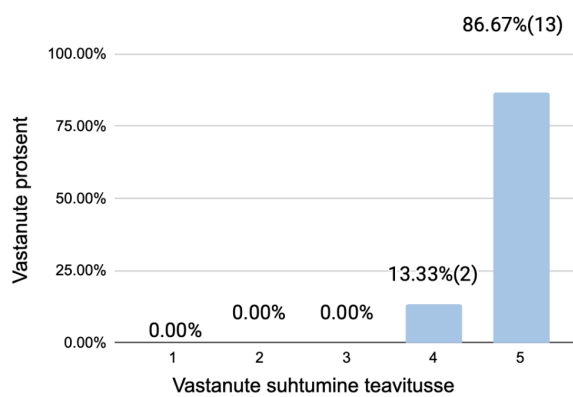
Joonis 25. Kasutajatestimise esimene voor: säästude jälgimise lihtsus



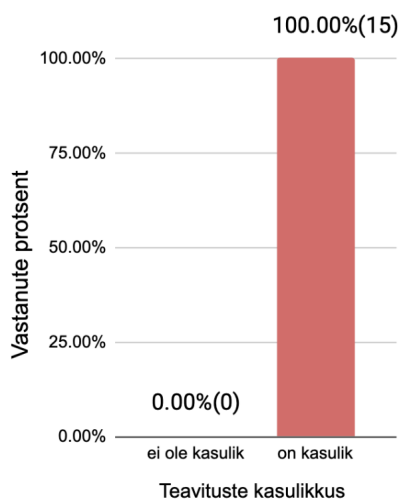
Joonis 26. Kasutajatestimise esimene voor: eelarve jagamise kasulikkus



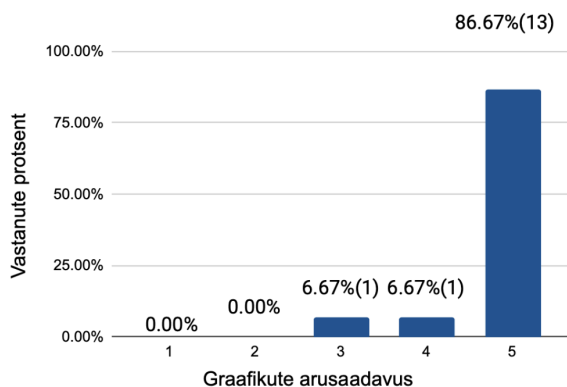
Joonis 27. Kasutajatestimise esimene voor: eelarve jagamise arusaadavus



Joonis 28. Kasutajatestimise esimene voor: vastanute suhtumine teavitusse



Joonis 29. Kasutajatestimise esimene voor: teavituste kasulikkus



Joonis 30. Kasutajatestimise esimene voor: graafikute arusaadavuse hinnang

## Lisa 8 – Kasutajatestimise teise vooru küsimustik

### SendiSemu kasutajatestimine, teine voor

Olen Mari-Nicole Berezovski TalTechi Informaatika 3. kursuse tudeng. Bakalaureusetöö raames viin läbi kasutajatestimist isikliku eelarve ja finantside haldamise veebirakendusele SendiSemu. Tegemist on prototüübiga.

Rakendus on saadaval lingil [SendiSemu](#)  
Registreeri ennast kasutajaks ja logi sisse!

Tegu on kasutajatestimise teise vooruga, Olen põhjalikult analüüsinud teie vastuseid ning tegin vastavaid muudatusi ja parandusi. Kasutajatestimine võtab aega alla 15 minuti. Küsimustik on anonüümne.

Teie tagasiside on mulle väga väärtuslik ja suureks abiks – aitäh juba ette!

#### Navigatsioon ja nähtavus

Kas leidsid nuppu "Loo Kulude Eelarve" kiiresti? \*

ei

jah

Kas nupp "Loo Säästude Eelarve" paistab hästi silma? \*

ei

jah

Kas leidsid nuppu "Kuva Ühiseid Eelarveid" kiiresti? \*

ei

jah

#### Ühised ja isiklikud eelarved

Kui jagad eelarvet kellegagi, kas ootad seda näha enda või ühiste eelarvete alt? \*

enda

ühiste

Joonis 31. Kasutajatestimise teise vooru küsimustik: navigatsioon ja eelarvete jagamine

**Tehingute kustutamine**

Kas hüplikaken tehingu kustutamisel on vajalik? \*

ei

jah

---

**Selgita \***

Your answer

---

**Eelarve korduv loomine**

Kui kasulik on eelarvete korduv loomine? \*

1      2      3      4      5

pole üldse                        väga

---

Kas soovid rohkem kontrolli selle üle kuidas korduvad eelarved kuude vahel jagunevad? \*

ei

jah

---

**Selgita \***

Your answer

---

**Üldine kogemus**

Kas muudatused muutsid kasutajakogemust paremaks? \*

ei

jah

---

Hinda kasutajakogemust \*

1      2      3      4      5

väga halb                        väga hea

---

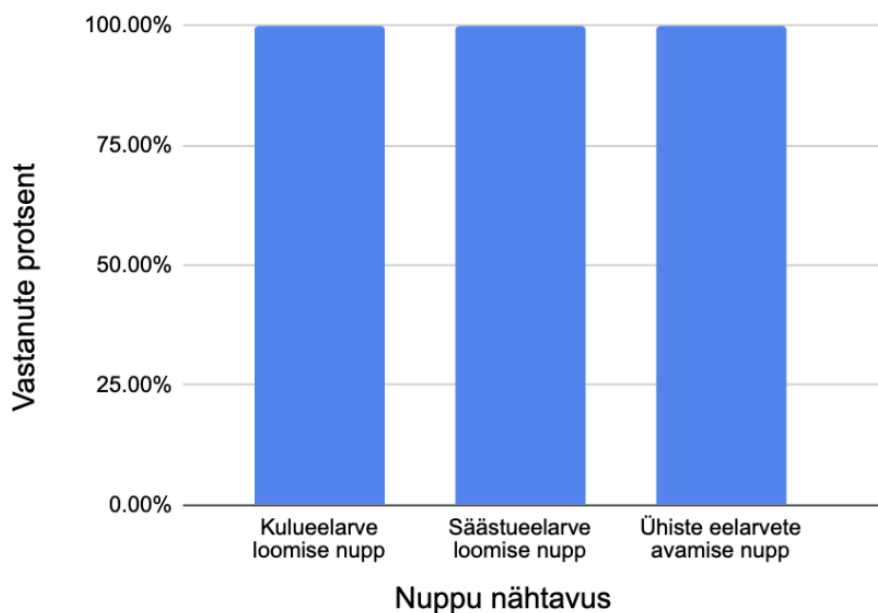
Kas on veel midagi, mis võiks olla intuitiivsem, kiirem või selgem? Tee ettepanekut! \*

Your answer

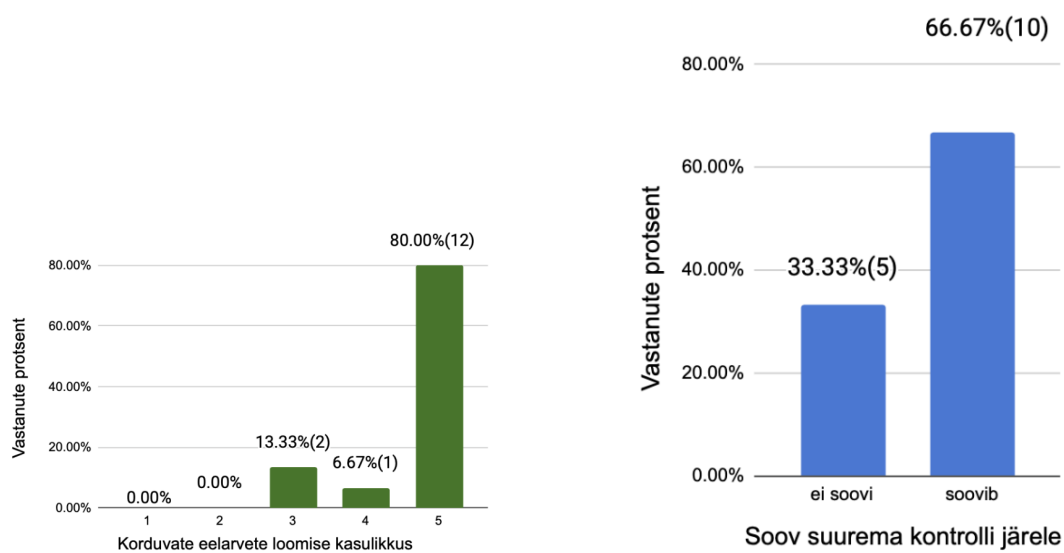
Joonis 32. Kasutajatestimise teise vooru küsimustik: tehingute kustutamine, eelarvete loomine ja kogemus



## Lisa 9 – Kasutajatestimise teise vooru tulemused

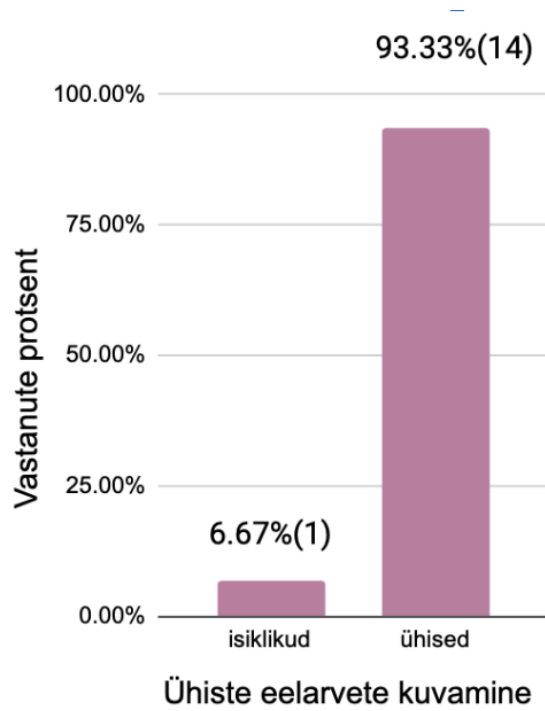


Joonis 33. Kasutajatestimise teine voor: nuppude nähtavuse hinnang

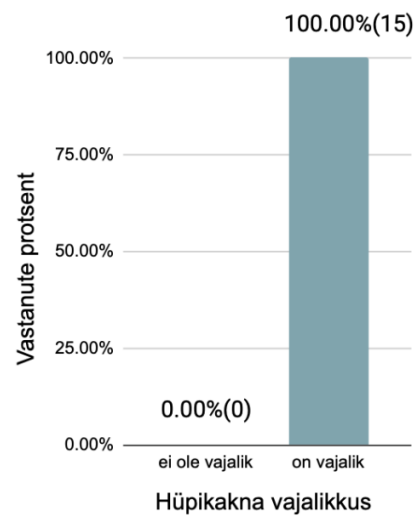


Joonis 36. Kasutajatestimise teine voor: ühiste eelarvete kuvamise hinnang

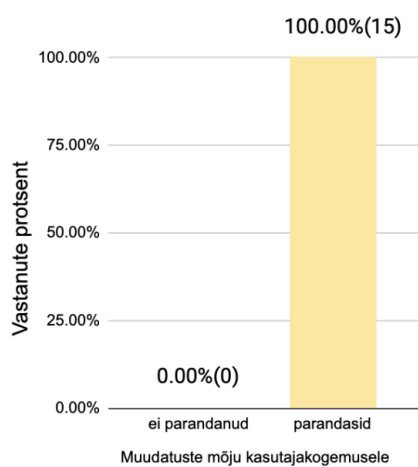
Joonis 37. Kasutajatestimise teine voor: hüpikakna vajalikkuse hinnang



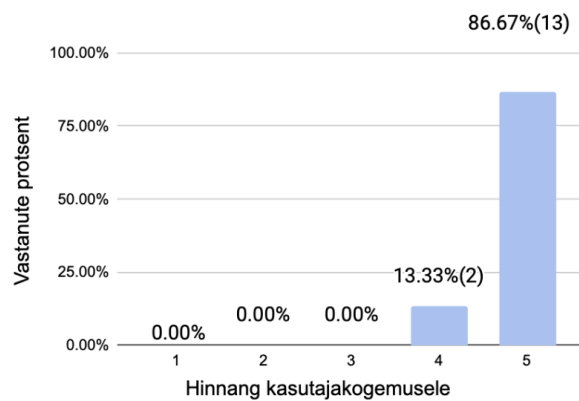
Joonis 34. Kasutajatestimise teine voor: korduvate eelarvete loomise kasulikkus



Joonis 35. Kasutajatestimise teine voor: soov korduvate eelarvete suurema kontrolli järele

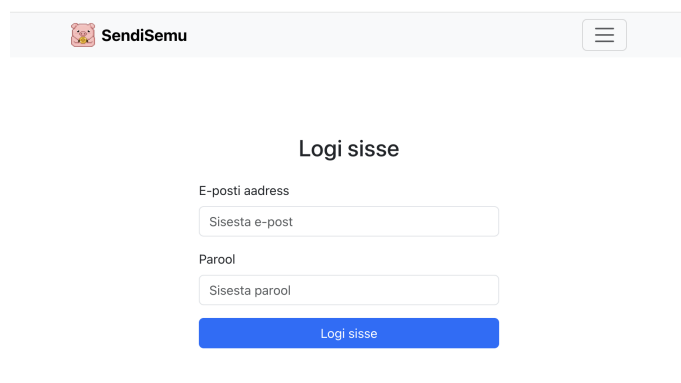


Joonis 38. Kasutajatestimise teine voor: tehtud muudatuste mõju kasutajakogemusele



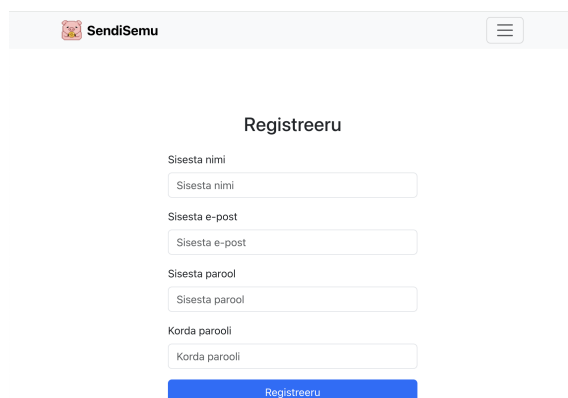
Joonis 39. Kasutajatestimise teine voor: hinnang üldisele kasutajakogemusele

## Lisa 9 – Rakenduse kuvatõmmised



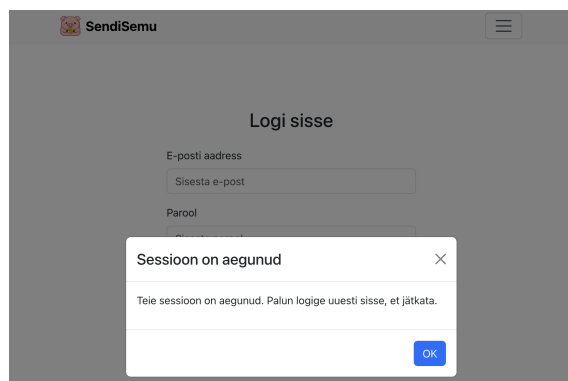
The screenshot shows the login page of the SendiSemu application. At the top left is the SendiSemu logo, and at the top right is a menu icon. The main heading is "Logi sisse". Below it, there are two input fields: "E-posti aadress" (Email address) with the placeholder "Sisesta e-post" and "Parool" (Password) with the placeholder "Sisesta parool". A blue button labeled "Logi sisse" is positioned below the password field.

Joonis 40. Rakenduse kuvatõmmis: sisselogimisleht



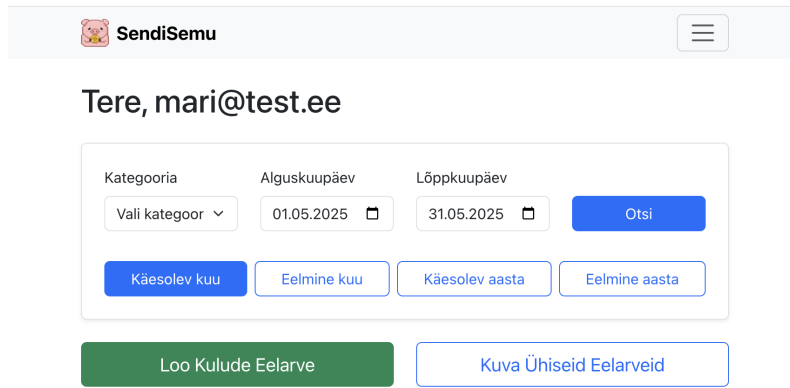
The screenshot shows the registration page of the SendiSemu application. At the top left is the SendiSemu logo, and at the top right is a menu icon. The main heading is "Registreeru". Below it, there are five input fields: "Sisesta nimi" (Enter name) with placeholder "Sisesta nimi", "Sisesta e-post" (Enter email) with placeholder "Sisesta e-post", "Sisesta parool" (Enter password) with placeholder "Sisesta parool", and "Korda parooli" (Repeat password) with placeholder "Korda parooli". A blue button labeled "Registreeru" is positioned below the last field.

Joonis 41. Rakenduse kuvatõmmis: registreerimisleht

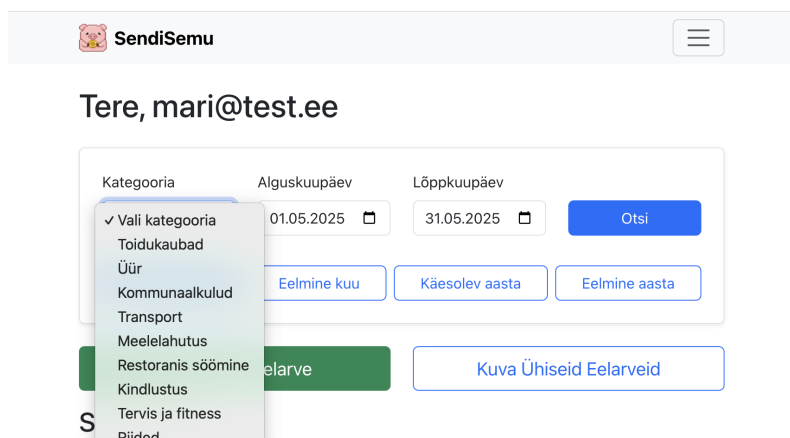


The screenshot shows the login page of the SendiSemu application with a session timeout dialog box overlaid. The background is dimmed. The dialog box has the title "Sessioon on aegunud" (Session has expired) and a close button (X). The text inside the dialog reads: "Teie sessioon on aegunud. Palun logige uuesti sisse, et jätkata." (Your session has expired. Please log in again to continue.) There is a blue "OK" button at the bottom right of the dialog.

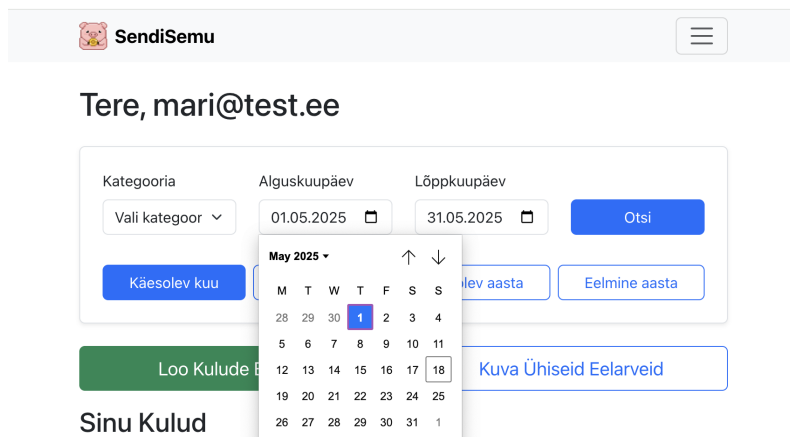
Joonis 42. Rakenduse kuvatõmmis: väljalogimise hü-pikaken



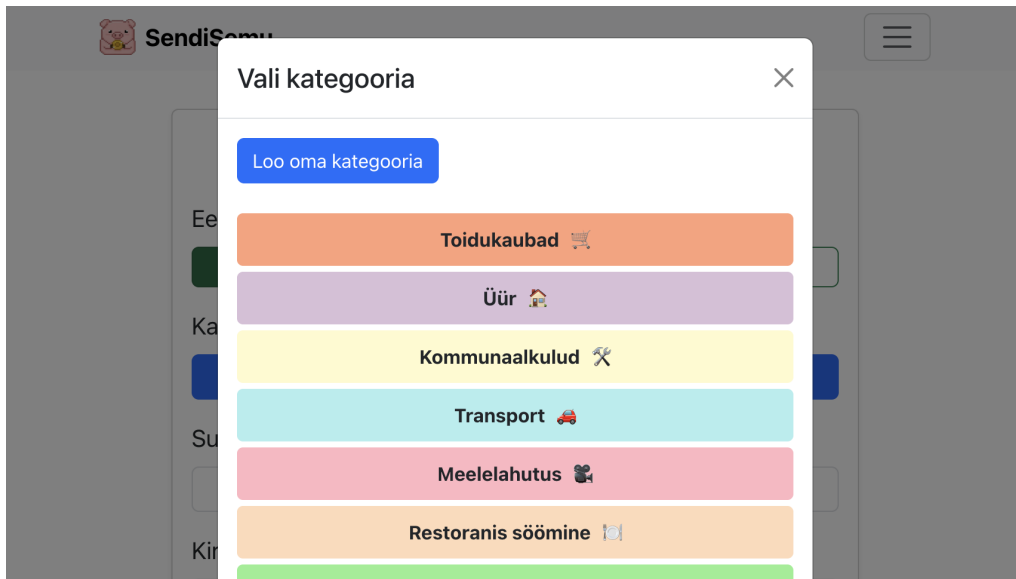
Joonis 43. Rakenduse kuvatõmmis: otsinguriba



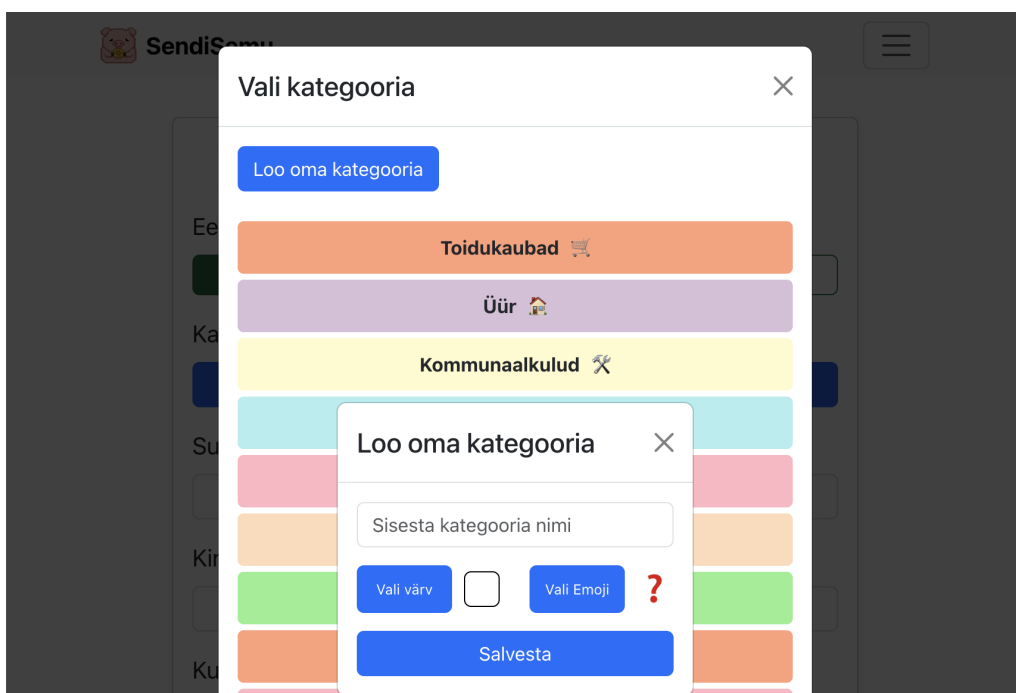
Joonis 44. Rakenduse kuvatõmmis: otsing kategooriate kaudu



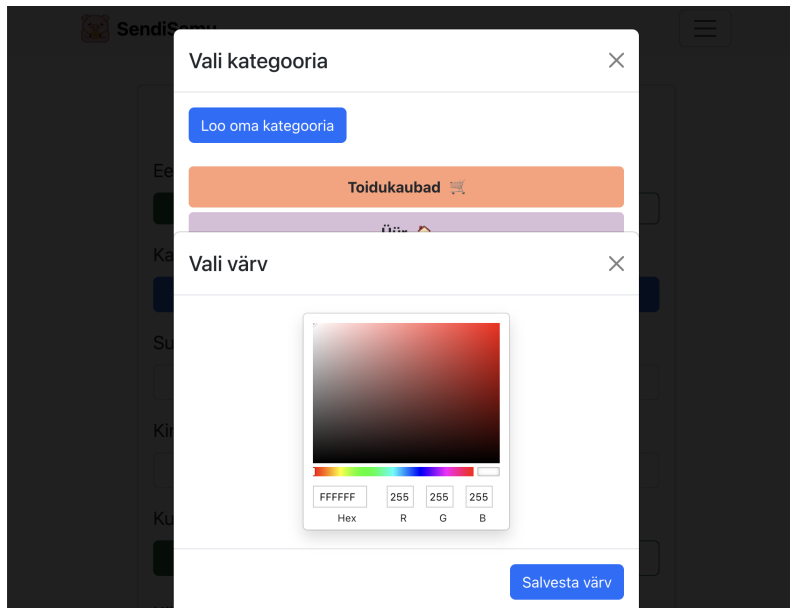
Joonis 45. Rakenduse kuvatõmmis: otsing kuupäevade kaudu



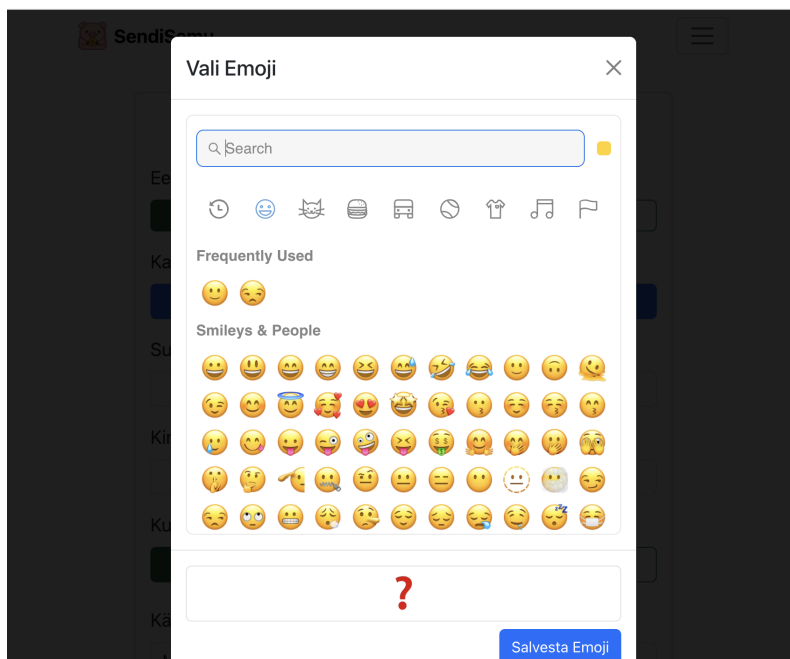
Joonis 46. Rakenduse kuvatõmmis: kategooriate valik



Joonis 47. Rakenduse kuvatõmmis: kategooria loomine



Joonis 48. Rakenduse kuvatõmmis: värvide valik kategooria loomisel



Joonis 49. Rakenduse kuvatõmmis: emotikoni valik kategooria loomisel



## Loo eelarve

Eelarve tüüp

Kulu 🌸

Sääst 💰

Kategooria

Vali kategooria

Summa €

Kirjeldus (Valikuline)

Kuupäeva valik

Vali kuu

Täpsed kuupäevad

Käesolev kuu

Mai

Vali aasta

2025

Alguskuupäevaks määratakse esimene ja lõppkuupäevaks selle kuu viimane päev.

Korda

1

Mitmeks järgnevas kuuks see eelarve tuleks korrata alates valitud kuust k.a valitud kuu.

Loo eelarve

Joonis 50. Rakenduse kuvatõmmis: kulueelarvete vaade

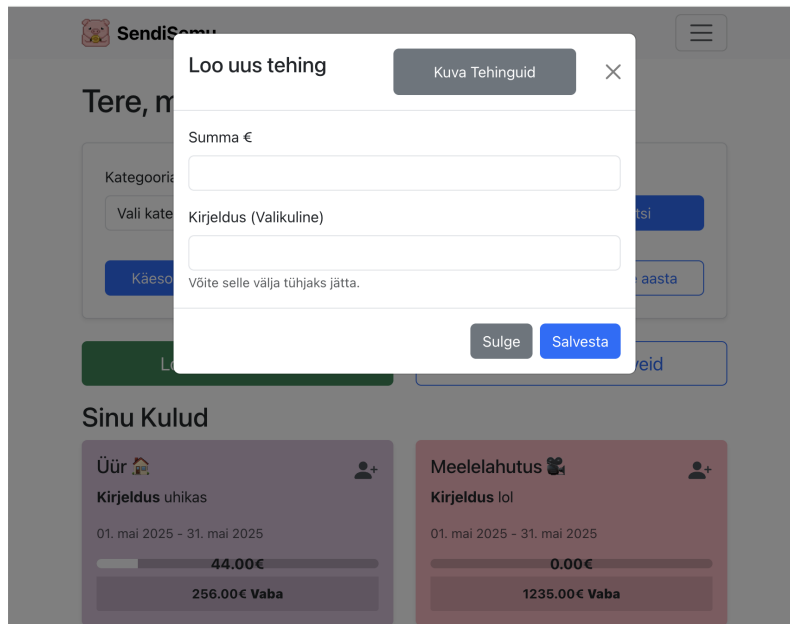
Tere, mari@test.ee

Kategooria	Alguskuupäev	Lõppkuupäev	
Vali kategoor <input type="text"/>	01.05.2025 <input type="text"/>	31.05.2025 <input type="text"/>	<input type="button" value="Otsi"/>
<input type="button" value="Käesolev kuu"/>	<input type="button" value="Eelmine kuu"/>	<input type="button" value="Käesolev aasta"/>	<input type="button" value="Eelmine aasta"/>

## Sinu Kulud



Joonis 51. Rakenduse kuvatõmmis: sissemakse tegemine



Joonis 52. Rakenduse kuvatõmmis: sissemakse tehtud



SendiSemu

Eelarve Üür kogusumma ületab 90% hinnangulisest summast. Alles on 0€.

Tere, mari@test.ee

Kategooria: Vali kategoor ▾ | Alguskuupäev: 01.05.2025 📅 | Lõppkuupäev: 31.05.2025 📅 | Otsi

Käesolev kuu | Eelmine kuu | Käesolev aasta | Eelmine aasta

Loo Kulude Eelarve | Kuva Ühiseid Eelarveid

### Sinu Kulud

Üür 🏠

Kirjeldus uhikas

01. mai 2025 - 31. mai 2025

300.00€

0.00€ Vaba

Meelelahutus 🎬

Kirjeldus lol

01. mai 2025 - 31. mai 2025

0.00€

1235.00€ Vaba

Joonis 53. Rakenduse kuvatõmmis: kulueelarve tehtud sissemaksega

SendiSemu

Tere, mari@test.ee

Kategooria: Vali kategoor ▾ | Alguskuupäev: 01.05.2025 📅 | Lõppkuupäev: 31.05.2025 📅 | Otsi

Käesolev kuu | Eelmine kuu | Käesolev aasta | Eelmine aasta

Loo Säästude Eelarve | Kuva Ühiseid Eelarveid

### Sinu Säästud

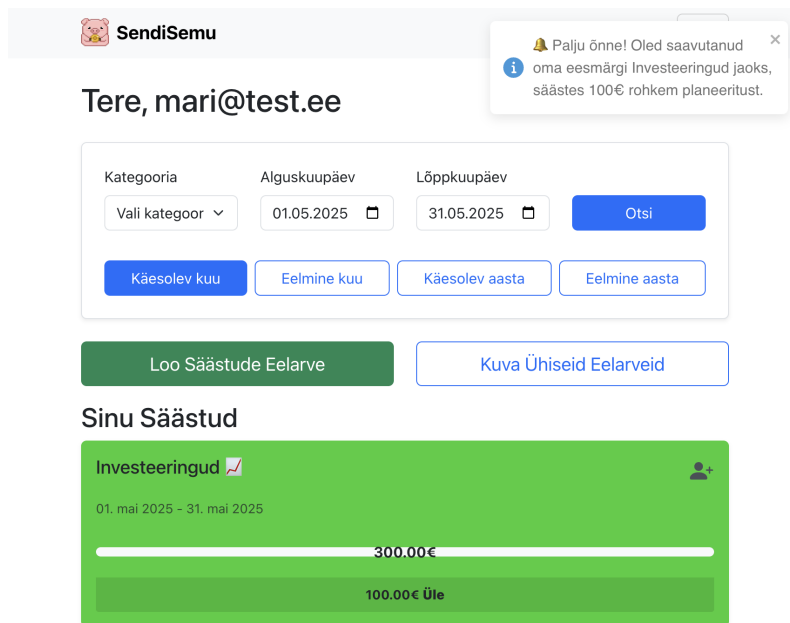
Investeeringud 📈

01. mai 2025 - 31. mai 2025

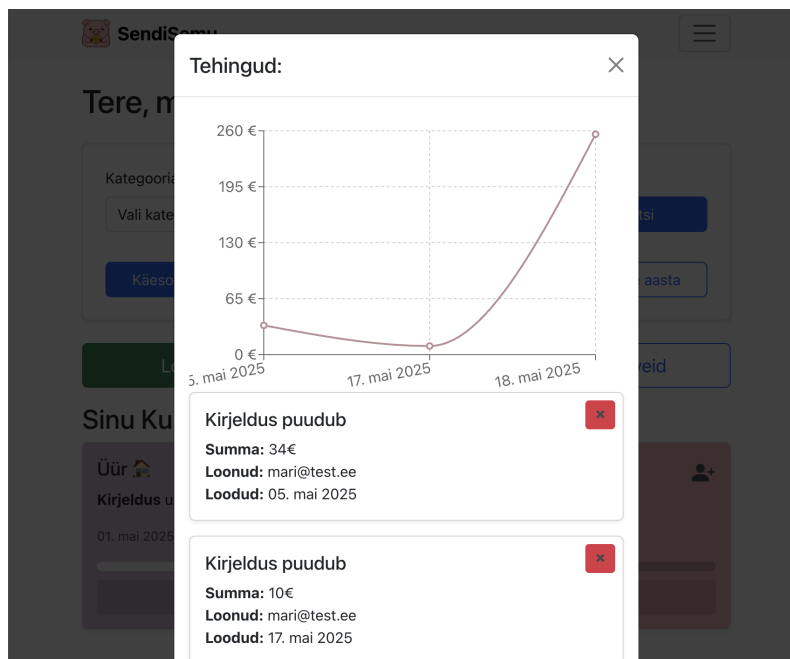
0.00€

200.00€ Jäänud koguda

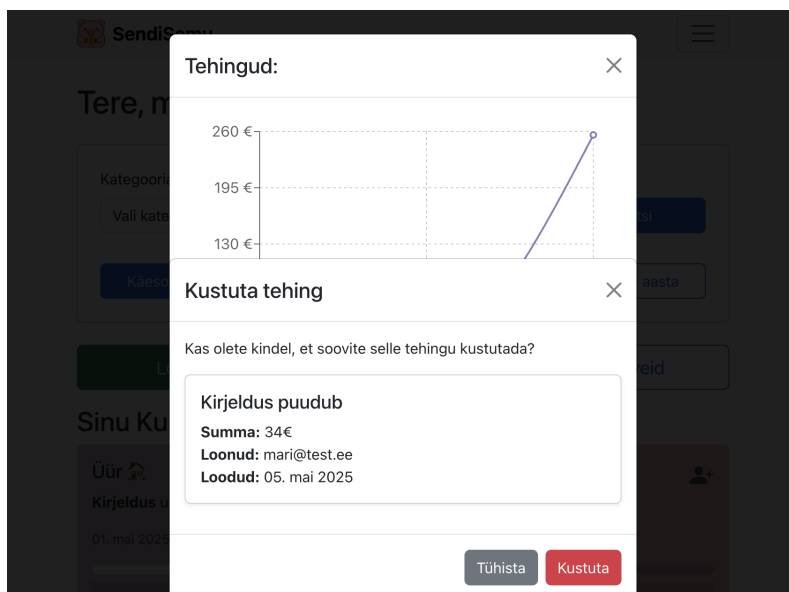
Joonis 54. Rakenduse kuvatõmmis: säästueesmärkide vaade



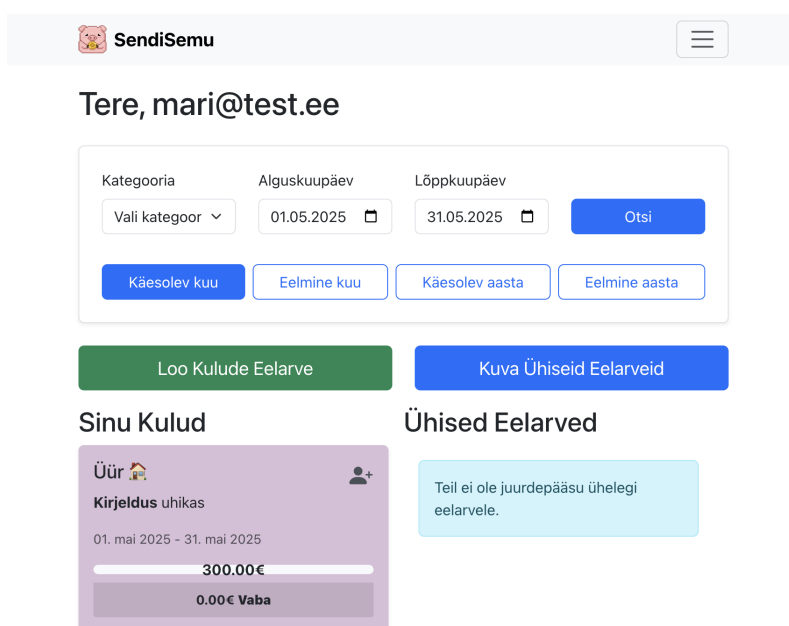
Joonis 55. Rakenduse kuvatõmmis: säästueesmärk täis



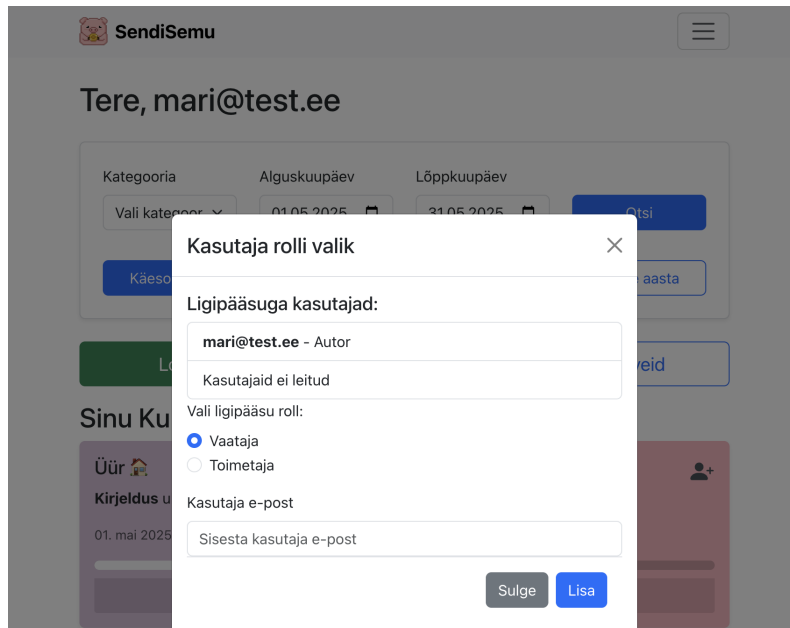
Joonis 56. Rakenduse kuvatõmmis: sissemaksete vaade



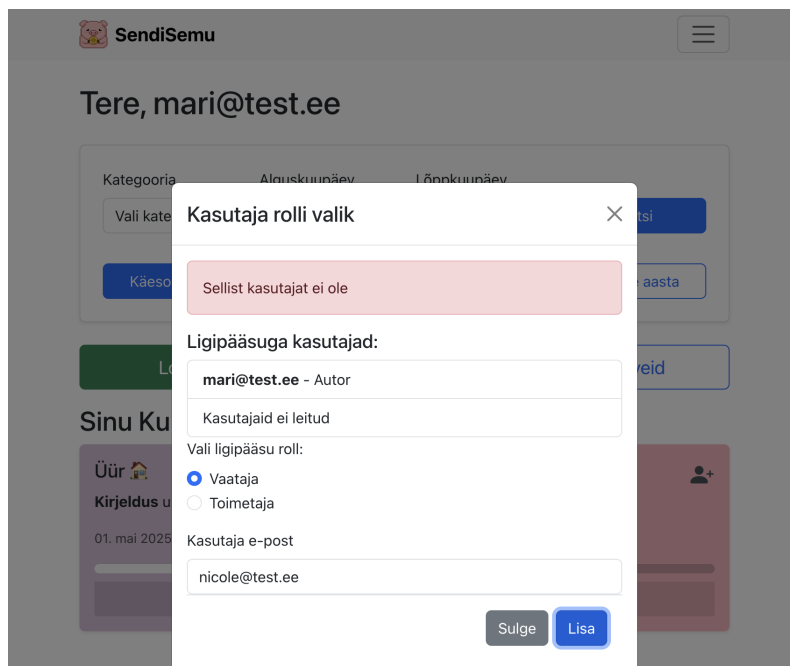
Joonis 57. Rakenduse kuvatõmmis: sissemakse kustutamise hüpikaken



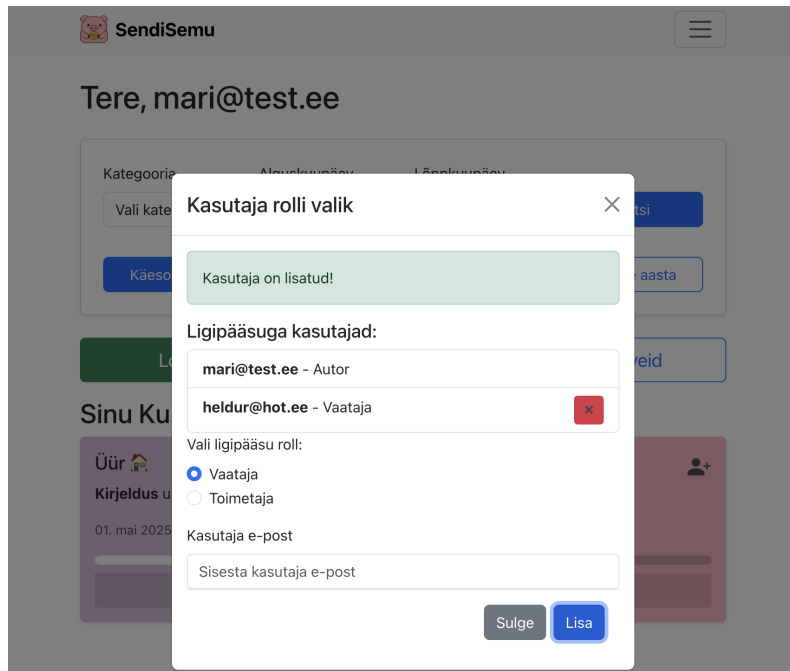
Joonis 58. Rakenduse kuvatõmmis: pole jagatud eelarveid



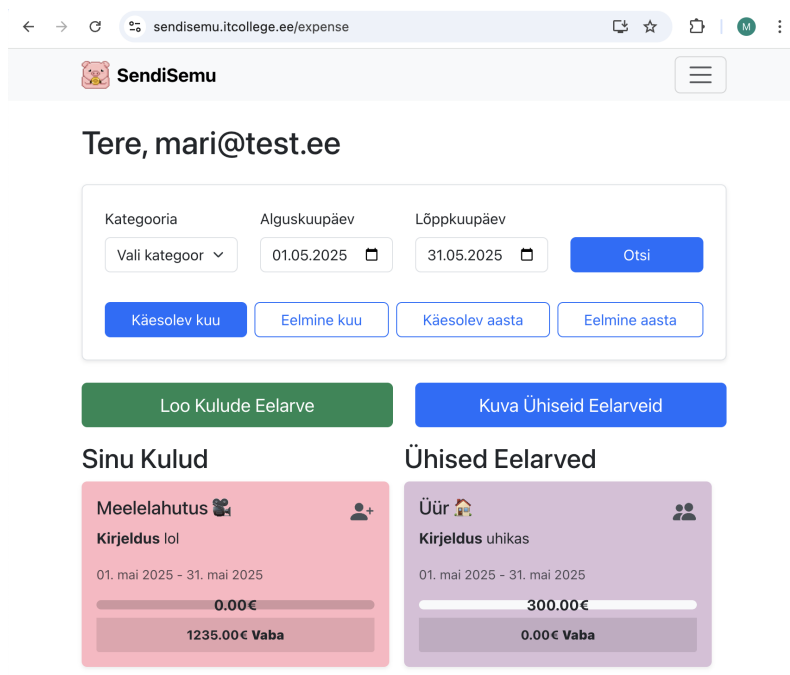
Joonis 59. Rakenduse kuvatõmmis: kasutaja lisamine



Joonis 60. Rakenduse kuvatõmmis: kasutaja lisamine, kasutaja ei eksisteeri



Joonis 61. Rakenduse kuvatõmmis: kasutaja lisatud



Joonis 62. Rakenduse kuvatõmmis: eelarved ligipääsuga

Tere, mari@test.ee

Alguskuupäev: 01.05.2025    Lõppkuupäev: 31.05.2025    Otsi

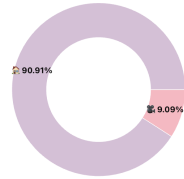
Käesolev kuu    Eelmise kuu    Käesolev aasta    Eelmise aasta

Kulu    Säast

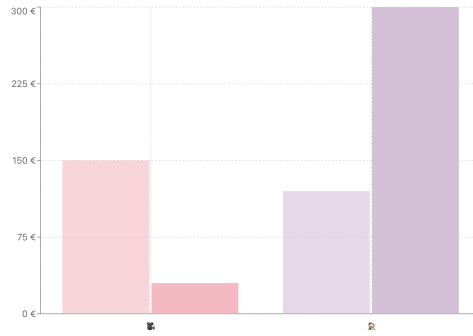
**Eelarve jaotus**  
kategoriate lõikes

Kokku: 330€

- Üür: 300€
- Meelilahnus: 30€

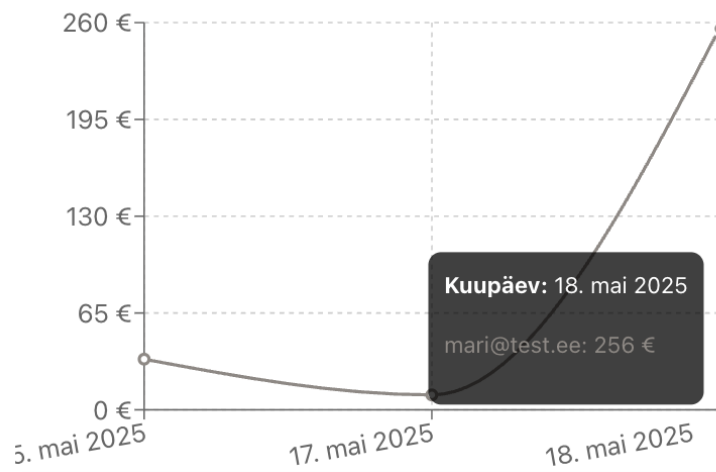


**Eelarve trendide võrdlus**

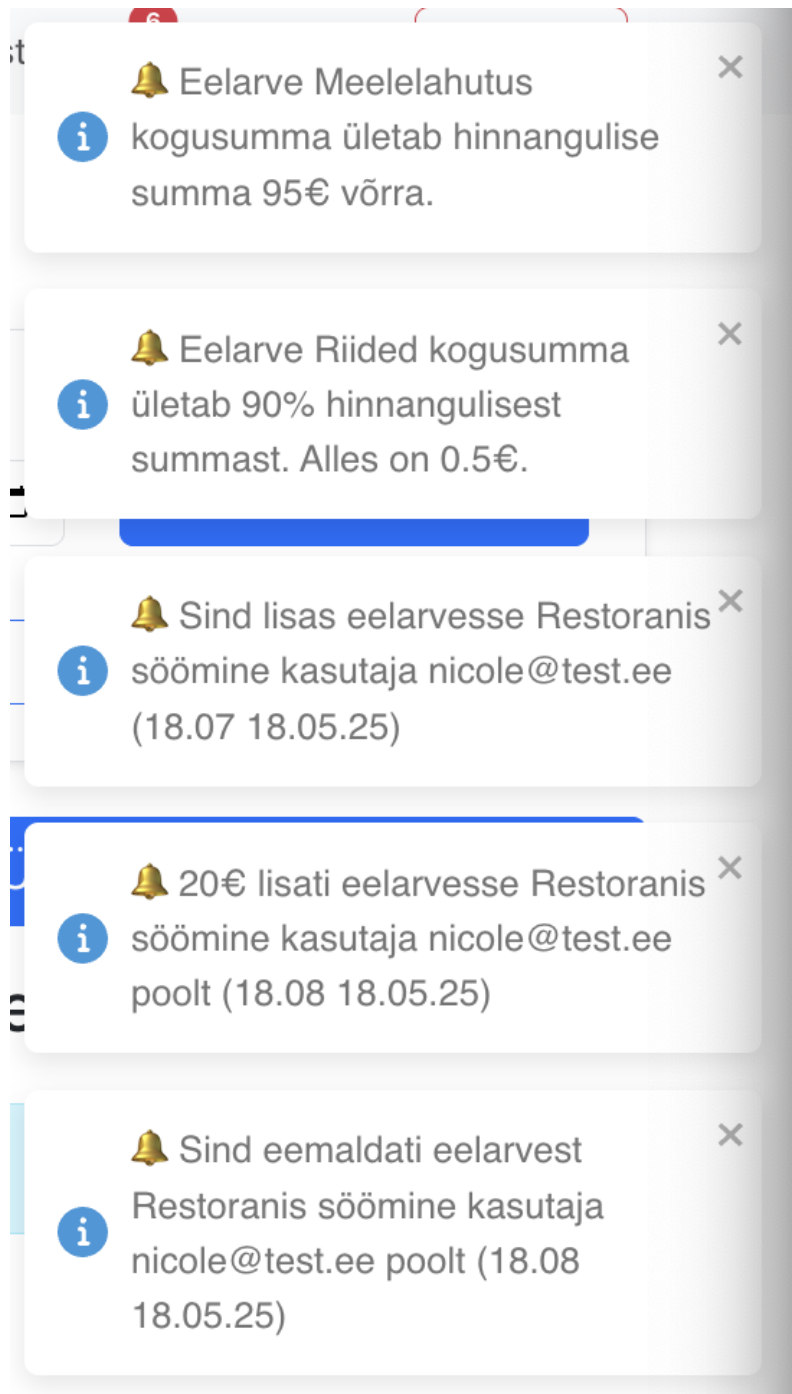


Joonis 63. Rakenduse kuvatõmmis: aruanded pealehel

**Tehingud:**



Joonis 64. Rakenduse kuvatõmmis: sissemaksete joondiagramm



Joonis 65. Rakenduse kuvatõmmis: teavitused