

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Paulius Janušaitis

**Asmeninių finansų planavimo ir valdymo
sistemos kūrimas ir tyrimas**

Magistro darbas

Darbo vadovas

doc. K. Motiejūnas

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Paulius Janušaitis

**Asmeninių finansų planavimo ir valdymo
sistemos kūrimas ir tyrimas**

Magistro darbas

Recenzentas

D. Rubliauskas

2008-05-26

Vadovas

doc. K. Motiejūnas

2008-05-26

Atliko

IFM-2/2 gr. stud.

Paulius Janušaitis

2008 05 26

Kaunas, 2008

Turinys

| | |
|--|----|
| 1. ĮVADAS | 8 |
| 1.1. Dokumento paskirtis | 8 |
| 1.2. Santrauka..... | 8 |
| 2. PROJEKTAVIMO IR PROBLEMINIŲ UŽDAVINIŲ BEI TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ | 9 |
| 2.1. Skyriaus paskirtis | 9 |
| 2.2. Literatūros apžvalga | 9 |
| 2.2.1. Struktūrizuotos sistemos | 9 |
| 2.2.1.1. Atvaizdavimo sluoksnis | 10 |
| 2.2.1.2. Valdymo sluoksnis | 10 |
| 2.2.1.3. Duomenų sluoksnis | 11 |
| 2.2.2. Modelis – Vaizdas – Valdiklis šablonas | 11 |
| 2.2.3. Sistemų saugumas..... | 12 |
| 2.3. Asmeninių finansų planavimo ir valdymo sistemos analizė | 14 |
| 2.3.1. Taikymo sritis, objektas ir problema | 14 |
| 2.3.2. Pradiniai reikalavimai | 15 |
| 2.3.3. Probleminiai uždaviniai | 16 |
| 2.3.4. Probleminių uždavinių sprendimai | 17 |
| 2.3.5. Realizavimo priemonių pasirinkimas | 19 |
| 2.3.6. Egzistuojančių sistemų palyginimo analizė | 21 |
| 2.3.6.1. Esminės problemos | 24 |
| 2.4. Analizės išvados..... | 25 |
| 3. ASMENINIŲ FINANSŲ PLANAVIMO IR VALDYMO SISTEMOS REALIZACIJA | 26 |
| 3.1. Skyriaus paskirtis | 26 |
| 3.2. Projekto tikslai | 26 |
| 3.3. Sprendžiami uždaviniai | 27 |
| 3.4. Priimti techniniai sprendimai | 28 |
| 3.4.1. Diegimo aplinka..... | 28 |
| 3.5. Veiklos kontekstas | 29 |
| 3.5.1. Veiklos padalinimas..... | 29 |
| 3.6. Panaudojimo atvejai | 30 |
| 3.7. Funkciniai reikalavimai..... | 31 |
| 3.8. Nefunkciniai reikalavimai | 32 |
| 3.9. Architektūros specifikacija..... | 34 |

| | | |
|----------|--|----|
| 3.9.1. | Bendras architektūros vaizdas | 34 |
| 3.9.2. | Sistemos architektūra | 34 |
| 3.9.2.1. | Paketų detalizavimas | 35 |
| 3.9.2.2. | Administravimas modulio architektūra | 35 |
| 3.9.2.3. | Finansai modulio architektūra | 36 |
| 3.9.2.4. | Kategorijų valdymas modulio architektūra | 36 |
| 3.9.2.5. | Apmokėjimai modulio architektūra | 37 |
| 3.9.2.6. | Atvaizdavimas ir pateikimas modulio architektūra | 39 |
| 3.10. | Sistemos dinaminis vaizdas | 40 |
| 3.10.1. | Būsenų diagramos | 40 |
| 3.10.2. | Veiklos diagramos | 41 |
| 3.10.3. | Sąveikos diagramos | 43 |
| 3.11. | Duomenų vaizdas | 44 |
| 3.12. | Praktinis darbo rezultatas | 45 |
| 4. | SISTEMOS KOKYBĖS IR NAUDINGUMO TYRIMAS | 46 |
| 4.1. | Kokybės analizės tikslai | 46 |
| 4.2. | Programinės įrangos kokybės vertinimo charakteristikos | 47 |
| 4.3. | Sistemos kokybės tyrimas | 49 |
| 4.4. | Realizuotos sistemos naudingumo įvertinimas | 53 |
| 4.5. | Tolimesnio sistemos tobulinimo ir plėtojimo galimybės | 56 |
| 5. | IŠVADOS | 58 |
| 6. | LITERATŪRA | 60 |
| 7. | TERMINŲ IR SANTRUMPŲ ŽODYNAS | 62 |
| 8. | PRIEDAI | I |

The Research and Development of Personal Finance Planning and Management System

Summary

The objective of this project was to develop a personal finance planning and management system that helps people to manage their personal finance and saves money and time. During its execution, the analysis of design and technology solutions was performed. Further, five basic goals of system realization were formulated, the potential solutions for which were presented in the thesis. The architecture of the developed software is based on the principle of three layer design. During the quality analysis of the system, it was evaluated as a good system except that its maintainability rate was rather low. Its comparison with the alternative systems disclosed it as an average system from the viewpoint of its expediency.

PAVEIKSLIUKŲ SĄRAŠAS

| | |
|---|----|
| <i>2.1 Pav. Projekto sistemos išskaidymas į sluoksnius</i> | 10 |
| <i>2.2 Pav. MVC diagrama</i> | 11 |
| <i>3.1 Pav. Sistemos veiklos kontekstas</i> | 29 |
| <i>3.2 Pav. Panaudojimo atvejų diagrama</i> | 30 |
| <i>3.3 Pav. Sistemos struktūra ir ryšys tarp sluoksnių</i> | 34 |
| <i>3.4 Pav. Sistemos pagrindinių modulių diagrama</i> | 34 |
| <i>3.5 Pav. Paketo administravimas klasių diagrama</i> | 35 |
| <i>3.6 Pav. Paketo Finansai klasių diagrama</i> | 36 |
| <i>3.7 Pav. Paketo Kategorijos klasių diagrama</i> | 37 |
| <i>3.8 Pav. Paketo Apmokėjimai klasių diagrama</i> | 38 |
| <i>3.9 Pav. Paketo Atvaizdavimas ir pateikimas klasių diagrama</i> | 39 |
| <i>3.10 Pav. Vartotojo registravimosi būsenos diagrama</i> | 40 |
| <i>3.11 Pav. Vartotojo prisijungimo būsenos diagrama</i> | 41 |
| <i>3.12 Pav. Išlaidų įvedimo veiklos diagrama</i> | 41 |
| <i>3.13 Pav. Ataskaitų sukūrimo veiklos diagrama</i> | 42 |
| <i>3.14 Pav. Mokesčių tvarkymo sąveikos (seku) diagrama</i> | 43 |
| <i>3.15 Pav. Duomenų bazės modelis</i> | 44 |
| <i>4.1 Pav. Eksperimentinio tyrimo aspektai</i> | 46 |
| <i>4.2 Pav. Kokybės vertinimo modelis</i> | 47 |
| <i>4.3 Pav. Asmeninių finansų tvarkymo sistemų įvertinimų palyginimas</i> | 54 |
| <i>4.4 Pav. Kriterijų svarbos pasiskirstymas</i> | 55 |

LENTELIŲ SĄRAŠAS

| | |
|---|----|
| <i>2.1 Lentelė. Asmeninių finansų tvarkymo programų palyginimas</i> | 23 |
| <i>3.1 Lentelė. Veiklos įvykių sąrašas</i> | 29 |
| <i>3.2 Lentelė. Sistemos panaudojamumo grupės</i> | 31 |
| <i>3.3 Lentelė. Duomenų bazės modelio esybės</i> | 44 |
| <i>4.1 Lentelė. Sistemos kokybės vertinimo charakteristikos</i> | 48 |
| <i>4.2 Lentelė. Asmeninių finansų planavimo ir valdymo sistemos kokybės vertinimas</i> | 50 |
| <i>4.3 Lentelė. Egzistuojančių sistemų naudingumo palyginimas su realizuota sistema</i> | 53 |
| <i>7.1 Lentelė. Terminų ir santrumpų žodynas</i> | 62 |

1. ĮVADAS

1.1. Dokumento paskirtis

Šiame dokumente nagrinėjama tyrimo sritis susijusi su asmeninių finansų planavimo ir valdymo sistemomis. Analitinėje darbo dalyje atsakomi pagrindiniai šio projekto klausimai, pristatomos bendros idėjos ir vėliau apžvelgiama panašių sistemų specifika. Projektinėje dalyje į sistemą pažvelgiama iš architektūros pusės. Sukonkretinami priimti sprendimai ir pateikiamas jų realizacijos aprašymas. Tiriamoje ir eksperimentinėje darbo dalyje atliekamas sistemos tyrimas. Įvertinama sistemos kokybė ir pateikiamas apibendrintas sistemos vaizdas kartu su rekomenduojamais patobulinimais. Išskiriamos silpnosios galutinio produkto dalys ir pateikiami pasiūlymai jų tobulinimui.

1.2. Santrauka

Populiarėjant informacinėms technologijoms, plintant interneto ryšiui bei didėjant asmeninių kompiuterių skaičiui [1], atsiranda vis daugiau žmonių, norinčių pasinaudoti įvairiomis elektroninėmis sistemomis, kurios palengvina kasdieninę veiklą. Taipogi, pastaruoju metu keičiasi ir šalies gyventojų požiūris į skolinimąsi, taupymą. Visuomenė paprastai pradeda intensyviau taupyti ir kontroliuoti išlaidas įvairių ekonominių (ir šalies, ir asmeninių) krizių laikotarpiu, kai atsiranda didesnis neužtikrintumas dėl rytdienos. Matome, kad asmeninių finansų planavimo ir valdymo sistema yra aktuali šiandieniniame gyvenime.

Šiam tikslui įgyvendinti ir buvo sukurta asmeninių finansų planavimo ir valdymo sistema, kuri padėtų vartotojui išspręsti asmeninių finansų planavimo ir valdymo problemas, padėtų taupyti laiką ir pinigus.

Šiuo metu pasaulyje yra sukurta nemažai asmeninių finansų tvarkymui skirtų sistemų. Išanalizavus jas nustatyta, jog jose esančios funkcijos nepakankamai pritaikytos Lietuvos rinkai. Todėl, norėdami palengvinti ir pritaikyti produktą Lietuvos rinkai, planuojame sukurti tokią asmeninių finansų planavimo ir valdymo sistemą, kuri būtų integruota ir susieta su mūsų šalyje egzistuojančiais mokesčių tvarkymo principais.

2. PROJEKTAVIMO IR PROBLEMINIŲ UŽDAVINIŲ BEI TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ

2.1. Skyriaus paskirtis

Šiame skyriuje atliekama analizės ir projektavimo metodų analizė bei aptariamos pasirinktos technologinės priemonės. Apibrėžiama tyrimo sritis, analizuojamas objektas bei įvardijamos problemos, kurias turėtų išspręsti kuriamas produktas. Atsakomi pagrindiniai šio projekto klausimai, pristatomos bendros idėjos ir vėliau apžvelgiama panašių sistemų specifiška. Suformuluojamas uždavinys, projektavimo metu iškilusios projektavimo ar programavimo inžinerijos problemos. Pateikiama galimų sprendimų apžvalga, priimto ir realizuoto sprendimo pagrindimas bei įvertinimas.

2.2. Literatūros apžvalga

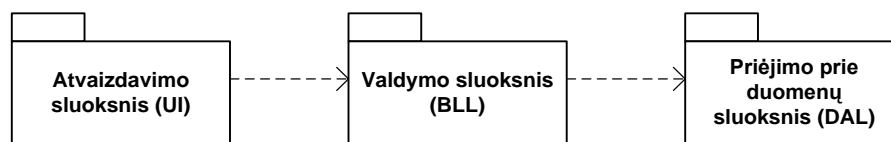
2.2.1. Struktūrizuotos sistemos

Projektuojant sudėtingas verslo lygio sistemas, sudarytas iš didelio sistemos komponentų kiekio, būtina atsižvelgti į šių komponentų išdėstymą abstrakčiuose sluoksniuose. Problema iškyla tada, kai siekiama struktūrizuoti sistemą, siekiant, kad ji atitiktų tokius svarbius sistemos reikalavimus kaip: palaikomumas, panaudojamumas, keičiamumas, patikimumas ir saugumas. Taip iškeliamas pagrindinis klausimas: kaip tokią sistemą struktūrizuoti.

Atsakymas į šį klausimą gali būti įvairus. Struktūrizuojant sistemą reikia įvertinti keletą aplinkos veiksnių:

- Pakeitimų lokalizavimas vienoje sistemos dalyje sumažina riziką, kad pakeitimai bus iššaukti ir kitose dalyse. Taip sumažinamas darbas reikalingas testavimui ir klaidų ištaisymui, taip pat palengvina sistemos prižiūrimumą ir pagerina visą sistemos lankstumą;
- Problemos atskyrimas tarp komponentų (pavyzdžiui, atskiriant vartotojo sąsają nuo biznio logikos, ir šią nuo duomenų bazės) padidina sistemos lankstumą, prižiūrimumą ir keičiamumą;
- Komponentai turėtų būti pakartotinai panaudojami;
- Panašūs komponentai turėtų būti „surišti“ tarpusavyje;
- Nepanašūs komponentai turėtų būti „nesurišti“;
- Norint sukurti saugią ir prieinamą internetinę sistemą, pirmiausia reikia ją sudalinti į keletą fizinių lygių. Taip užtikrinant, kad dalis sistemos bus neprieinama, o kita dalis galės būti laisvai pasiekama internetu [2].

Taigi, pirmiausia reikia išskirti komponentus į sluoksnius. Komponentai šiame sluoksnyje turėtų būti vienas į kitą panašūs, t.y. tampriai surišti. Kiekvienas sluoksnis turėtų skirtis nuo šalia jo einančios sluoksnio. Šiuo atžvilgiu, internetinės (angl. *web*) sistemos dažniausiai naudoja trijų sluoksnių principą, kai atskiriami vartotojo sąsajos, programos logikos ir duomenų priėjimo sluoksniai (2.1 Pav.).



2.1 Pav. Projekto sistemos išskaidymas į sluoksnius

Programa, kuri vadovaujasi tokia architektūra turi vaizdinius, kurie atsako tik už puslapių atvaizdavimą. Šie puslapiai vykdo, valdymo (angl. *Business Logic Layer – BLL*) sluoksnyje, esančią programos veiksmų logiką. Šis sluoksnis, gali kreiptis į priėjimo prie duomenų sluoksnį (angl. *Data Access Layer – DAL*) tam, kad gautų reikiamus duomenis. Visa informacija gauta ir apdorota iš šių sluoksnių yra grąžinama į atvaizdavimo sluoksnį (angl. *User Interface - UI*).

2.2.1.1. Atvaizdavimo sluoksnis

Jei sistema yra sukurta naudojantis internetine sąsaja, tai tokia vartotojo sąsaja – internetinių puslapių rinkinys. Šie puslapiai atvaizduoja vartotojui sistemoje esančią informaciją bei perteikia sistemos teikiamas funkcijas. Atvaizdavimo sluoksnis yra atsakingas už bendrą atvaizdavimo struktūrą. Jis formuoja ir pateikia visą sistemos informaciją vartotojui. Taip pat, atvaizdavimo sluoksnis atsakingas už puslapių generavimą ir dinamiškai generuojamą turinį. Tai yra, jis bendradarbiaudamas su kitais lygiais gauna tam tikrą informaciją arba duomenis, ir pagal suprogramuotą struktūrą sugeneruoja informacijos pateikimo formą. Kita šio lygio funkcija yra: apdoroti gaunamą iš vartotojo informaciją (rasti įvestus duomenis ir t.t.) ir perduoti ją valdymo sluoksniui.

2.2.1.2. Valdymo sluoksnis

Visos funkcijos, kurios turi būti realizuotos programoje, turi tam tikrus loginius scenarijus ir tam tikrus apribojimus arba vykdymo sąlygas. Visą šį veiklos priklausomumą galima vadinti veiklos taisyklėmis. Valdymo sluoksnis kaip tik ir yra atsakingas už visą veiklos logiką realizuotą programoje. Šis sluoksnis yra valdantis sluoksnis, kuris apima:

- Visus reikalingus skaičiavimus;
- Darbo srautų valdymą (taip pat sesijos duomenų stebėjimą ir valdymą);
- Visų atvaizdavimo sluoksnio duomenų užklausų apdorojimą.

2.2.1.3. Duomenų sluoksnis

Priėjimo prie duomenų modulio paskirtis yra aprūpinti valdymo sluoksnį reikalingais duomenimis bei perkelti iš jo gautus duomenis į duomenų bazę. Kitaip tariant, šis sluoksnis yra tarpininkas tarp fizinės duomenų bazės ir programos, o konkrečiu atveju valdymo sluoksnio.

Šis sluoksnis atsako už ryšį su duomenų baze. Užtikrina ryšio palaikymą, nutraukimą ir duomenų gavimą iš duomenų bazės.

Šiame sluoksnyje yra tik minimalūs patikrinimai ar veiklos logikos elementai, kadangi visą tai turi užtikrinti valdymo sluoksnis. Čia išlieka tik tokie veiklos logikos elementai, kaip:

- objekto validavimo patikrinimas;
- išorinio rakto ryšio apribojimai.

2.2.2. Modelis – Vaizdas – Valdiklis šablonas

Kurdami ir projektuodami asmeninių finansų planavimo ir valdymo sistemą, išnagrinėjome aukščiau aprašytą struktūrizuotą architektūrą, ir savo darbe pritaikėme Modelis-Vaizdas-Valdiklis (angl. *Model-View-Controller* – *MVC*) programavimo šabloną (2.2 Pav.).

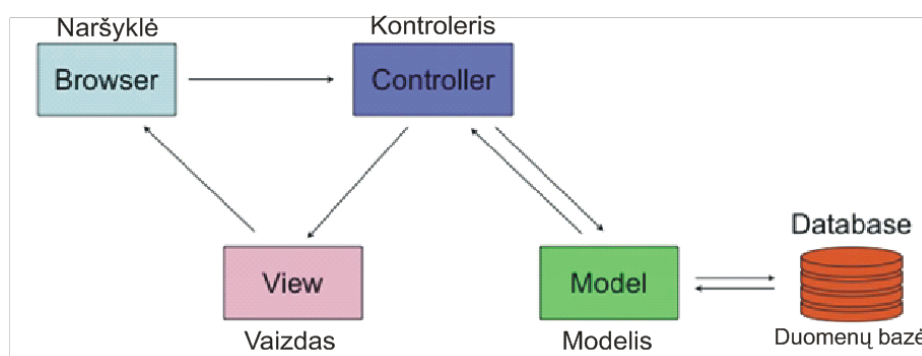
MVC veikimas yra paremtas problemos atskyrimu (angl. *separation of concerns* – *SoC*) principu [3]. Kaip ir minėjome ankstesniame 2.2.1 skyrelyje, tai reiškia, kad programa yra išskaidoma į tokias dalis, kad jų funkcionalumas bei poreikiai nepersidengtų.

Pačiame MVC šis principas pasireiškia logikos, prezentacijos ir turinio atskyrimu. Trys dalys atsakingos už savo veiklos sritį:

Modelis (Model) - tam tikros srities duomenų valdymo/pateikimo vienetas.

Vaizdas (View) - suteikia programos modeliui išvaizdą, tinkamą vartotojui suprasti (formos, sąrašai). Web programose, tai ne kas daugiau, negu kad sugeneruotas HTML kodas.

Valdiklis (Controller) - atsako už modelio valdymą bei vaizdų pateikimus pagal atitinkamus vartotojo veiksmus: mygtuko paspaudimas, teksto įvedimas ir pan.



2.2 Pav. MVC diagrama

Kaip matome iš aukščiau pateikto paveikslėlio (2.2 Pav.), vartotojo naršyklė pirmiausia kreipiasi į valdiklį (kontrolerį), šis savo ruožtu, jei reikia, gali pasiimti reikalingus duomenis modelio pagalba, kuris visada kreipiasi į duomenų bazę. Valdiklis gavęs duomenis arba jų net neprašęs gali iškart atvaizduoti puslapį. Šis sugeneruotas puslapis ir yra atvaizduojamas vartotojo naršyklėje.

Taigi, kaip matome, šis šablonas gali būti naudojamas sistemose su sudėtingom vartotojo sąsajom, kurios gali naudoti ir atvaizduoti tuos pačius duomenų masyvus skirtingais būdais ir skirtingais vaizdais. Taip pat šis šablonas tinkamas griežtam atvaizdavimo sluoksnio atskyrimui nuo valdymo sluoksnio [4].

2.2.3. Sistemų saugumas

Sistemų saugumas yra vienas svarbiausių klausimų, kurių turi išspręsti šiuolaikinė įmonė ir sistemos kūrėjai. Jis apima jų konfidencialumo, prieinamumo ir vientisumo užtikrinimą. Konfidencialumas reiškia, kad duomenys bus prieinami tik tiems vartotojams, kuriems suteikta tokia teisė. Vientisumas užtikrina, kad saugoma informacija nebus pakeista (iš dalies ar visai sugadinta) nesankcionuotu būdu. Duomenų prieinamumas garantuoja, kad reikiami IS resursai reikiamu metu yra prieinami sankcionuotam vartotojui.

Informacija yra vienas didžiausių jos turtų, tačiau jos saugumas sunkiai apibrėžiamas, o pasikėsinimas į jį nelengvai pastebimas ir net įrodomas. Kol kas nėra sureguliuoti ir juridiniai klausimai, reglamentuojantys bausmes už nusikaltimus elektroninėje erdvėje, todėl visa atsakomybė už turimą nuosavybę tenka pačiai kompanijai ar sistemos kūrėjui.

Todėl, siekiant išvengti saugumo problemų, arba bent dalinai jas minimizuoti, būtina sistemos projektavimo metu į tai atsižvelgti, ir skirti tam gana didelį dėmesį. Todėl ir mes, analizuodami asmeninių finansų valdymo sistemas, būtinai į tai atsižvelgsime.

Atsižvelgiant į tai, kad kiekviena sistema gali turėti tik jei būdingų problemų, būtų galima įvardinti ir keletą bendrinių saugumo problemų [5]. Pagrindinės probleminės sritys, kuriose būtų galima išvelgti saugumo atsiradimo problemas, yra šios:

- Programavimo karkaso „sumanumas“;
- Saugaus programavimo principai;
- Rizikos mažinimo modeliavimas;
- E-komercijos apmokėjimų valdymas;
- Apgaudinėjimas (angl. *Phishing*);
- Web servisai;
- Ajax ir kitos „turtingos“ sąsajos technologijos;
- Autentifikacija ir autorizacija;

- Sesijų valdymas;
- Duomenų validavimas;
- Komandų įterpimas;
- Buferio persipildymas;
- Klaidų valdymas ir auditas;
- Failų sistema;
- Administravimo sąsaja;
- Sistemos kokybės priežiūra;
- Konfigūravimas;
- Diegimas;
- Priežiūra.

Kai kurias, šioms grupėms būdingas problemas, padeda išspręsti pasirinktos pažangios projekto kokybės valdymo priemonės, taip pat įvairių ISO standartų reikalavimų laikymasis bei CMM metodų taikymas. CMM modelis nustato bendruosius PĮ kūrimo ir priežiūros projektų planavimo, projektavimo ir valdymo metodus. Įmonė, įgyvendinusi CMM metodus, įgyja gebėjimą gerai valdyti ir nuolat gerinti PĮ kūrimo ir priežiūros procesų bei sukurtų produktų kokybę.

Visas kitas čia paminėtas problemines sritis, būtina įvertinti sistemos projektavimo, kūrimo ir diegimo metu. Vėlesnėse stadijose aptiktos klaidos gali labai brangiai kainuoti. Pavyzdžiui, santykiniai klaidos ištaisymo kaštai gali būti tokie: reikalavimuose 1x, specifikacijoje 2x, planavime 3x, projektavime 4x, realizuojant 10x, integruojant 52x, prižiūrint 368x karto didesni [6].

2.3. Asmeninių finansų planavimo ir valdymo sistemos analizė

2.3.1. Taikymo sritis, objektas ir problema

Taikymo sritis – tai asmeninių finansų planavimo ir valdymo sistemos sprendimas vartotojams, kuris padėtų vartotojui išspręsti asmeninių finansų planavimo ir valdymo problemas, padėtų taupyti laiką ir pinigus.

Šios srities objektas ir problema apima: namų buhalteriją, asmeninių finansų valdymą bei asmeninių finansų planavimo ir valdymo sistemas. Todėl, detaliau panagrinėsime kiekvieną iš jų.

- **Namų buhalterija**

Pabrangę produktai bei paslaugos varo į neviltį. Kad nejaustumė pinigų trūkumo, turime kontroliuoti išlaidas. Galbūt didžiąją pinigų dalį išleidžiame nelabai reikalingiems dalykams. Todėl namų biudžetą nuo bankroto apsaugos specialios išlaidų ir pajamų apskaitos programos.

Bankrotas gali gręsti ir mums. Ypač jei neatsakingai žiūrėsime į pajamų ir išlaidų balansą. Juk tikrai pabosta paskutinę savaitę prieš stipendiją ar algą vien tik keptas bulves su aliejumi valgyti. Taip neatsitiks, jei savo išlaidas planuosime ir kontroliuosime. Dažnai patys stebimės, kad net nežinome, kur dingę turėti pinigai. Specialios namų biudžeto planavimo programos padės pamatyti, kam išleidžiame daugiausiai pinigų. Gali būti, kad daugiausiai pinigų „suvalgo“ visai nebūtinoms pramogoms ar pirkiniams [7].

- **Asmeninių finansų valdymas**

Bendrajai prasme finansų valdymas šiuolaikiniame pasaulyje nėra jau toks paprastas ir lengvas dalykas. Reikia gana gerai išmanyti šią sritį, kad galėtum gerai valdyti ir kontroliuoti tam tikrus finansus. Todėl dauguma įmonių ir organizacijų samdo tokiam darbui šios srities specialistus [8].

Tačiau, mūsų atžvilgiu, įsigilinti į paprastų vartotojų finansų valdymo subtilybes yra gan paprasčiau. Mes kasdien susiduriame su paprasčiausiais pinigų judėjimo srautais: išleidžiame pinigus, gauname pajamas, skolinamės, peržiūrime savo balansą, planuojame pirkinius ir kt.

Dauguma žmonių nelinkę per daug dėmesio skirti savo finansų valdymui ir planavimui, bet yra ir tokių, kuriems rūpi žinoti kur, kam ir kaip iškeliauja jų pinigai. Taigi, tokiems klausimams atsakyti ir asmeniniams finansams tvarkyti, naudojamos įvairios asmeninių finansų planavimo ir valdymo priemonės, apie kurias detaliau pašnekėsime 2.3.6 skyrelyje „Egzistuojančių sistemų palyginimo analizė“.

- **Asmeninių finansų planavimo ir valdymo sistemos**

Asmeninių finansų planavimo ir valdymo sistemos pagrindinė užduotis – padėti vartotojui tvarkyti asmeninius finansus, įvesti ir peržiūrėti pajamas, išlaidas, mokesčius; padėti stebėti jų pasiskirstymo srautus, generuoti ataskaitas, pasiskaičiuoti mokesčių dydžius ir kt. Naudojantis asmeninių finansų programine įranga, asmeninių finansų valdymas perkeliamas į visai kitą lygį, tačiau norint išnaudoti visas galimybes, kurias teikia šios programos, yra labai svarbu pasirinkti tokią programinę įrangą, kuri atitinka jūsų asmeninius poreikius.

Pasaulio rinkoje egzistuoja gana nemažai įvairių užsieniškų paketų, kurie palengvina asmeninių finansų planavimą ir valdymą. Tačiau, jie yra menkai arba išvis nepritaikyti Lietuvos rinkai. Didelė dalis programų bei jų funkcijų yra pritaikytos vakarų rinkoms, o ir finansų valdymo įpročiai skiriasi nuo egzistuojančių Lietuvoje. Taigi, kuriant programinę įrangą būtina orientuotis į Lietuvos rinkos vartotojus ir jų asmeninių finansų planavimo ir valdymo principus bei įpročius.

2.3.2. Pradiniai reikalavimai

Toliau, šiame skyriuje, pateikiami pradiniai funkciniai ir nefunkciniai užsakovo reikalavimai ir pageidavimai kuriamai sistemai. Šie reikalavimai įtakos kuriamos sistemos architektūros tipo pasirinkimą, funkcionalumą, vartotojo sąsajos pasirinkimą bei kt.

Pageidavimai kuriamai sistemai (nefunkciniai reikalavimai):

- Daugiavartojiškumas;
- Sistemos pasiekiamumas internetu (on-line);
- Funkcionalumas ir paprastumas;
- Ruby on Rails karkaso¹ panaudojamas.

Šio projekto užsakovas – Univelas įmonė, kurios tikslas buvo sukurti kažką naujo šioje srityje. Kaip matyti iš aukščiau pateiktų reikalavimų, o ypač iš paskutiniojo, tikimasi panaudoti pakankamai naują kalbą dinamiškai interneto svetainei sukurti, kuria galėtų naudotis daug interneto vartotojų. Iš čia ir seka kitas reikalavimas, t.y sistemos prieinamumas internetu (on-line), kad sistema tenkintų daugumos vartotojų poreikius, bet tuo pačiu būtų ir kuo paprastesnė savo funkcionalumu.

Taip pat, užsakovas pateikė sekančius funkcinis reikalavimus sistemai:

- Automatizuoti finansų planavimą ir valdymą, sumažinant reikalingų pasikartojančių duomenų įvedimą;

¹ Ruby on Rails programavimo kalbos karkasas

- Vartotojų administravimo posistemės sukūrimas ir integravimas su būsima pagrindine finansų svetaine;
- Asmeninių finansų kategorijų valdymas:
 - a) pajamų/išlaidų grupavimas;
 - b) mokesčių ir atsiskaitymų grupavimas – galimybė pačiam vartotojui susikurti norimas mokesčių grupes/kategorijas ir pagal jas vesti atsiskaitymus;
 - c) patogus ir greitas kategorijų valdymas – naujų įrašų sukūrimui, redagavimui bei rūšiavimui;
 - d) kategorijų grafinis atvaizdavimas „pyrago“ (angl. *pie*) forma.
- Pajamų/išlaidų ir mokesčių valdymas:
 - a) galimybė įvesti bei tvarkyti pajamas ir išlaidas;
 - b) mokesčių ir atsiskaitymų valdymo galimybės;
 - c) atsiskaitymo knygelės tvarkymo galimybės;
 - d) grafinis duomenų pateikimas
- Duomenų išsaugojimo/išvedimo funkcionalumas.

Detaliau šiuos reikalavimus ir su jais susijusius probleminius uždavinius aptarsime sekančiame skyrelyje.

2.3.3. Probleminiai uždaviniai

Analizuodami 2.3.2 skyrelyje pateiktus užsakovo reikalavimus, automatiškai iškeliamo sau ir sistemos realizavimo probleminius uždavinius. Šiuos uždavinius suskirstėme į tokias penkias kategorijas:

1. Universalios ir paprastos sistemos struktūros sukūrimas

Čia atsakomi tokie pagrindiniai klausimai kaip:

- Kokius įrankius naudoti pakartotiniams sistemos elementams kurti;
- Ar galima panaudoti tam tikrą karkasą vartotojo sąsajos elementams kurti.

2. Finansų valdymas

Atsako į klausimą: kokie finansų valdymo principai.

3. Informacijos įvedimas/duomenų dubliavimas

Atsako į tokius klausimus:

- Koks turėtų būti informacijos įvedimas ir atvaizdavimas, kaip sumažinti reikiamų įvesti duomenų kiekį;

- Ar dažnai reikės įvedinėti pasikartojančią informaciją.

4. Vartotojų valdymas ir saugumas

Atsako į klausimą: kokios priemonės reikalingos vartotojų valdymui ir saugumui užtikrinti;

5. Integracija su administravimo posisteme

Kelia klausimą: kaip greitai ir be sutrikimų integruoti ir vėliau modifikuoti administravimo ir sistemos sąsajas.

Siekiant sėkmingai išspręsti šiuos uždavinius, analizės ir projektavimo metu turės būti atsakyta į visus šiuos iškeltus klausimus. Todėl, pradedant nuo sekančio skyrelio, ir pabandydysime detaliau aptarti šių probleminių uždavinių sprendimus.

2.3.4. Probleminių uždavinių sprendimai

Čia aptariami 2.3.3 skyrelyje iškelti probleminiai uždavinių sprendimai.

- **Universalios ir paprastos sistemos struktūros sukūrimas**

Išanalizavus struktūrizuotų sistemų ir MVC šablono modelius, buvo pasirinkta programavimo aplinka, kuri detaliau yra aptarta „2.3.5 Realizavimo priemonių pasirinkimas“ skyrelyje.

Detaliau paanalizavus įvairius universalios sistemos struktūros generavimo karkasus bei standartinius struktūros kūrimo įrankius, buvo iškelta tokia pagrindinė problema: skirtingi įrankiai ir karkasai generuoja tik riboto funkcionalumo struktūras, tad kyla klausimas, dėl tinkamo jų pasirinkimo.

Šios problemos sprendimui sistemos struktūrą galima suskirstyti į lygius pagal standartinį „Ruby on Rails“ karkaso naudojamą MVC modelį. Taip supaprastinant sistemos struktūrą. O siekiant universaliai vartotojo sąsajai sukurti, panaudojamas standartines galimybes išplečiantis, papildomas išorinis sąsajos karkasas.

- **Finansų valdymas**

- a) **Balansas**

Pvz.: kai pirkiniai, įvairūs apmokėjimai ir kita informacija yra įvedami į tikrinamą sąskaitą, asmeninių finansų programa už jus suskaičiuoja tikslų tos sąskaitos balansą. Naudojant tokią programinę įrangą, išvengiama matematinių klaidų bei stebimas vedamos sąskaitos einamas balansas [9].

Šiai vartotojo problemai išspręsti, reikia sukurti: įvestų vartotojų duomenų, pajamų ir išlaidų balanso, skaičiavimo funkcijas. Taip kiekviename vartotojo sąsajos

lange rodant atitinkamą mėnesio balansą, bet ir paliekant vartotojui galimybę peržiūrėti buvusių mėnesių balansus.

b) Mokesčių kategorijos

Vartotojas gali įtraukti, atnaujinti ar pašalinti įvairius mokesčius ir jų kategorijas. Mokesčių kategorijos yra įtrauktos daugumoje finansų programų, bet paliekama galimybė susikurti bet kokią kitą kategoriją, kurios nėra programoje.

Taigi, reikia išspręsti kategorijų valdymo problemą. Šios problemos sprendimui, galima pasitelkti vartotojų administravimo posistemę, kurios pagalba, sukuriant naują vartotoją, prie jau esamos jo informacijos būtų sukuriamos įvairios standartinės finansų sistemos kategorijos.

c) Pajamų-Išlaidų analizė

Skirta padėti vartotojui peržiūrėti įvairias pajamų ir išlaidų kategorijas. Vartotojo atžvilgiu, tai pagrindinis įrankis, kuris padeda identifikuoti didžiausias išlaidų sritis. Todėl būtina išspręsti šį klausimą, įvedant sistemoje duomenų atvaizdavimo ir pateikimo funkcijas. Pasinaudodamas šiomis funkcijomis, vartotojas galės analizuoti, kiek pinigų yra išleidžiama tam tikriems dalykams, grupuojant pajamas ir išlaidas į tam tikras grupes.

Šiai problemai išspręsti gali pasitarnauti lengvas ir patogus naudoti grafinis duomenų atvaizdavimo įrankis.

a) Ataskaitų pateikimas

Vartotojui reikalinga, kad įvairiais pjūviais būtų pateikiamos ataskaitos, tai gali būti jau ir minėtas grafinis atvaizdavimas. Ataskaitos gali būti: pajamų-išlaidų, mokesčių, atsiskaitymo knygelės mėnesinės, metinės ir kitos ataskaitos. Todėl, sistema turi išanalizuoti turimus duomenis ir pateikti vartotojui norimą ataskaitą.

Šios problemos realizacijos sprendimui, galima panaudoti lankstų PDF ataskaitų generavimo įrankį, kuris palaikytų ir įvairius ataskaitų atvaizdavimo stilius.

- **Informacijos įvedimas/duomenų dubliavimas**

Kuriant asmeninių finansų planavimo ir valdymo sistemą, reikia atsižvelgti į funkcijas, kurias jau atlieka bankai. Pavyzdžiui, vartotojas norėdamas įvesti į finansų programą duomenis iš savo banko, transakcijų taip lengvai negalės importuoti iš internetinės bankininkystės sistemų. Todėl, vartotojui papildomai dar tenka įvedinėti duomenis ir į finansų sistemą, taip gaunamas duomenų dubliavimas, dėl kurio nukenčia sistemos funkcionalumas. Tas pats galioja ir su mokesčių apmokėjimais.

Šios problemos galutinai tikriausiai ir nepavyktų išspręsti, nebent išsivysčius sistemai, bandyti pasirašyti sutartis su bankais, bet tai šiuo metu mažai tikėtinas variantas. Todėl, pradžioje bus siekiama kaip tik įmanoma sumažinti reikalingų įvedamų duomenų kiekį.

- **Vartotojų valdymas ir saugumas**

Vartotojų valdymui ir duomenų saugumui užtikrinti turi būti skirtas ypatingas dėmesys. Sistema gali saugoti svarbius vartotojo duomenis, tai gali būti: sąskaitų ar kredito kortelių numeriai, įvairių mokesčių ar pervedimo duomenys, turto sąrašai, asmeninių finansų duomenys ir kita konfidencialia informacija. Todėl būtina vartotojų saugumui skirti tikrai nemažą dėmesį.

Sistemos svetainė turi būti patalpinta gana rimtoje serverius prižiūrinčioje organizacijoje. Sistemos svetainės administravimui turi būti suteikti visi reikalingi įrankiai. Visa prieiga prie svarbios informacijos turi būti apsaugota prisijungimo vardu ir slaptažodžiu, šie duomenys galėtų būti saugomi vartotojo kompiuteryje užkoduotų sausainėlių (angl. *cookies*) pavidalu.

Vartotojų prisijungimo prie sistemos bei perduodami svarbūs duomenys internetu galėtų būti apsaugomi juos šifruojant atitinkamais raktais: SSL, DES, RSA ir t.t. [10]

- **Integracija su administravimo posisteme**

Išanalizavimus vartotojų valdymo ir saugumo problemas, pagal užsakovo pradinius reikalavimus buvo nuspręsta iš pradžių sukurti vartotojo administravimo posistemę. Dėl to vėliau pasikeitus reikalavimas, galėtų kilti integravimo su pagrindine finansų sistema problemos. Tam išvengti reikėtų laikytis sekančių taisyklių:

- a) administravimo posistemė, turėtų būti universali ir lengvai pritaikoma;
- b) finansų sistemos ir administravimo posistemės turėtų būti parašytos ta pačia programavimo kalba.

2.3.5. Realizavimo priemonių pasirinkimas

Vis daugiau programinės įrangos kuriama interneto paslaugų pavidale. Tokios paslaugos plačiai naudojamos bankininkystėje, finansų, draudimo, turizmo industrijoje, elektroniniame versle ir pan. Prognozuojama, kad ateities programos jau neturės didelių sukompiliuotų paleidžiamųjų kodų (*EXE, DLL ir kt*). Programos bus padalintos į mažesnes paslaugas, kurias paprasčiau sukurti, palaikyti ir valdyti. Dėl pakankamai naujų galimybių ir greito sistemų kūrimo, internetinės realizacijos yra labai svarbios programinės įrangos technologijų pasaulyje.

Išanalizavę sistemai keliamus reikalavimus galime iš anksto nuspėti, kad jos kūrimui reikės technologijų, kurios leistų realizuoti internetinį puslapį.

Šiuo metu pasaulyje, internetinių sistemų kūrimui, naudojamos trys pagrindinės platformos: *PHP*, *.NET* ir *Java*. Tačiau, atsiranda ir kitų kalbų, kurios evoliucionuoja, iš ankstesnių kalbų perima geriausias savybes, ir taip mažais žingsniais skverbiasi į šią rinką. Todėl, toliau pabandydysime plačiau panagrinėsime vieną iš tokių kalbų „Ruby on Rails“.

Ruby programavimo kalba atsirado palyginti neseniai, todėl šioje srityje susiduriama su tam tikromis problemomis, būdingomis naujoms technologijoms: idėjų plėtojimas vyksta technologiniame lygmenyje, o modeliavimo bei projektavimo metodai atsiranda vėliau.

Ruby yra programavimo kalba, savo galimybėmis panaši į populiarias PHP bei Perl programavimo kalbas. Rails yra programinis karkasas (angl. *framework*), skirtas paprastesniam ir greitesniam internetinių sistemų pagal MVC (liet. *Modelis-Vaizdas-Valdiklis*) modelį kūrimui [11].

Ruby on Rails – tai pastaruoju metu vis dažniau naudojamas derinys „Web 2.0“ svetainių kūrimui, ypač pamėgtas programuotojų dėl keleto priežasčių:

- *programavimo greitis* – su „Ruby on Rails“ norimas rezultatas pasiekiamas keletą kartų greičiau nei programuojant kitomis kalbomis;
- *mažiau kodo eilučių* – o tai reiškia mažiau klaidų kuriamose sistemose, nes didelė dalis dažniausiai naudojamo kodo jau suprogramuota, ištestuota ir įtraukta į „Rails karkasą“;
- *elegantiškas programų kodas* – Ruby sintaksės bei MVC modelio dėka programos yra lengviau skaitomos ir suprantamos.

Skirtumai tarp „Ruby on Rails“ ir tradicinių svetainių kūrimui naudojamų programavimo kalbų (*PHP*, *Perl*) lemia ne tik kitokį – greitesnį, paprastesnį ir malonesnį – Web sistemų kūrimą, bet ir kiek specifinį internetinių sistemų talpinimą.

Dėl šių, aukščiau išvardintų priežasčių, ir buvo nuspręsta pasirinkti „Ruby on Rails“ platformą. Taip pat, apsisprendimą nulėmė ir kelios kitos priežastys. Pirmiausia tai, kad sistema naudosis MySQL DBVS, o „Ruby on Rails“ platforma ir jei skirti programų kūrimo aplinkos (angl. *IDE*) turi geriausią ir patogiausią integraciją su šia duomenų bazių sistema. Antra, buvo siekiama susipažinti, išmokti ir įgyti patirtį dirbant su šia nauja programavimo kalba.

2.3.6. Egzistuojančių sistemų palyginimo analizė

Rinkoje egzistuoja nemažai įvairių paketų, kurie palengvina asmeninių finansų planavimą ir valdymą. Šiuo metu populiariausi ir labiausiai tarpusavyje konkuruojantys produktai yra „Microsoft Money“ ir „Intuit Quicken“, bet egzistuoja ir kitų gana neblogų produktų. Tarp jų yra ir lietuviškų programų, kurių funkcijos yra šiek tiek mažesnės nei didžiųjų, tačiau jos yra gana gerai pritaikytos Lietuvos rinkai. Taigi, toliau pasistengsime detaliau apžvelgti kiekvieną iš šių programų, įvertinti jų privalumus ir trūkumus.

- **AceMoney Lite 3.10.3**

Su „AceMoney Lite“ išlaidas galima priskirti įvairioms kategorijoms, nes jų programoje yra labai daug. Nors namų biudžeto tvarkytoja pasižymi daugeliu funkcijų, jos valdymas yra aiškus ir paprastas, tačiau kai kurie mygtukų užrašai išversti į lietuvių kalbą netiksliai. „AceMoney Lite“ patiks tiems, kurie investuoja į užsienio vertybinius popierius ar valiutas. Ji gali iš interneto parsisiųsti akcijų kursus bei svarbiausiu biržų indeksus.

Gaila, bet galima vesti tik vienos banko sąskaitos apskaitą. Norėdami, kad programa susilaikytų su daugiau sąskaitų, turėsite įsigyti mokamą „AceMoney“ versiją, kainuojančią apie 30 Lt. Dar vienas didelis trūkumas – neturi komunalinių mokesčių skaičiuoklės [7].

Privalumai: daug išlaidų bei pajamų kategorijų; galima sekti investicijas į akcijas.

Trūkumai: galima tik vienos banko sąskaitos apskaita; nėra komunaliniu mokesčių skaičiuoklės.

- **Microsoft Money 2008**

Tai rimta, draugiška vartotojui asmeninių finansų tvarkymo programa, kurios funkcijos leidžia tvarkyti finansus, biudžetą, sekti investicijas, yra suderinama su internetine bankininkyste, leidžia planuoti mokesčius ir kt. [12]

Priešingai nei daugelis kitų kompanijos „Microsoft“ produktų, namų biudžeto planavimo programa „Microsoft Money 2008“ kainuoja netiek jau ir daug – apie 44 litus. Ši namų biudžeto tvarkytoja puikiai tvarkosi su banko sąskaitų apskaita. Gaila, bet negali automatiškai ištraukti iš el. bankininkystės sistemos duomenų apie operacijas su sąskaitoje esančiais pinigais. Šį darbą programa nuveiktų, jei turėtume sąskaitą JAV banke. Programoje taip pat trūksta specialios skaičiuoklės, kuri padėtų suskaičiuoti, kiek esame skolingi už dujas, vandenį ar elektrą. Didelis programos trūkumas – nėra jos vertimo į lietuvių kalbą, todėl galime susipainioti tarp specifinių anglišku terminų.

Privalumai: patogus vesti banko sąskaitų apskaitą yra akcijų ir kitų investicijų apskaita.

Trūkumai: nėra lietuviškos versijos; nėra komunalinių mokesčių skaičiuoklės.

- **Butas 2006**

Tai lietuviška programa, kuri padės sutaupyti pinigų ir įvertinti savo išlaidų bei pajamų balansą. „Butas 2006“ atrodo patraukliai, be to, programa lengva naudotis net ir nepatyrusiam kompiuterio vartotojui.

Programoje visas savo išlaidas nesunkiai išskirstysite į atskiras kategorijas ir galėsite sekti, kam išleidžiate per daug pinigų. Taip pat galite susikurti sąrašą žmonių, kurie yra mums skolingi (yra šeimos įvykių kalendoriaus). Deja, programa turi ir trūkumų. Joje negalime vesti savo banko sąskaitos apskaitos. Be to, norint atlikti kai kuriuos veiksmus, pavyzdžiui, išsispausdinti išlaidų ataskaitą, turėsite įsigyti specialius programos priedus. Papildomas priedų rinkinys yra mokamas ir kainuoja apie 50 litų.

Privalumai: yra komunalinių mokesčių skaičiuoklė; aiškus valdymas.

Trūkumai: ataskaitas galima spausdinti tik turint mokamą priedą.

- **Namų buhalteris 2000**

Ši programa sukurta beveik prieš dešimt metų, todėl išvaizda ganėtinai atsilieka nuo kitų bandytų programų. Nors „Namų buhalteris 2000“ funkcijų neturi daug, ji galbūt patiks tiems, kurie neturi kantrybės savo išlaidas skirstyti į daugybę kategorijų, pildyti daug lentelių. Programoje galėsime apytiksliai pasiskaičiuoti savo išlaidas ir pajamas. Biudžeto programą galima pagirti už tai, kad joje yra patogi ir lengva naudotis komunalinių mokesčių skaičiuoklė. Ja apskaičiuosite ne tik mokesčius už vandenį ar elektrą, bet ir už internetą ar telefoną. Tai namų biudžeto tvarkymo programą tiems, kurie mėgsta paprastumą, aiškumą ir greitą darbą.

Privalumai: paprasta ir patogiu naudotis; patogi mokesčių skaičiuoklė.

Trūkumai: mažai funkcijų; programoje nėra pagalbos.

- **Intuit Quicken 2008**

Kaip ir „Microsoft“ paketas, „Intuit Quicken 2008“ (kaina 66 Lt) padeda tvarkytis su banko operacijomis, investicijomis, mokesčiais, finansiniu planavimu, biudžetu ir dar daugiau. Taip pat, leidžia apmokėti sąskaitas internetu, atlikti mokėjimo knygelės balansą bei patogiai ir greitai suderinti banko operacijas. Įvairios ataskaitos ir grafikai, parodo, kur išleidžiate daugiausia pinigų ir palygina jūsų dabartines išlaidas su turimu biudžetu (pajamomis). Atėjus mokesčių metui, Quicken duomenys gali būti greitai perkeltami į taip vadinamus greituosius mokesčius (angl. *Turbo Tax*), taip sumažinant duomenų įvedimą. Tačiau kaip ir „Microsoft“, „Intuit Quicken“ yra pritaikytas tik konkrečiam regionui, šiuo atveju Australijai ir Naujajai Zelandijai [11].

Privalumai: patogiu vesti banko sąskaitų apskaitą yra akcijų ir kitų investicijų apskaita.

Trūkumai: nėra lietuviškos versijos; nėra komunalinių mokesčių skaičiuoklės.

- **Moneydance 2008**

Tai mažiau žinoma asmeninių finansų programa, siūlanti: į atsiskaitymo knygele panašią sąsają, greitą ir patogų banko operacijų suderinimą, leidžia stebėti išlaidas, ir atspausdinti įvairias ataskaitas. Ši programinė įranga neturi tokių įmantrių funkcijų, kurias siūlo „Microsoft Money“ ir „Intuit Quicken“, bet jeigu jums reikia tik paprastos programos, tai gali būti visai nebloga alternatyva. Ši programa yra daugiaplatforminė, t.y gali būti naudojama su Windows, Linux arba Mac tipo operacinėmis sistemomis. „Moneydance 2008“ kainuoja netiek jau ir mažai – apie 100 litų.

Privalumai: daugiaplatformė; yra paskolų skaičiuoklė.

Trūkumai: kaina, nėra lietuviškos versijos; nėra komunalinių mokesčių skaičiuoklės.

Sekančioje lentelėje pateiksime visų anksčiau paminėtų asmeninių finansų tvarkymo sistemų palyginimą.

2.1 Lentelė. Asmeninių finansų tvarkymo programų palyginimas

| Programa ir jos vertinimas | Svarba | AccMoney Lite | Microsoft Money | Butas 2006 | Namų buhalteris | Intuit Quicken | Moneydance |
|--|------------|---------------|-----------------|-------------|-----------------|----------------|-------------|
| Diegimas ir pagalba | 22% | 7,55 | 3,14 | 8,23 | 7,32 | 3,68 | 3,50 |
| Lietuviška versija | 10% | 10 | 1 | 10 | 10 | 1 | 1 |
| Spausdintas vartotojo vadovas | 4% | 1 | 1 | 1 | 1 | 1 | 1 |
| Įdiegimas | 3% | 10 | 10 | 10 | 10 | 10 | 10 |
| Užimama vieta kietajame diske | 1% | 8 | 5 | 10 | 10 | 5 | 8 |
| Pagalba programoje | 2% | 5 | 5 | 9 | 1 | 9 | 5 |
| Pagalba internete | 1% | 4 | 5 | 9 | 5 | 9 | 5 |
| Išdiegimas | 1% | 10 | 5 | 10 | 10 | 5 | 10 |
| Programos valdymas ir galimybės | 23% | 7,61 | 7,52 | 8,35 | 7,61 | 9,78 | 8,04 |
| Valdymo patogumas | 10% | 8 | 8 | 8 | 10 | 10 | 7 |
| Mygtukų bei pranešimų aiškumas | 5% | 5 | 8 | 10 | 8 | 9 | 7 |
| Galimybė vesti apskaitą įvairiomis valiutomis | 3% | 10 | 1 | 10 | 1 | 10 | 10 |
| Duomenų apsauga slaptažodžiu | 3% | 10 | 10 | 10 | 10 | 10 | 10 |
| Galimybė naudotis programa keliems vartotojams | 2% | 5 | 10 | 1 | 1 | 10 | 10 |
| Banko sąskaitos apskaita | 18% | 2,33 | 5,67 | 1 | 1 | 5,89 | 5,89 |
| Galimybė gauti duomenis iš mūsų el. bankininkystės sistemos | 10% | 1 | 3 | 1 | 1 | 3 | 3 |
| Kelių skirtingų sąskaitų apskaitą, galima vesti? | 4% | 2 | 10 | 1 | 1 | 10 | 10 |
| Sąskaitos įplaukų bei išlaidų rodymas | 4% | 6 | 8 | 1 | 1 | 9 | 9 |
| Namų biudžeto planavimo funkcijos | 22% | 4,68 | 6,59 | 8,09 | 6,05 | 6,59 | 6,59 |
| Ataskaitos apie mūsų įplaukas / išlaidas bei finansinę būklę | 3% | 6 | 8 | 10 | 6 | 8 | 7 |
| Išlaidų kategorijų skaičius | 3% | 10 | 10 | 10 | 3 | 10 | 10 |
| Priminimai apie išsikolinimus bei | 3% | 5 | 8 | 5 | 1 | 8 | 9 |

| | | | | | | |
|---|-------------|-------------|-------------|-------------|-------------|-------------|
| reguliarius ir atliktus mokėjimus | | | | | | |
| Skolininkų ir skolintojų sąrašas | 3% | 10 | 10 | 10 | 10 | 10 |
| Perspėjimas apie viršytą nustatytą išlaidų limitą | 3% | 1 | 10 | 1 | 1 | 10 |
| Atsiskaitymų už paslaugas (elektrą, dujas, vandenį) skaičiuoklė | 7% | 1 | 1 | 10 | 10 | 1 |
| Papildomos funkcijos | 15% | 8,67 | 8,20 | 1,80 | 3,13 | 7,93 |
| Vertybinių popierių bei kitų investicijų apskaita | 5% | 10 | 10 | 1 | 1 | 10 |
| Galimybė iškelti duomenis į kitą formatą | 4% | 5 | 10 | 4 | 9 | 7 |
| Galimybė kuri savo pačių ataskaitas | 3% | 10 | 10 | 1 | 1 | 10 |
| Paskolų ir kitos skaičiuoklės | 3% | 10 | 1 | 1 | 1 | 10 |
| Galutinis vertinimas balais | 100% | 6,16 | 6,12 | 5,96 | 5,34 | 6,76 |
| | | 6,24 | | | | |

Apžvelgus lyginamas sistemas (2.1 Lentelė), galime išskirti keturis lyderius (*AceMoney Lite*, *Microsoft Money*, *Intuit Quicken* ir *MoneyDance*), nedaug nuo jų atsilieka ir likusios programos. Šios programos pasižymi funkcijų gausa ir valdymo patogumu. Kitos sistemos taip realizuoja vartotojui reikalingiausias funkcijas. Taip pat, viskas priklauso ir nuo poreikių, pvz. vartotojui gali užtekti ir „Microsoft Excel“ lentelės blanko pavadinimu „Šeimos mėnesinis biudžetas“ (angl. *Family monthly budget*). Viskas atsiremia į pinigus ir laiką, kuriuos norime sutaupyti.

Todėl mūsų kuriamos sistemos tikslas yra pateikti kuo daugiau naudingesnių funkcijų vartotojui, o taip pat ir konkuruoti su visomis rinkoje esančiomis sistemomis, bet nepamirštant ir turimus realius resursus sistemai sukurti.

Verta paminėti, kad beveik visos minėtos sistemos veikia tik vartotojo operacinės sistemos (OS) aplinkoje, kitaip nei mūsų planuojama sistema, kuri veiktų internetinės sistemos pagrindu. Internetu veikiančios sistemos yra universalesnės, nereikalaujančios didelių sistemos resursų ir nepriklauso nuo operacinės sistemos.

2.3.6.1. Esminės problemos

Nors galima rasti daugybę asmeninių finansų valdymui skirtų programų mes apžvelgėme tik keletą, taip pat pavyko išbandyti Lietuvos rinkai pritaikytas programas. Visos asmeninių finansų valdymui skirtos programos labai panašios viena į kitą, turi panašių savybių. Tarpusavyje jos skiriasi tik papildomomis funkcijomis, naudojimosi patogumu bei kainomis. Tačiau pasigesta funkcijų, kurios gana gerai tiktų Lietuvos rinkai. Visose sistemose yra papildomų funkcijų, kurių nereikia ar neprireiks daugumai Lietuvos vartotojų, už kurias, jeigu vartotojas pirktų, kurią nors prieš tai apžvelgtą sistemą, reikėtų sumokėti. Kuriant naują asmeninių finansų valdymo ir planavimo sistemą bus panaudotos visos reikalingos ir būtiniausios išvardintų sistemų savybės.

Išbandžius „Microsoft Money“ ir „Intuit Quicken“, pagrindinis dalykas, kuris krenta į akis tai, kad programos daugiausia pritaikytos Amerikos vartotojams. Įsigijus tokią

programinę įrangą Lietuvoje, jos funkcionalumas ir pritaikomumas ženkliai sumažėtų. Tokia investicija galima sakyti neatsipirktų. Tačiau, jei vertinti programos funkcijas ir galimybes, tai jos tikrai yra nemažos, pradedant paprasčiausiomis ir baigiant akcijų ir investicijų valdymu. Čia kiekvienas vartotojas suras sau reikalingą įrankį ar informaciją. Internetinės bankininkystės įrankių pagalba, galima greitai valdyti ir planuoti savo finansus, tačiau kaip jau minėjau, tokios funkcijos prieinamos tik Amerikos vartotojams. Mūsų šalyje tokias sistemas turi kiekvienas bankas ir internetinės bankininkystės sistemos atskirai yra prieinamos daugumai vartotojų. Taigi, norėdami pilnai pritaikyti visas funkcijas Lietuvos vartotojams ir konkuruoti, turime kurti savo asmeninių finansų planavimo ir valdymo programinę įrangą.

2.4. Analizės išvados

1. Apžvelgti esminiai akcentai susiję su sistemos struktūros bei architektūros kūrimu, saugumo ir palaikomumo užtikrinimu.

2. Apibrėžta sistemos taikymo sritis – asmeninių finansų planavimo ir valdymo sistemos sprendimas vartotojams, kuris padėtų vartotojui išspręsti asmeninių finansų planavimo ir valdymo problemas, padėtų taupyti laiką ir pinigus.

3. Išanalizavus užsakovo iškeltus reikalavimus sistemai, nustatyti penkti probleminiai sistemos realizavimo uždaviniai: universalios ir paprastos sistemos struktūros sukūrimas, finansų valdymas, informacijos įvedimas/duomenų dubliavimas, vartotojų valdymas ir saugumas, integracija su administravimo posisteme. Šiems probleminiams uždaviniams išspręsti buvo pateikti jų galimi sprendimai.

5. Palyginus plačiausiai naudojamas internetines platformas, skirtas įvairių internetinių sistemų realizavimui, taip pat atsižvelgus į sistemai keliamus vartotojo reikalavimus ir numatomus poreikius jų realizavimui, pasirinkta sistemos kūrimo platforma – „Ruby on Rails“.

6. Išanalizavus egzistuojančius sprendimus, atlikta jų palyginimo analizė. Taip pat, įsitikinta, kad alternatyvių produktų pritaikymas Lietuvos rinkai yra gana problematiškas. Didelė dalis apžvelgtų programų bei jų funkcijų buvo pritaikytos turtingoms vakarų rinkoms. Nustatyta, kad asmeninių finansų valdymas šiose programose skiriasi nuo egzistuojančių ir įprastų Lietuvoje.

3. ASMENINIŲ FINANSŲ PLANAVIMO IR VALDYMO SISTEMOS REALIZACIJA

3.1. Skyriaus paskirtis

Šiame skyriuje trumpai aptariami projekto tikslai, sprendžiami uždaviniai bei priimti techniniai sprendimai. Vėliau apžvelgiamas bendras sistemos architektūros vaizdas, tuo pačiu pateikiamos esminės sistemos architektūrą paaiškinančios diagramos. Galiausiai pateikiamas sistemos duomenų modelis ir aptariama projekto išėiga.

3.2. Projekto tikslai

Projekto tikslas – sukurti asmeninių finansų planavimo ir valdymo sistemą, kuri padėtų vartotojui išspręsti asmeninių finansų planavimo ir valdymo problemas, padėtų taupyti laiką ir pinigus. Sistema pasižymėtų tokiomis pagrindinėmis savybėmis:

- Saugumas;
- Daugiavartojiškumas;
- Sistemos pasiekiamumas internetu (on-line);
- Paprastumas;
- Funkcionalumas;
- „Ruby on Rails“ karkaso panaudojimas.

Vartotojai prisijungę prie sistemos galės atlikti įvairias funkcijas: įvesti bei tvarkyti mokesčius, stebėti balansą, įvesti bei peržiūrėti gaunamas pajamas, išleidžiamus pinigus, skirstyti bei peržiūrėti išlaidas pagal tam tikras kategorijas, sudaryti ir įvairiais pjūviais peržiūrėti ataskaitas. Tai tik dalis pagrindinių sistemos funkcijų, be kurių asmeninių finansų sistema yra neįsivaizduojama. Šių ir kitų funkcijų dėka, vartotojas palengvins savo asmeninių finansų tvarkymo uždavinius.

Vartotojo atžvilgiu, galime išskirti tokius pagrindinius asmeninių finansų sistemai keliamus tikslus:

- Užtikrintas sistemos bei duomenų saugumas;
- Produkto sudėtingumas negali viršyti esamų rinkoje sukurtų sistemų;
- Turi būti lengvai naudojama (viskas lietuvių kalba);
- Geras darbo našumas (sistemos veikimo efektyvumas);
- Kuo didesnis naudingų funkcijų pasirinkimas;
- Reikalingas kuo mažesnis duomenų dubliavimas;
- Kuo paprastesnis finansinių išraiškų pateikimo būdas;
- Produkto palaikymas ir tolimesnių funkcijų praplėtimas reikalui esant.

3.3. Sprendžiami uždaviniai

Šiame skyriuje apžvelgiami projekto tikslai iš programų inžinerijos pusės. Toliau pateikiamas sistemos sprendžiamų uždavinių sąrašas su trumpais paaiškinimais:

- Sistema turi būti suskurta naudojantis nauja „Ruby on Rails“ programavimo kalba bei naujomis interneto (angl. *Web 2.0*) technologijomis. Tai buvo vienas iš pagrindinių užsakovo reikalavimų. Nes tik taip galima užtikrinti, kad sistema atitiks šiuolaikinius vartotojo keliamus reikalavimus, ir žiūrint į ateitį, užtikrins nemažą tobulinimo potencialą.
- Sistema turi būti prieinama iš bet kurios vietos. Kad sistema būtų galima naudotis beveik prie kiekvieno kompiuterio, tai vienas iš pagrindinių reikalavimų buvo jos pasiekiamumas. Dėl šios priežasties reikia sukurti sistemą, kuri būtų prieinama internetu. Toks uždavinys įneša ne vien tik papildomus reikalavimus sistemos vykdymo charakteristikoms bei nefunkciniams reikalavimams, bet tuo pačiu ir priverčia atsisakyti kitų sistemos kūrimo variantų.
- Vartotojo duomenys turi būti prieinami tik jam pačiam. Kadangi, tai bus internetinė sistema, būtina užtikrinti vartotojo informacijos konfidencialumą. Nes sistema turės dirbti su vartotojo asmeniniais finansiniais duomenimis, nors ir nesitikima, kad vartotojas šioje sistemoje laikys labai svarbią informaciją. tačiau būtina keliais lygiais užtikrinti vartotojo ir priėjimo prie informacijos saugumą.
- Reikia realizuoti vartotojo sąsają, kuria būtų paprasta ir lengva naudotis. Vartotojui svarbiausia, kad sistema, kuria jis naudojasi, būtų nesudėtinga. Taigi, svarbu sistemos funkcionalumą kuo paprasčiau pateikti sistemos naudotojui. Tai padaryti, kartais nebūna taip lengva.
- Sistema turi atlikti minimalias asmeninių finansų apskaitos funkcijas. Sistema turi leisti vartotojui atlikti paprasčiausius pajamų, išlaidų, mokesčių, atsiskaitymo knygelės funkcijas bei leisti rezultatus peržiūrėti grafiniu pavidalu ar išsisaugoti ataskaitos forma. Tai reiškia, kad reikės sekti pajamų–išlaidų balansą, grupuoti informaciją pagal įvairias kategorijas ir pan.
- Turi būti realizuotas lankstus mechanizmas sukauptiems vartotojo finansų duomenims atvaizduoti sistemoje. Tai apima duomenų grafinį atvaizdavimą ekrane, ataskaitų spausdinimo, duomenų eksporto ir kitas funkcijas.

Aukščiau pateikti tik bendri sistemos sprendžiami uždaviniai, kurie realizacijos metu gali būti toliau smulkinami ir koreguojami. Pagal šiuos pateiktus uždavinius taip pat galima susidaryti bendrą sistemos panaudos atvejų schemą.

Dėl šių uždavinių išskylantys papildomi funkciniai ir nefunkciniai sistemos reikalavimai bei kitos problemos bus aptariamoms šioje ir kitoje darbo dalyse.

3.4. Priimti techniniai sprendimai

Įvertinus visus projektui iškeltus reikalavimus ir apribojimus nuspręsta projektą kurti kaip internetinį puslapį, jį realizuojant objektiškai orientuotos „Ruby on Rails“ programavimo kalbos pagalba bei remiantis Modelis – Vaizdas – Valdiklis šablonu. Kitų galimų realizavimo variantų čia nenagrinėsime, o parinktą Ruby on Rails programavimo kalbą aptarėme 2.3.5 skyrelyje „Realizavimo priemonių pasirinkimas“.

3.4.1. Diegimo aplinka

Kuriama internetinė sistema bus įdiegta serveryje ir pasiekama per interneto naršyklę, todėl nutolusiems jos naudotojams jokių specialių diegimo priemonių nereikės.

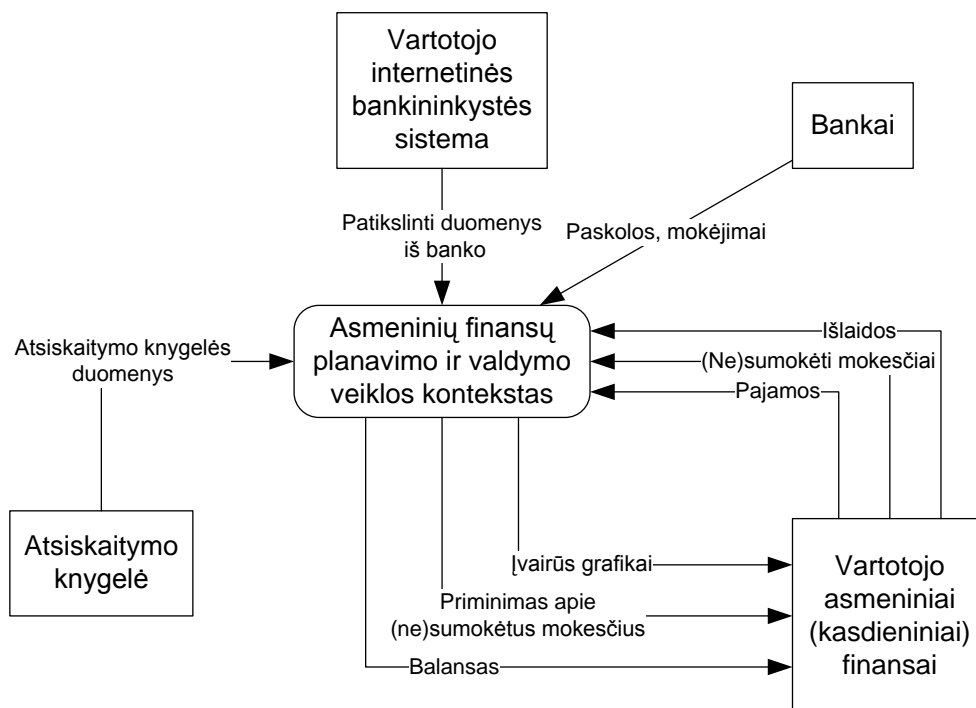
Minimalūs reikalavimai vartotojų programinei įrangai:

- Windows, Linux ar kita operacinė sistema (OS);
- Interneto naršyklė su įjungtais ActiveX komponentais.

Minimalūs reikalavimai serverio techninei įrangai:

- Ruby on Rails palaikymas;
- MySql duomenų bazė;
- Administravimo priemonės.

3.5. Veiklos kontekstas



3.1 Pav. Sistemos veiklos kontekstas

Aukščiau esančiame paveiksliuke (3.1 Pav.), pateikiamas sistemos veiklos kontekstas. Čia matome sistemos veikimo metu vykstančius duomenų mainus. Pagal šią schemą galime susidaryti bendrą supratimą apie funkcinius sistemos reikalavimus.

3.5.1. Veiklos padalinimas

Žvelgdami į sistemos veiklos kontekstą galime išskirti šiuos apdorojamus informacijos srautus:

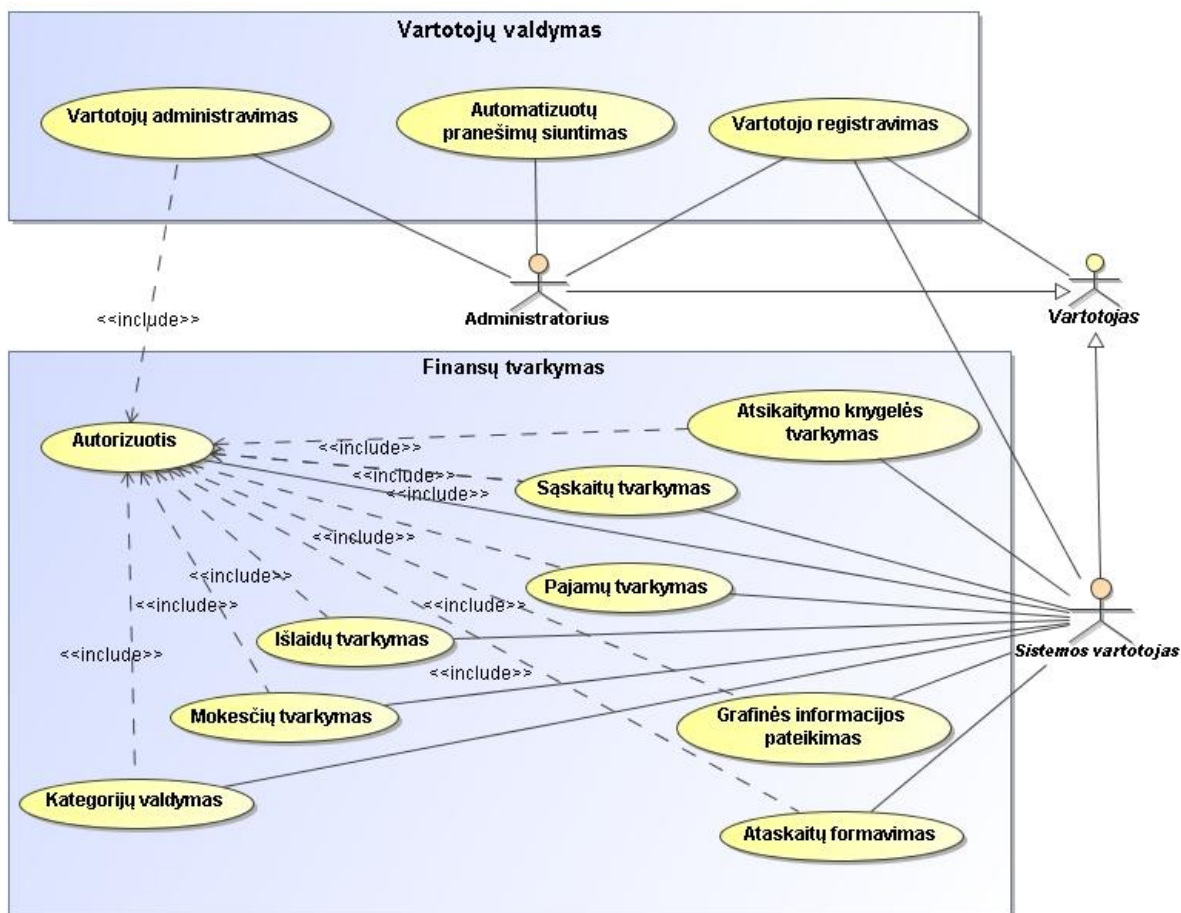
3.1 Lentelė. Veiklos įvykių sąrašas

| Įvykio pavadinimas | Įeinantys/Išeinantys informacijos srautai |
|---|--|
| Vartotojas įveda informaciją iš atsiskaitymo knygelės | Atsiskaitymo knygelės duomenys (<i>in</i>) |
| Įvedami patikslinti duomenys iš internetinės bankininkystės sistemos | Patikslinti duomenys iš banko (<i>in</i>) |
| Vartotojas įveda savo asmeninių finansų duomenis (išlaidas, pajamas, mokesčius) | Išlaidos, pajamos, mokesčiai (<i>in</i>) |
| Įvedama informacija gaunama iš bankų (paskolos, mokėjimai) | Paskolos, mokėjimai (<i>in</i>) |
| Išvedamas apskaičiuotas balansas | Balansas (<i>out</i>) |
| Vartotojui primenama apie (ne)sumokėtus mokesčius | Priminimas apie (ne)sumokėtus mokesčius (<i>out</i>) |
| Vartotojui pateikiami pajamų, išlaidų, mokesčių pasiskirstymo grafikai | Pajamų, išlaidų, mokesčių grafikai (<i>out</i>) |

Pasitelkdami informacija iš šios lentelės, galime susidaryti bendrą sistemos struktūros vaizdą. Matome atskirus modulius, bendrus duomenis ir ryšius tarp šių modulių ir jų duomenų.

3.6. Panaudojimo atvejai

Nagrinėjant sistemos veiklos kontekstą, bei surinktus sistemos reikalavimus, buvo sudaryta žemiau pateikta sistemos panaudojimo atvejų diagrama (3.2 Pav.). Šiame darbe pateikiami tik aukščiausio lygio panaudos atvejai. Kiekvieno iš jų detalesnis nagrinėjimas nepateikiamas.



3.2 Pav. Panaudojimo atvejų diagrama

Pagal šią diagramą galime vartotojus suskirstyti pagal jiems priklausančias funkcijas. Šios schemos pagalba, taip pat galime apžvelgti ir bazinius sistemos saugumo reikalavimus (nes matome, prie kurių sistemos funkcijų vartotojai turi priejimą, koks funkcionalumas jiems yra būtina reikalingas).

Panagrinėję diagramą, lengvai išskiriame dvi pagrindines sistemos panaudojamumo grupes arba modulius (3.2 Lentelė).

| Grupė | Grupę sudarantys panaudojimo atvejai |
|--|---|
| <i>Sistemos administravimas</i> | Vartotojų valdymas; Vartotojo registravimas. |
| <i>Sistemos teikiamų paslaugų naudojimas</i> | Sąskaitų tvarkymas; Pajamų tvarkymas; Išlaidų tvarkymas; Mokesčių tvarkymas; Grafinės informacijos pateikimas; Atsiskaitymo knygelės tvarkymas; Kategorijų valdymas; Ataskaitų formavimas. |

3.7. Funkciniai reikalavimai

Sistemos reikalavimai buvo renkami naudojantis *Volere* šablonu [12]. Žemiau pateikiami pagrindiniai sistemos funkciniai reikalavimai (sutrumpintas aprašymas):

1. Sistema turi leisti pačiam vartotojui prisiregistruoti;
2. Sistema turi pranešti, jei vartotojo teisės naudojantis sistema yra apribotos;
3. Sistemos funkcionalumas turi būti prieinamas tik identifikuotiems vartotojams;
4. Administratorius turi turėti vartotojų administravimo galimybes;
5. Sistema turi leisti sukurti vieną ir daugiau vartotojo sąskaitų;
6. Sistema turi leisti pakeisti įvestos sąskaitos duomenis;
7. Sistema turi leisti vartotojui patogiai įvesti pasikartojančius mokesčius;
8. Sistema turi leisti vartotojui koreguoti įvestus mokesčius;
9. Sistema turi leisti vartotojui pasiskaičiuoti mokesčių dydžius;
10. Vartotojas turi turėti galimybę įvesti pajamas;
11. Vartotojas turi turėti galimybę koreguoti visas įvestas pajamas;
12. Vartotojas turi turėti galimybę įvesti išlaidas;
13. Vartotojas turi turėti galimybę koreguoti įvestas išlaidas;
14. Vartotojas turi matyti suskaičiuotą balansą;
15. Vartotojas turi turėti galimybę kurti naujas pajamų/išlaidų kategorijas;
16. Vartotojas turi turėti galimybę koreguoti pajamų/išlaidų kategorijas;
17. Vartotojas turi turėti galimybę įvesti atsiskaitymo knygelės duomenis;
18. Vartotojas turi turėti galimybę koreguoti atsiskaitymo knygelės įrašus;
19. Vartotojas turi grafiškai matyti mėnesio pajamas-išlaidas pagal grupes (kategorijas);

20. Vartotojas turi turėti galimybę matyti pajamų-išlaidų pokyčius ir jų grafiką;
21. Vartotojas turi turėti galimybę išsifiltruoti duomenis pagal pasirinktą kriterijų;
22. Vartotojas turi turėti galimybę išsieksportuoti duomenis į pasirinktą formatą;
23. Vartotojas turi turėti galimybę atlikti paiešką duomenų;
24. Vartotojas turi grafiškai matyti mokesčių ir atsiskaitymo knygelės duomenis pagal grupes (kategorijas);
25. Sistemoje reikalingas kalendorius žymintis įvairius įvykius;
26. Vartotojas turi matyti kalendoriuje savo įvestas pajamas, išlaidas, mokesčius ir kitą aktualią informaciją;
27. Sistema turi leisti iš turimų duomenų suformuoti ir pateikti įvairias ataskaitas;
28. Sistema turi leisti administratoriui valdyti vartotojo būsenas t.y. patvirtintas, nepatvirtintas ir pan.;
29. Sistema turi leisti administratoriui suskirstyti vartotojus į grupes;
30. Sistema turi leisti siųsti automatinius pranešimus vartotojui ir administratoriui apie vartotojo sukūrimą arba jo duomenų pasikeitimus.

3.8. Nefunkciniai reikalavimai

Šiame skyriuje glaustai pateikiamas nefunkcinių reikalavimų sąrašas sugrupuotas į kelias pagrindines kategorijas.

- a) Reikalavimai sistemos išvaizdai
 1. Lengvai skaitoma ir suprantama sąsaja;
 2. Paprastas (nesudėtingas) sąsajos naudojimas;
 3. Prieinamumas, kad vartotojas nesivaržytų naudodamas sistemą;
 4. Neįkyri sąsaja (nereikalaujanti pastoviai ką nors kelis kartus patvirtinti);
 5. Novatoriška ir šiuolaikinė išvaizda.
- b) *Reikalavimai panaudojamumui*
 6. Sistema turi būti panaudojama be išankstinio apmokymo;
 7. Sistema turi leisti vartotojui taisyti jo padarytas klaidas;
 8. Sistemos vartotojo sąsaja turi būti lietuviška;
 9. Turi būti galimybė pasižiūrėti raktinės informacijos paaiškinimus.
- c) *Reikalavimai vykdymo charakteristikoms*
 10. Efektyvus veiksmų vykdymas;
 11. Skaičiai rodomi su fiksuotu kieku skaitmenų po kablelio.

d) Reikalavimai veikimo sąlygoms

12. Sistema turi veikti visose populiariausiose interneto naršyklėse.

e) Reikalavimai sistemos priežiūrai

13. Sistemą turi veikti nuo operacinės sistemos nepriklausomame serveryje.

f) Reikalavimai saugumui

14. Sukurti turi garantuoti saugų vartotojo prisijungimą;

15. Sistema turi neleisti neautorizuotiems vartotojams prisijunkti prie sistemos ir ja naudotis;

16. Kiekvienas prisijungęs vartotojas turi priėjimą tik prie savo duomenų;

17. Neaktyvūs vartotojai turi būti atjungiami nuo sistemos;

18. Visi svarbūs ir konfidencialūs duomenys turi būti saugomi užkoduotame formate.

g) Kultūriniai–politiniai reikalavimai

19. Sistemoje naudojama korektiška lietuvių kalba, nežargoniniai terminai;

20. Sistema neturi įžeisti religinių ar etninių grupių.

h) Teisiniai reikalavimai

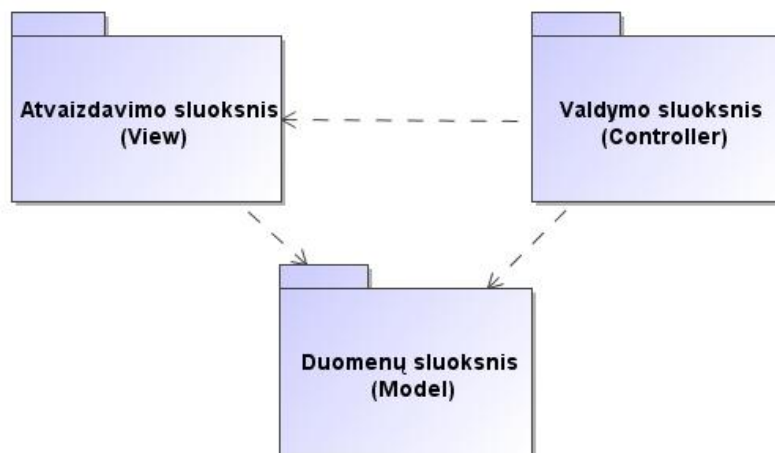
21. Sistema turi veikti pagal Lietuvos respublikoje galiojančius įstatymus.

3.9. Architektūros specifikacija

3.9.1. Bendras architektūros vaizdas

Asmeninių finansų planavimo ir valdymo sistemos kūrimo panaudotas trijų lygių modelis, kuris plačiau buvo aprašytas 2.2.1 skyrelyje „Struktūrizuotos sistemos“. Taip pat, sistemai pritaikytas ir 2.2.2 skyrelyje išanalizuotas „Modelis – Vaizdas – Valdiklis šablonas“.

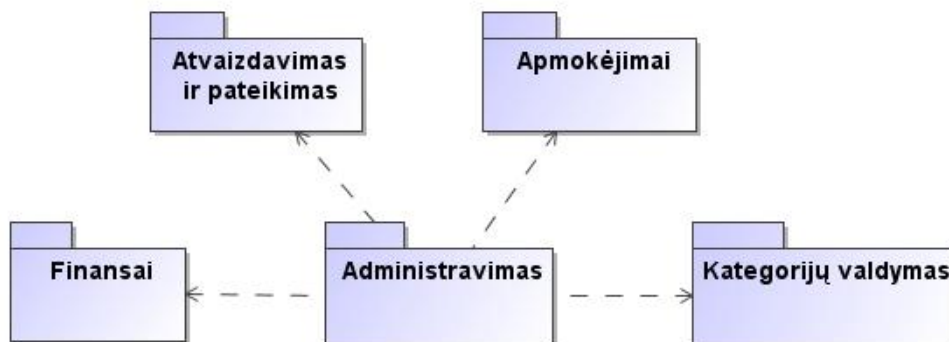
Remiantis šiuo modeliu ir šablonu, asmeninių finansų planavimo ir valdymo sistemą sudarys vartotojo sąsajos (atvaizdavimo), valdymo ir duomenų sluoksniai, kurie suskirstyti į atitinkamus paketus (3.3 Pav.).



3.3 Pav. Sistemos struktūra ir ryšys tarp sluoksnių

3.9.2. Sistemos architektūra

Sistemos architektūra trumpai bus apžvelgiama sekančiuose skyreliuose. Šiuose skyreliuose bus pateiktos realizuotos klasės ir jų struktūra. Kuriama sistema nėra labai didelė ir sudėtinga, todėl ji nėra išskiriama į posistemas, bet yra atliekama paketų dekompozicija. Žemiau pateiktoje diagramoje (3.4 Pav.) pavaizduoti penki pagrindiniai sistemos paketai, kurie įkomponuoja pagrindines sistemos funkcijas.



3.4 Pav. Sistemos pagrindinių modulių diagrama

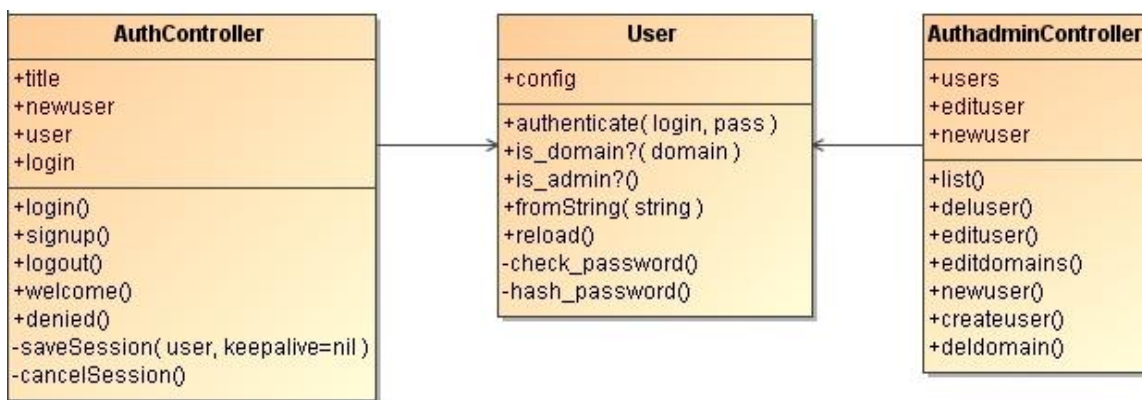
3.9.2.1. Paketų detalizavimas

Kadangi mūsų kuriama sistema remiasi MVC architektūra, todėl toliau visi pateikti paketai bus labai panašūs ir pagrinde bus sudaryti iš modelio bei valdiklių klasių, kadangi beveik visos atvaizdavimo (view) klasės savyje metodų ar funkcijų aprašymų neturi, tai jos bus paliktos nuošaliau, t.y. klasių diagramose jų nesimatys. Taipogi, kiekvienas valdiklis (kontroleris) turi labai daug vaizdinių puslapių, todėl juos čia visus patalpinti ir pavaizduoti būtų labai sudėtinga ir nepraktiška. Taip pat, matysime, kad duomenų modeliai turi mažai metodų, taip yra dėl to, kad dauguma veiksmų (naujo įrašo sukūrimas, redagavimas, peržiūra, sunaikinimas) atlieka „Ruby on Rails“ programinis karkasas (angl. *framework*).

3.9.2.2. Administravimas modulio architektūra

Apibrėžimas. Pakete pateikiamos klasės (3.5 Pav.), kurios valdo sistemos vartotojo prisijungimą prie sistemos, t.y. per jas yra išskviečiami pagrindiniai prisijungimo, atsijungimo kiti sistemos metodai. Beveik kiekvienas „AuthController“ (už prisijungimą atsakingo valdiklio) metodas po veiksmo generuoja vaizdą (view), pvz. įvykdžius prisijungimo metodą „login()“ bus sugeneruotas ir užkrautas „login.rhtml“ vaizdinys.

Atsakomybės. Paketas reprezentuoja sistemos administravimo branduolį ir vykdo prisiregistravimo prie sistemos valdymą. Vykdo prisijungimo, atsijungimo funkcijas, bei atitinkamų puslapių atvaizdavimą. Prisijungimo „AuthController“ valdo paprastų vartotojų (User) prisijungimą, o „AuthAdminController“ suteikia vartotojų administravimo galimybes, šie kontroleriai taip pat išskviečia puslapių atvaizdavimą.



3.5 Pav. Paketo administravimas klasių diagrama

Žemiau aprašomos pagrindinės šio modulio klasės:

AuthController – Klasė skirta valdyti paprastų vartotojų (User) prisijungimą bei prisiregistravimą, taip pat išskviečia atitinkamų puslapių atvaizdavimą (View). Klasė taip leidžia uždrausti vartotojų prisijungimą bei valdo vartotojų sesijas.

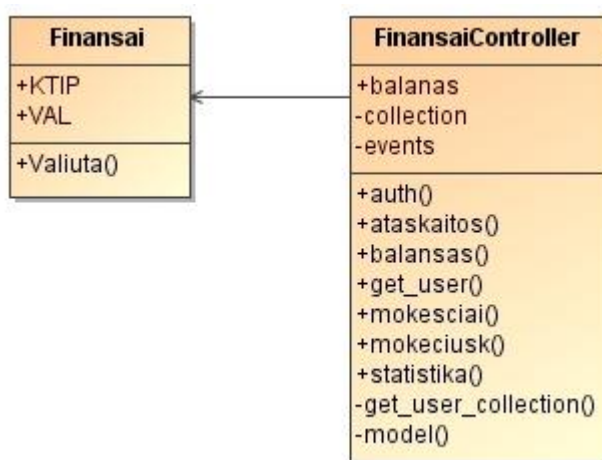
User – Pagrindinė vartotojų duomenų klasė. Atsakomybės: vartotojų autentifikavimas, duomenų valdymas, keitimas, atnaujinimas.

AuthAdminController – Klasė skirta administruoti vartotojams (User) bei jų priėjimo sritims (angl. *domains*). Atsakomybės: Naujų vartotojų sukūrimas, pateikimas, redagavimas, šalinimas, naujų sričių įtraukimas bei šalinimas.

3.9.2.3. Finansai modulio architektūra

Apibrėžimas. Pakete pateikiamos vartotojo finansų (sąskaitų) klasės (3.6 Pav.).

Atsakomybės. Pakete pateikiamos klasės, kurios atsakingos už vartotojo informacijos apie esamas sąskaitas, saugojimą ir apdorojimą.



3.6 Pav. Paketo Finansai klasių diagrama

Žemiau aprašomos pagrindinės šio modulio klasės:

FinansaiController – Klasė skirta vartotojo sąskaitoms (banko kortelėms, sąskaitoms) tvarkyti bei atlieka mokesčių skaičiuoklės funkcijas, taip pat valdo ataskaitų puslapių atvaizdavimą.

Finansai – Klasė skirta vartotojo sąskaitų duomenims saugoti. Atsakomybės: naujų įrašų sukūrimas, redagavimas, atnaujinimas.

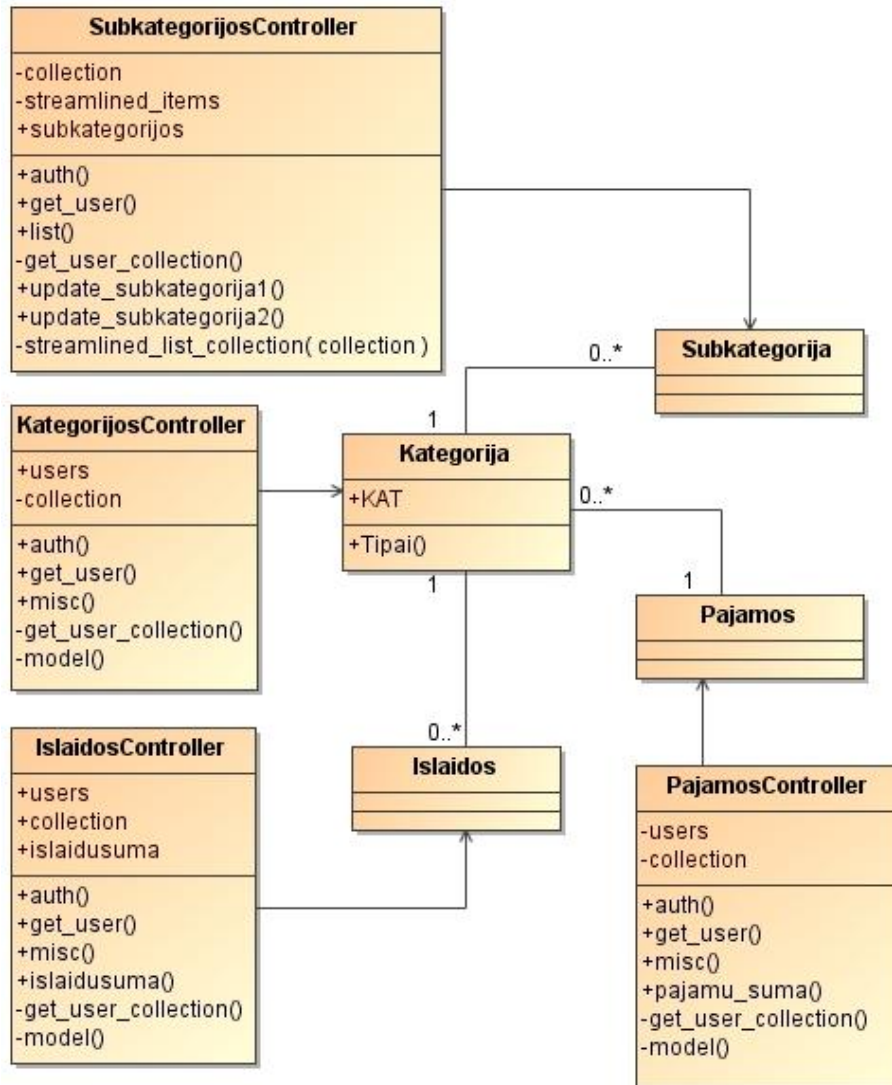
3.9.2.4. Kategorijų valdymas modulio architektūra

Apibrėžimas. Pakete pateikiamos kategorijų valdymo klasės (3.7 Pav.), kurios naudojamos kituose moduluose.

Atsakomybės. Pakete saugomos kategorijos ir subkategorijos yra skirtos ir dažniausiai naudojamos atskirti ir grupuoti pajamas bei išlaidas.

Struktūra. Paketą sudaro klasės *Kategorija* ir *Subkategorija*, kurios saugo bei apdoroja kategorijos bei subkategorijos duomenis duomenų bazėje. Kategorijų ir

subkategorijų kontrolieris yra atsakingas už su kategorijomis susijusių veiksmų vykdymą. Pajamų bei išlaidų duomenų klasės susijusios su mūsų minėtomis klasėmis ir yra atsakingos už savo duomenų saugojimą ir atnaujinimą, atitinkamai šios klasės turi savo valdiklius. Klasių struktūra pateikta diagramoje.

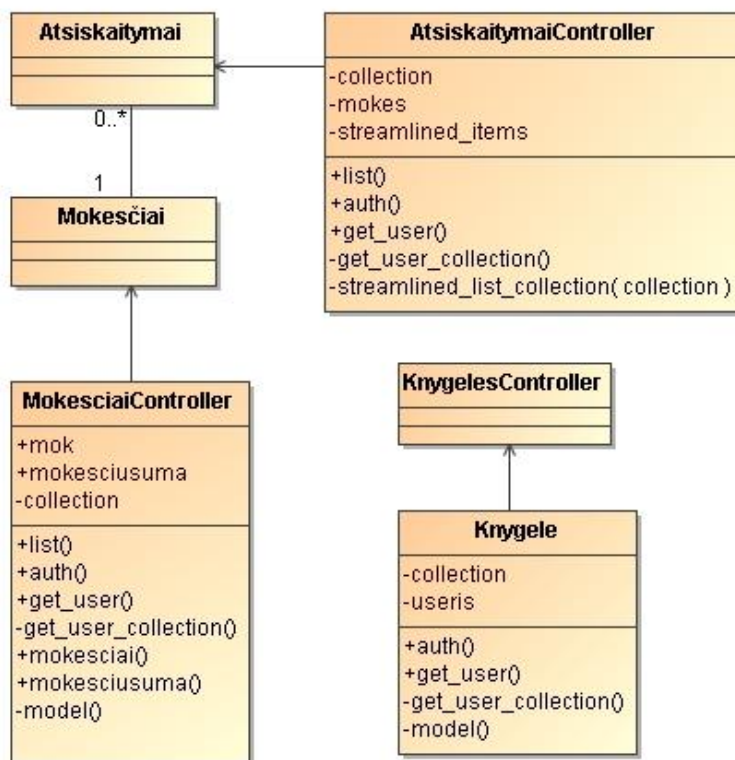


3.7 Pav. Paketo Kategorijos klasių diagrama

3.9.2.5. Apmokėjimai modulio architektūra

Apibrėžimas. Pakete pateikiamos atsiskaitymo knygelės bei įvairių mokesčių apmokėjimo klasės (3.8 Pav.).

Atsakomybės. Pakete saugomi atsiskaitymo knygelės, apmokėjimų bei mokesčių duomenys yra skirti įvairiems vartotojo atsiskaitymams įvesti bei saugoti.



3.8 Pav. Paketo Apmokėjimai klasių diagrama

Žemiau aprašomos pagrindinės šio modulio klasės:

MokesčiaiController – Klasė skirta tvarkyti vartotojų mokesčiams. Klasė leidžia vartotojui susikurti norimas mokesčių grupes ir pagal jas toliau įvesti norimus atsiskaitymus.

Mokesčiai – Klasė skirta mokesčių duomenims saugoti duomenų bazėje. Atsakomybės: naujų mokesčių grupių sukūrimas, redagavimas, atnaujinimas, pašalinimas.

AtsiskaitymaiController – Klasė skirta vartotojo atsiskaitymams (pagal mokesčių grupes) tvarkyti. Klasė leidžia vartotojui įvesti įvairiausių atsiskaitymus į prieš tai sukurtas mokesčių grupes.

Atsiskaitymai– Klasė skirta atsiskaitymo duomenims saugoti duomenų bazėje. Atsakomybės: naujų atsiskaitymų sukūrimas, pateikimas, redagavimas, šalinimas.

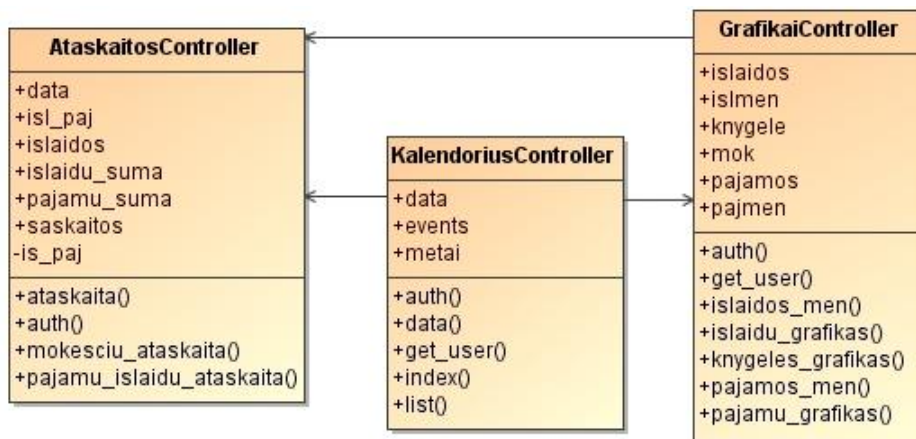
KnygelesController – Klasė skirta vartotojo atsiskaitymo knygelei tvarkyti. Klasė leidžia vartotojui įvesti einamo mėnesio atsiskaitymus už įvairias komunalines paslaugas.

Knygele – Klasė skirta atsiskaitymo knygelės duomenims saugoti duomenų bazėje. Atsakomybės: naujų knygelės įrašų sukūrimas, pateikimas, redagavimas, šalinimas.

3.9.2.6. Atvaizdavimas ir pateikimas modulio architektūra

Apibrėžimas. Pakete pateikiamos ataskaitų, kalendoriaus bei grafikų apdorojimo klasės (3.9 Pav.).

Atsakomybės. Paketas atsakingas už vizualios informacijos pateikimą vartotojui, tai gali būti ataskaitos suformavimas, įvykių kalendoriuje atvaizdavimas ar grafikų suformavimas.



3.9 Pav. Paketo Atvaizdavimas ir pateikimas klasių diagrama

Žemiau aprašomos pagrindinės šio modulio klasės:

AtaskaitosController – Klasė skirta pasirinktoms vartotojo ataskaitoms generuoti ir atvaizduoti. Klasė leidžia vartotojui suformuoti norimą ataskaitos variantą pagal pasirinktą datą.

KalendoriusController – Klasė skirta įvairiems vartotojo apmokėjimas atvaizduoti kalendoriuje. Šios klasės pagalba, vartotojas gali lengvai peržiūrėti kalendoriuje, bet kokio pasirinkto mėnesio buvusias ar esančias pajamas, išlaidas ir apmokėjimus.

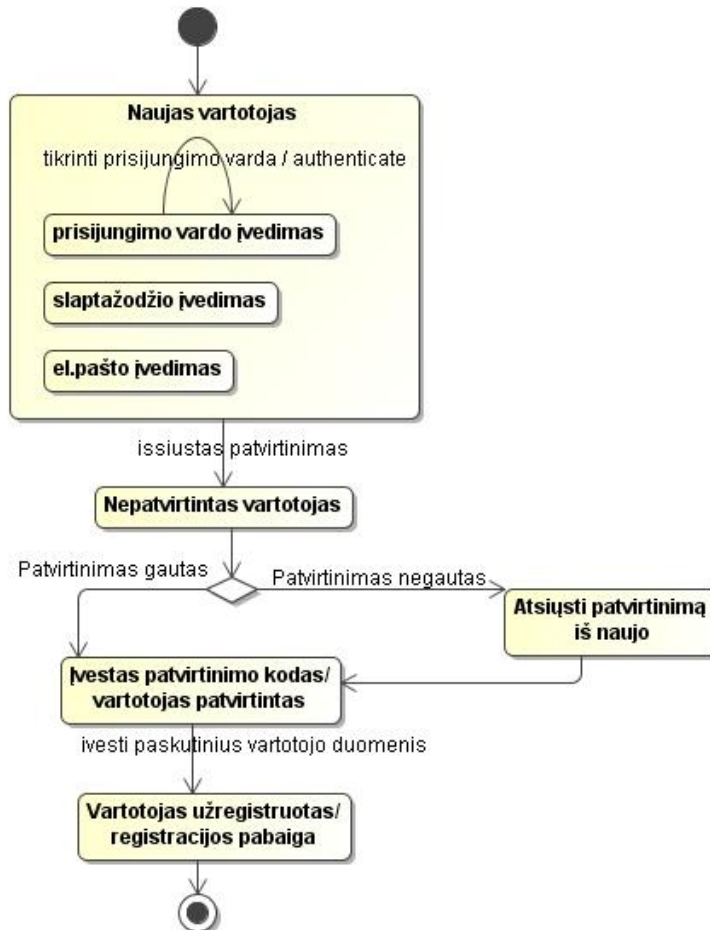
GrafikaiController – Klasė skirta grafiniam duomenų atvaizdavimui pateikti. Šios klasės pagalba, vartotojas sistemoje gali grafiškai pamatyti savo įvestų duomenų pasiskirstymą pagal įvairius kriterijus. Tai gali būti pajamų pokyčio, išlaidų pasiskirstymo ir kiti grafikai.

3.10. Sistemos dinaminis vaizdas

Šiame skyriuje pateikiamos esminės sistemos būsenų, veiklos ir sąveikos UML diagramos.

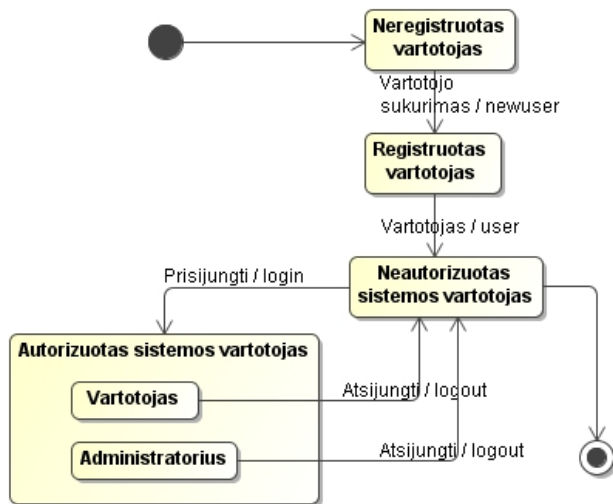
3.10.1. Būsenų diagramos

Žemiau pateiktose diagramose, pavaizduota naujo vartotojo sukūrimo (3.10 Pav.) ir prisijungimo (3.11 Pav.) veiksmų būsenų diagramos.



3.10 Pav. Vartotojo registravimosi būsenos diagrama

Šio paveikslėlio (3.10 Pav.) esmė, parodyti kaip keičiasi vartotojo būsenos sistemoje, registravimosi metu. Matome, kad norint prisijungti prie sistemos pirmiausia reikia susikurti naują vartotoją, įvedant savo prisijungimo duomenis. Įvedus šiuos duomenis, vartotojui atsiunčiamas patvirtinimas į nurodytą (registracijos metu) el. pašto dėžutę. Patvirtinęs savo registraciją, vartotojas jau gali pradėti naudotis sistema.

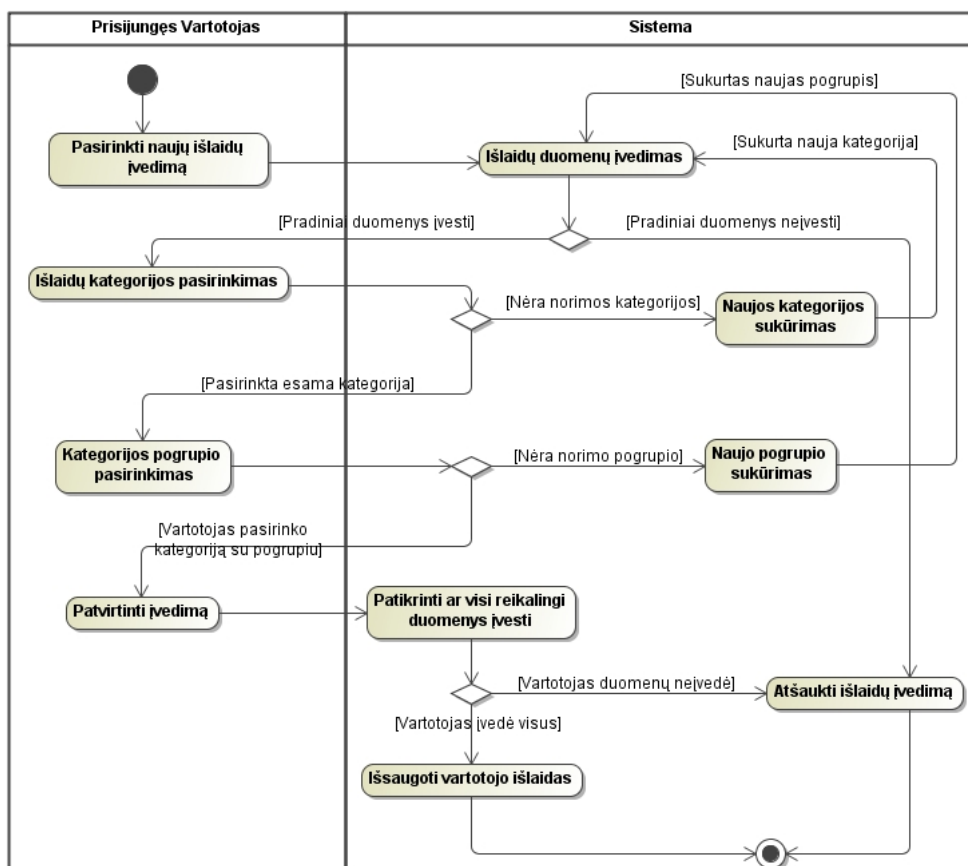


3.11 Pav. Vartotojo prisijungimo būsenos diagrama

Šiame paveikslėlyje (3.11 Pav.) matosi pagrindinės vartotojo būsenos sistemoje, ir kaip šios keičiasi priklausomai nuo vartotojo atliekamų veiksmų.

3.10.2. Veiklos diagramos

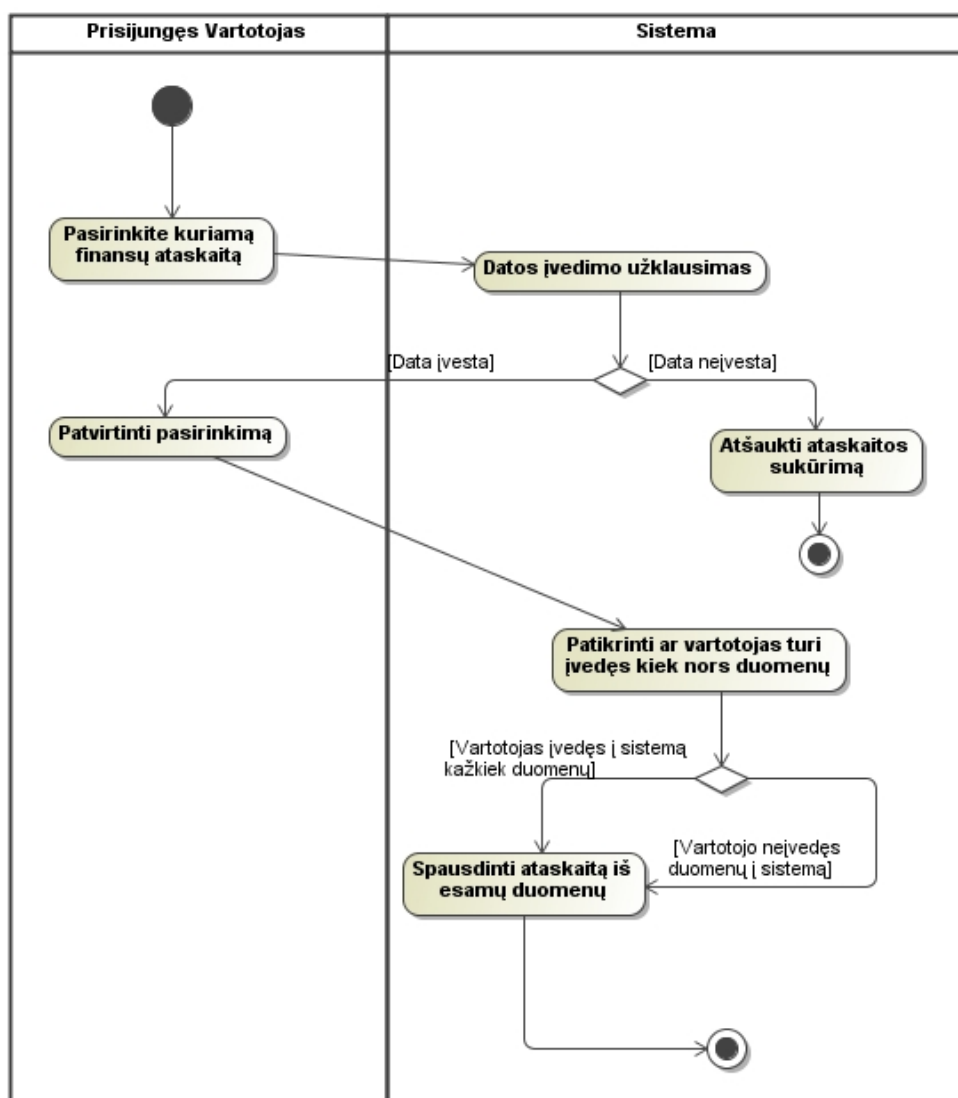
Žemiau pateiktame (3.12 Pav.) paveikslėlyje pavaizduota prisijungusio vartotojo atliekama veikla sistemoje, įvedant naujas išlaidas. Kadangi panašūs veiksmai atliekami ir su pajamų bei mokesčių įvedimu, todėl ši diagrama tinka ir minėtoms veikloms atvaizduoti.



3.12 Pav. Išlaidų įvedimo veiklos diagrama

Pateiktoje išlaidų įvedimo veiklos diagramoje, matome, kad vartotojas norėdamas įvesti naujas išlaidas į sistemą, pirmiausia turi: įvesti pradinis duomenis, tokius kaip išlaidų pavadinimas ir data. Vėliau, šiems išlaidoms priskirti kategoriją ir jos pogrupį (subkategoriją). Jei reikalingų kategorijų vartotojas neranda sistemoje, jas galima susikurti pačiam. Taigi, pasirinkus norimą išlaidų kategoriją, vartotojas patvirtina įvedimą, o sistema patikrina įvestus duomenis, ir jei viskas tvarkoje, juos išsaugo.

Žemiau pateiktame (3.13 Pav.) paveikslėlyje pavaizduota sistemos vartotojo atliekama veikla sistemoje, kuriant naują finansų ataskaitą.

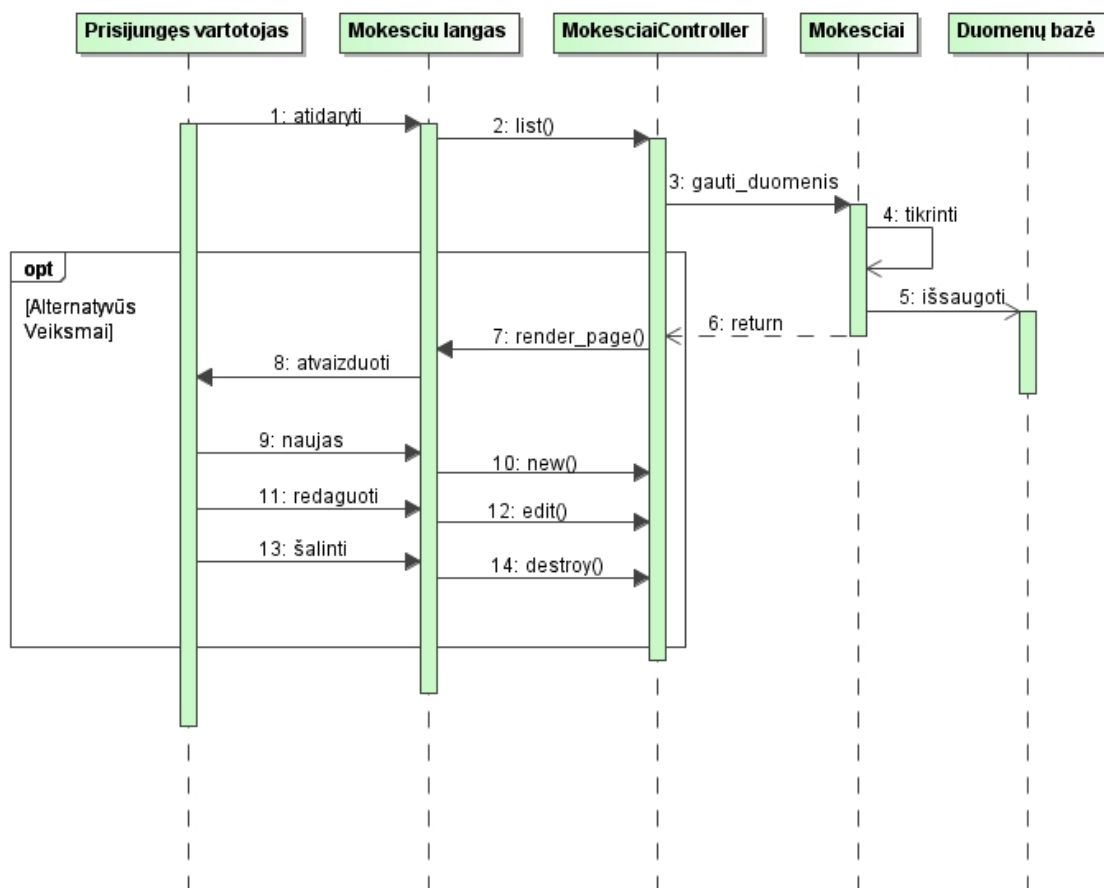


3.13 Pav. Ataskaitų sukūrimo veiklos diagrama

Iš šios diagramos matome, kad pradėjus ataskaitos sukūrimo vykdymą, iš pradžių prašoma įvesti ataskaitos formavimo pradžios ir pabaigos datas. Tai įvykdžius, sistema toliau tikrina ar vartotojas turi sukaupęs kokių nors duomenų iš norimos formuoti ataskaitos tipo. Bet koku atveju atvaizduojama tuščia arba ne ataskaita.

3.10.3. Sąveikos diagramos

Čia bus pateiktos vieno iš panaudojimo atvejų sekų diagrama. Kadangi, visų likusių panaudos atvejų sekų diagramos, dėl naudojamo MVC programavimo šablono, atrodo panašiai, tai čia pateiksime tik mokesčių tvarkymo sekos diagramą (3.14 Pav.).

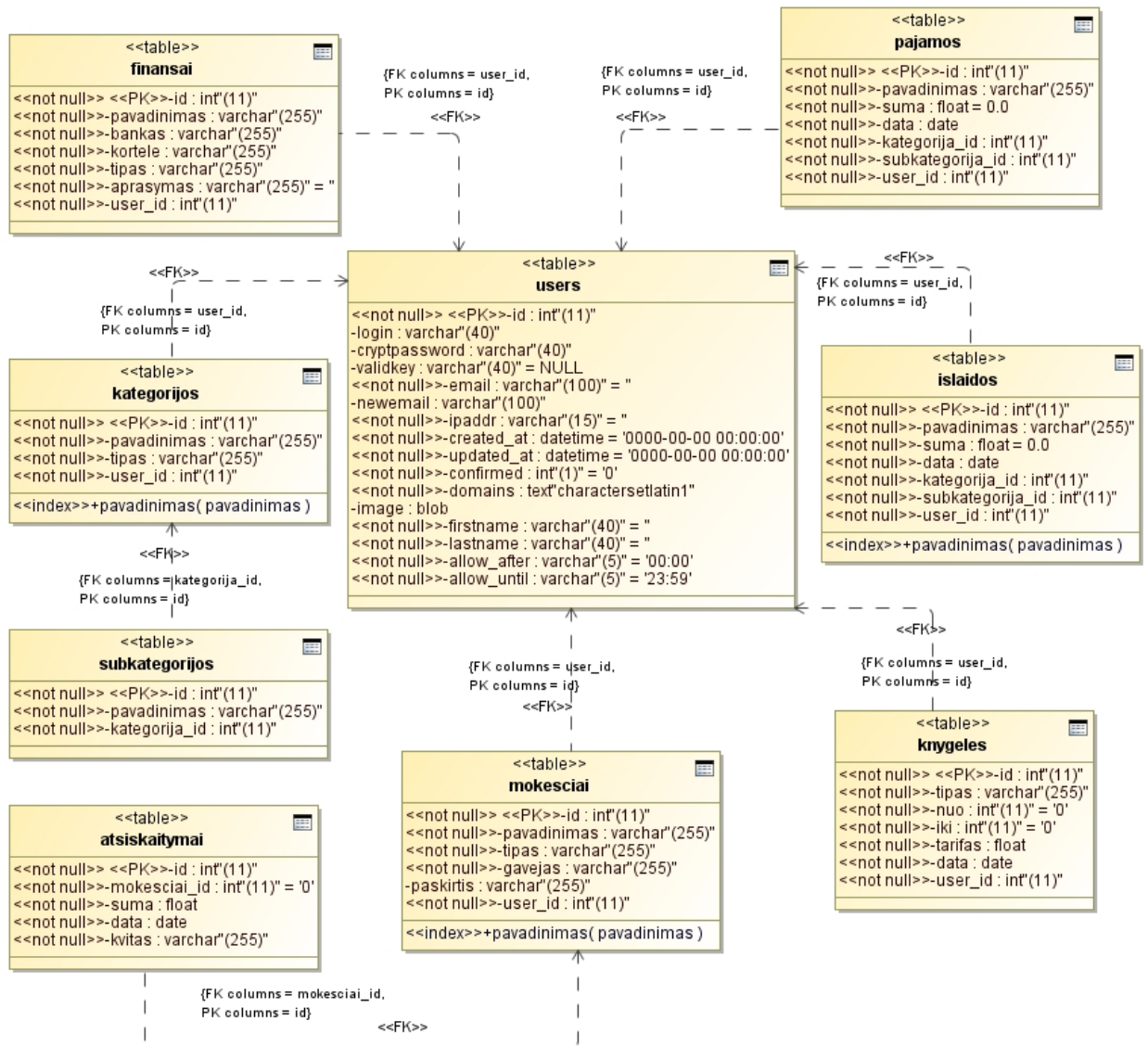


3.14 Pav. Mokesčių tvarkymo sąveikos (seku) diagrama

Iš šios sekų diagramos matome, visas su šiuo veiksmu susijusias klases ir jų bendravimo eiliškumą. Vartotojas mokesčių tvarkymo metu gali ne tik peržiūrėti įrašus bet juos ir sukurti/redaguoti/šalinti. Atlikus bet kurį iš šių veiksmų, mokesčių valdiklis pateikia vartotojui atnaujintą informaciją. Prieš atnaujinant informaciją, valdiklis dar kreipiasi į mokesčių duomenų klasę, kuri įvykdo duomenų gavimo/redagavimo/pašalinimo veiksmus iš duomenų bazės.

3.11. Duomenų vaizdas

Žemiau pateikiamas asmeninių finansų planavimo ir valdymo sistemos duomenų bazės modelio vaizdas:



3.15 Pav. Duomenų bazės modelis

Duomenų bazės modelyje esančių esybių aprašymai pateikti sekančioje lentelėje.

3.3 Lentelė. Duomenų bazės modelio esybės

| Lentelė | Aprašymas |
|--------------------|--|
| Vartotojai (Users) | Saugoma informacija apie vartotojus. |
| Finansai | Saugoma informacija apie vartotojo sąskaitas. |
| Kategorijos | Saugoma informacija apie pajamų-išlaidų kategorijas. |
| Subkategorijos | Saugoma informacija apie kategorijų subkategorijas. |
| Pajamos | Saugoma informacija apie įvestas vartotojo pajamas. |

| Lentelė | Aprašymas |
|----------------|--|
| Išlaidos | Saugoma informacija apie įvestas vartotojo išlaidas. |
| Knygelė | Saugoma informacija apie atsiskaitymo knygelės įrašus. |
| Mokesčiai | Saugoma informacija apie vartotojo mokesčius. |
| Atsiskaitymai | Saugoma informacija apie vartotojo mokesčių atsiskaitymus. |

3.12. Praktinis darbo rezultatas

Užbaigus darbą gautas galutinis programinės įrangos produktas, kuris pristatytas užsakovui ir savo ruožtu toliau sėkmingai tobulinamas ir eksploatuojamas pas užsakovą.

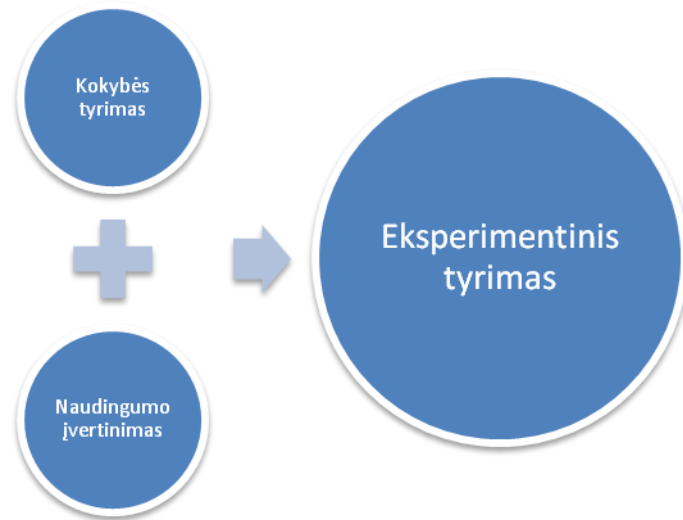
Taip pat, darbo vykdymo metu buvo sukurta programinės įrangos techninė dokumentacija. Visą dokumentų sąrašą sudaro:

- projekto paraiškos dokumentas;
- projektavimo metodologijos ir technologijos analizė;
- detalus projekto planas;
- reikalavimų specifikacija;
- architektūros specifikaciją;
- detalios architektūros specifikacija;
- programinės įrangos licencija;
- testavimo planas;
- kokybės analizės dokumentas;
- vartotojo dokumentacija ir kiti pagalbinių dokumentai.

Dalis šių dokumentų medžiagos panaudota ir šiame darbe. Taip pat darbų pradžioje buvo sukurta projekto informacinė sistema, kurioje yra saugoma visa su projektu susijusi dokumentacija.

4. SISTEMOS KOKYBĖS IR NAUDINGUMO TYRIMAS

Šiame skyriuje pateikiama magistrinio darbo metu sukurtos programinės įrangos kokybės ir naudingumo analizė. Aprašomi atlikti asmeninių finansų planavimo ir valdymo sistemos eksperimentiniai tyrimai ir gauti rezultatai.



4.1 Pav. Eksperimentinio tyrimo aspektai

Kaip parodyta aukščiau pateiktame paveiksliuke (4.1 Pav.), programų sistemos eksperimentinis tyrimas bus atliekamas dviem aspektais: pagal 4.2. skyriuje pateiktas kokybės vertinimo charakteristikas ir atliekant realizuotos sistemos naudingumo įvertinimą.

4.1. Kokybės analizės tikslai

- **Aptikti klaidas funkcionavime, logikoje, realizacijoje**

Kokybės analizės darbai atliekami pirmiausia tam, kad įsitikinti, jog programa veikia pagal pradinę sistemos specifikaciją. Reikia patikrinti, ar programuotojo sudarytas sistemos veikimo vaizdas sutampa su pradiniais užsakovo reikalavimais. Šiame etape tikrinamas konkretus sistemos funkcijų veikimas. Tikrinamos logikos klaidos ir atliekami bendro pobūdžio darbo su programa veiksmai. Aptiktos klaidos registruojamos, išanalizuojamos ir joms pateikiami pataisymai.

- **Patikrinti ar programų sistema atitinka reikalavimų specifikaciją**

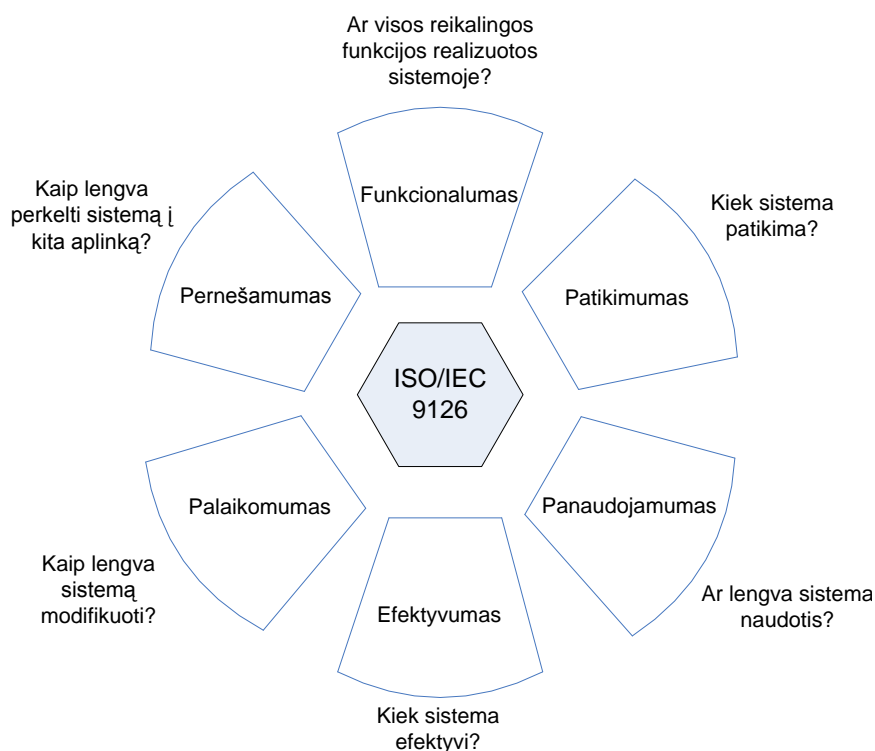
Patikrinus sistemos logiką ir realizaciją, reikia atlikti aukštesnio lygio sistemos kontrolės darbus. Sekančiame etape tikrinama ar programa atitinka reikalavimų specifikacijoje iškeltus reikalavimus. Tikrinama, ar realizuoti visi pradiniai panaudojimo atvejai. Svarbu kuo anksčiau projektavimo eigoje aptikti neatikimus su reikalavimų specifikacija, nes ankstyvoje stadijoje pokyčiams įgyvendinti reikia mažiausiai pastangų [6].

- **Įsitikinti, ar programų sistema sukurta pagal standartus**

Kai pirmieji pagrindiniai kokybės kontrolės etapai jau atlikti belieka tik patikrinti ar programinė įranga atitinka tam tikrus keliamus reikalavimus. Pavyzdžiui, banke programinei įrangai keliami reikalavimai žymiai griežtesni ir turi būtinai atitikti tam tikrus bankams keliamus standartus. Todėl, sekančiame 4.2 skyrelyje, aptarsime programinės įrangos kokybės vertinimo kriterijus.

4.2. Programinės įrangos kokybės vertinimo charakteristikos

Sukurtos programinės įrangos kokybei įvertinti, naudosime ISO/IEC 9126 standarto apibrėžtas vertinimo charakteristikas. Šio standarto tikslas – pateikti karkasą programinės įrangos kokybei nustatyti. ISO/IEC 9126 standartas nenurodo, kokie turi būti programinės įrangos reikalavimai, bet jis apibrėžia kokybės modelį, kuris gali būti pritaikomas bet kokiai programinei įrangai [14]. IS produktų kokybė susijusi su būdingomis sistemų savybėmis, kurias patiria vartotojai. Žemiau esančiame paveikslėlyje (4.2 Pav.) pateiktos šešios pagrindinės šį modelį sudarančios produkto kokybės charakteristikos.



4.2 Pav. Kokybės vertinimo modelis

- Funkcionalumas – tai kokybės faktorius, atspindintis programos atitikimą specifikacijai ir vartotojo poreikių tenkinimą;
- Patikimumas – parodo sistemos naudojimo patikimumą;
- Panaudojamumas – faktorius nusakantis sistemos panaudojimo lengvumą;

- Efektyvumas – apibrėžia kompiuterinius resursus reikalingus programinei įrangai;
- Palaikomumas – nusako pastangas, reikalingas nustatyti ir ištaisyti sistemos klaidas;
- Pernešamumas – tai pastangos, reikalingos programinei įrangai perkelti iš vienos techninės konfigūracijos į kitą ir/ar iš vienos operacinės sistemos į kitą.

Šių charakteristikų skirstymas į smulkesnes subcharakteristikas pateikiamas žemiau esančioje lentelėje (4.1 Lentelė).

4.1 Lentelė. Sistemos kokybės vertinimo charakteristikos

| Charakteristika | Subcharakteristika | Paiškinimas |
|-----------------------|----------------------|--|
| Funkcionalumas | Tinkamumas | Programinės įrangos atributai nurodantys esančias ir teisingas funkcijas reikalingas tam tikrai užduočiai. |
| | Tikslumas | Programinės įrangos atributai nurodantys gražinamus teisingus ar sutartus rezultatus, ar efektus. |
| | Bendradarbiavimas | Programinės įrangos atributai nurodantys galimybes sistemai sąveikauti su kitomis apibrėžtomis sistemomis. |
| | Atitikimas | Programinės įrangos atributai, kurie padaro programinę įrangą atitinkančią standartus, konvencijas ar įstatymus ir panašius aprašus. |
| | Apsauga | Programinės įrangos atributai saugantys nuo neteisėto panaudojimo ir priėjimo, nesvarbu tyčinio ar ne, prie programos ar duomenų. |
| Patikimumas | Užbaigtumas | Programinės įrangos atributai nurodantys klaidų dažnį programinėje įrangoje. |
| | Tolerancija klaidoms | Programinės įrangos atributai nurodantys sistemos galimybes išlaikyti tam tikrą veikimo lygį įvykus klaidoms. |
| | Atstatomumas | Programinės įrangos atributai nurodantys galimybę atstatyti normalų veikimo darbą ir atstatyti duomenis, tiesiogiai paveiktus klaidų bei laiką ir pastangas reikalingas tai atlikti. |
| Panaudojamumas | Suprantamumas | Programinės įrangos atributai nurodantys vartotojui reikalingas pastangas, loginių principų ir jų pritaikymo supratimui. |
| | Išmokstamumas | Programinės įrangos atributai nurodantys vartotojui reikalingas pastangas, išmokti naudotis programa. |
| | Veiklumas | Programinės įrangos atributai nurodantys vartotojo pastangas atliekant operacijas ir operacijų kontrolę. |
| Efektyvumas | Laikinė elgsena | Programinės įrangos atributai nurodantys sistemos veikimo laikus ir atliekamų operacijų skaičius per tam tikrą laiką. |

| Charakteristika | Subcharakteristika | Paaškinimas |
|---------------------|--------------------|--|
| | Resursų naudojimas | Programinės įrangos atributai nurodantys sunaudojamų resursų kiekį ir kiek laiko jie yra naudojami. |
| Palaikomumas | Analizuojamumas | Programinės įrangos atributai nurodantys reikalingą pastangų kiekį siekiant nustatyti netikslumus, klaidų priežastis ar identifikuoti dalis, kurias reikia modifikuoti. |
| | Keičiamumas | Programinės įrangos atributai nurodantys pastangų kiekį reikalingą modifikavimui, klaidų pašalinimui ar aplinkos pakeitimams. |
| | Stabilumas | Programinės įrangos atributai nurodantys riziką apie nenumatytus efektus atliekant modifikavimą. |
| | Testuojamumas | Programinės įrangos atributai nurodantys pastangas reikalingas ištestuoti programinę įrangą. |
| Pernešamumas | Pritaikomumas | Programinės įrangos atributai nurodantys galimybes pritaikyti skirtingoms aplinkoms nenaudojant jokių kitokių priemonių kaip tos kurios priklauso tai programinei įrangai. |
| | Įdiegiamumas | Programinės įrangos atributai nurodantys pastangas reikalingas instaliuoti programinę įrangą specifikuotoje aplinkoje. |
| | Atitikimas | Programinės įrangos atributai kurie leidžia programinei įrangai laikytis standartų ir konvencijų susijusių su pernešamumu. |
| | Pakeičiamumas | Programinės įrangos atributai nurodantys galimybes naudoti šią programinę įrangą vietoj kitos programinės įrangos tos programinės įrangos aplinkoje. |

4.3. Sistemos kokybės tyrimas

Asmeninių finansų planavimo ir valdymo sistemos kokybės vertinimas atliktas remiantis kokybės vertinimo charakteristikomis, pagal ISO 9126 standartą. Sistemos kokybė vertinta jos eksploatacijos metu, remiantis sistemos kūrimo pradžioje iškeltais reikalavimais, stebint sistemos veikimą ir vartotojų darbą su sistema. Sistemos kokybę vertino užsakovas.

Kokybės charakteristikos susideda iš vieno ar daugiau jų nusakančių subcharakteristikų, kurie buvo vertinami pasitelkiant skalę nuo 1 iki 9 (1-3 = žemas, 4-6 = vidutinis, 7-9 = aukštas).

Asmeninių finansų planavimo ir valdymo sistemos kokybės vertinimo rezultatai pateikiami žemiau esančioje (4.2 Lentelė) lentelėje. Remiantis nustatyta vertinimo skale, atliekamas sistemos vertinimas remiantis kiekviena subcharakteristika, o po to nustatomas bendras įvertinimas kiekvienai kokybės charakteristikai.

4.2 Lentelė. Asmeninių finansų planavimo ir valdymo sistemos kokybės vertinimas

| Kokybės charakteristika | | | |
|--------------------------------|---------------------------|--|--------------------|
| Funkcionalumas | | | |
| | <i>Subcharakteristika</i> | <i>Subcharakteristikos vertinimo aprašymas</i> | <i>Įvertinimas</i> |
| Tinkamumas | | Realizuotos visos reikalavimų dokumente numatytos sistemos funkcijos. | 8 |
| | | Sistema atitinka visus iškeltus funkcinis reikalavimus. | 9 |
| | | Sistema atitinka visus iškeltus nefunkcinius reikalavimus. | 8 |
| Tikslumas | | Sistemoje atliekami skaičiuoklės skaičiavimai pagal formules yra teisingi. | 9 |
| | | Įvairūs sistemoje atliekami finansų skaičiavimai yra teisingi. | 9 |
| Bendradarbiavimas | | Sistemoje, kai kurios galimybės yra, tačiau reikalauja nemažai projektavimo išlaidų. | 5 |
| Atitikimas | | Sistemos vartotojo sąsaja atitinka keliamus Lietuvių kalbos reikalavimus. | 9 |
| | | Sistema užtikrina privalomą asmeninių duomenų saugumą bei privatumą. | 8 |
| Apsauga | | Sistemos administratorius turi priėjimą tik prie viešų vartotojo prisijungimo duomenų. | 7 |
| | | Duomenų bazės prieiga apsaugota administratoriaus slaptažodžiu. | 9 |
| | | Sistema funkcionuoja tik po sėkmingos vartotojo identifikacijos. | 9 |
| | | Vartotojų slaptažodžiai yra užkoduoti. | 9 |
| <i>Bendras įvertinimas</i> | | | 7,9 |
| Patikimumas | | | |
| | <i>Subcharakteristika</i> | <i>Subcharakteristikos vertinimo aprašymas</i> | <i>Įvertinimas</i> |
| Užbaigtumas | | Sistemos sutrikimai ir klaidos ne daugiau kaip 5 per mėnesį. | 8 |
| | | Visos sistemos klaidos registruojamos, sistemos sutrikimai pranešami vartotojui. | 8 |
| Tolerancija klaidoms | | Klaida, kurioje nors sistemos dalyje arba vartotojo sąsajoje, neapriboja galimybių naudotis likusia sistemos dalimi. | 7 |
| | | Sistemai sutrikus, jos atstatymo laikas yra neilgesnis nei 1 val. | 8 |
| Atstatomumas | | Atsiradus klaidai sistemoje, vartotojo duomenys neišgadinami. | 9 |
| | | | |
| <i>Bendras įvertinimas</i> | | | 8,0 |
| Panaudojamumas | | | |
| | <i>Subcharakteristika</i> | <i>Subcharakteristikos vertinimo aprašymas</i> | <i>Įvertinimas</i> |
| Suprantamumas | | Bendra vartotojo sąsaja yra intuityvi. Tie patys vartotojo sąsajos valdymo principai išlaikomi visoje sistemoje. | 8 |
| | | | |

| | | | |
|------------------------------|---------------------------|--|--------------------|
| | | Pateikiami paaiškinimai vartotojo sąsajoje. | 7 |
| Išmokstamumas | | Susipažinimas su vartotojo sąsaja ir sistemos funkcijomis užtrunka iki 1 valandos. | 9 |
| | | Sistema turi detalių vartotojo vadovą. | 9 |
| Veiklumas | | Sistemoje naudojamas lengvai suprantamas, intuityvus ir vienodas vartotojo sąsajos dizainas. | 8 |
| | | Vartotojui pateikiami pranešami apie atliekamus veiksmus, jų sėkmingumą ir kt. | 8 |
| <i>Bendras įvertinimas</i> | | | 8,2 |
| Efektyvumas | | | |
| | <i>Subcharakteristika</i> | <i>Subcharakteristikos vertinimo aprašymas</i> | <i>Įvertinimas</i> |
| | Laikinė elgsena | Visų sistemos funkcijų vykdymas yra greitas. | 8 |
| | Resursų naudojimas | Normalus sistemos vartojimas reikalauja labai nedaug sistemos resursų. | 9 |
| <i>Bendras įvertinimas</i> | | | 8,5 |
| Palaikomumas | | | |
| | <i>Subcharakteristika</i> | <i>Subcharakteristikos vertinimo aprašymas</i> | <i>Įvertinimas</i> |
| | Analizuojamumas | Prie visų klaidų pranešimų pateikiama informacija, kurioje vietoje ji įvyko. | 8 |
| Keičiamumas | | Norint atlikti pakeitimus sistemoje nereikia būti susipažinus su visa sistemos realizacija. | 5 |
| | | Dalį sistemos veikimo galima keisti per parametrus, nereikalaujant sistemos perkrovimo. | 6 |
| Stabilumas | | Pakeitimai vienoje sistemos dalyje neiššaukia pakeitimų ar klaidų kitoje. | 5 |
| Testuojamumas | | Automatiniai sistemos testai apima didžiąją dalį sistemos funkcijų. | 5 |
| | | Sistema nereikalauja specialios aplinkos sukongūravimo prieš pradėdant jos testavimą. | 7 |
| <i>Bendras įvertinimas</i> | | | 6,1 |
| Pernešamumas | | | |
| | <i>Subcharakteristika</i> | <i>Subcharakteristikos vertinimo aprašymas</i> | <i>Įvertinimas</i> |
| | Prisitaikomumas | Sistema gali būti naudojama įvairiose operacinės sistemos aplinkose. | 9 |
| Įdiegiamumas | | Sistemos dalies diegimo sudėtingumas. | 5 |
| | | Sistemos parametrų keitimo sudėtingumas. | 6 |
| Atitikimas | | Naudojami standartiniai protokolai duomenų mainams sistemos viduje. | 9 |
| Pakeičiamumas | | Dalis sistemos funkcijų gali būti pritaikomos alternatyviuose sprendimuose. | 7 |
| <i>Bendras įvertinimas</i> | | | 7,6 |
| Galutinis įvertinimas | | | 7,7 |

Kaip matome iš aukščiau pateiktos lentelės, sistemos kokybės vertinimas buvo atliktas įvertinus 21 subcharakteristikų, kurių reikšmės nustatomos pagal vieną ar kelis atributus.

Kokybės subcharakteristikos reikšmė (ji lentelėje nepateikiama) nustatoma pagal tokią formulę (1):

$$Subcharakteristikos_įvertinimas = \frac{\sum Subchar_atributo_reikšmė}{Subchar_atributų_kiekis}. \quad (1)$$

Kokybės charakteristikos įvertinimas (paskutinė tos charakteristikos eilutė) paskaičiuojamas pagal 2-ą formulę:

$$Charakteristikos_įvertinimas = \frac{\sum Subcharakteristikos_įvertinimas}{Subcharakteristikų_kiekis}. \quad (2)$$

Sistemos kokybės galutinis įvertinimas buvo paskaičiuotas pagal 3-ą formulę:

$$Bendras_įvertinimas = \frac{\sum Charakteristikos_įvertinimas}{Charakteristikų_kiekis}. \quad (3)$$

Atlikus visus skaičiavimus gautas bendras sistemos kokybės įvertinimas – 7,7. Galime daryti išvadą, kad galutinės sistemos kokybė yra aukštesnė už vidutinišką ir atitinka jai iškeltus reikalavimus. Sistemos vertinimo metu, nebuvo aptikta nė viena rimta problema, kuri keltų grėsmę sistemos kokybei.

Pažvelgus į atskirus kiekvieno kokybės kriterijaus rezultatus pastebime, kad blogiausiai įvertintas sistemos palaikomumo kriterijus. Kaip ir daugumoje programinės įrangos sistemų, jos palaikomumui reikalingas ypatingas dėmesys. Neveltui, programinės įrangos palaikymo stadija yra viena svarbiausių sistemos gyvavimo cikle. Dėl šios priežasties, toliau šiame darbe bus kalbama ne tik apie sistemos funkcionalumo tobulinimą, bet apie sistemos palaikomumo gerinimą.

Kita problema yra ta, kad dalis sistemos funkcijų yra gana „surištos“, t.y. pakeitimai vienoje sistemos dalyje iššaukia pakeitimus ar klaidas kitoje. Dėl šios priežasties, sistemos stabilumo kriterijus buvo įvertintas tik 5 balais. Taip pat galime atsižvelgti, kad nėra visas sistemos funkcijas patikrinančių automatinių sistemos testų. Šią problemą, dalinai gali padėti išspręsti, testų sukūrimas pagrindiniams sistemos moduliams.

4.4. Realizuotos sistemos naudingumo įvertinimas

Šiame skyrelyje atliekamas realizuotos asmeninių finansų planavimo ir valdymo sistemos (sutrump. *AFPVS*) naudingumo įvertinimas. Vertinimas buvo atliktas lyginant realizuotą sistemą su 2.3.6 skyrelyje analizuotomis egzistuojančiomis sistemomis.

Sistemos naudingumo įvertinimas remiasi sistemos funkcijų reikšmingumo įvertinimu, kuris gautas atlikus sistemos vartotojų apklausą (1 priedas. **Apklausos rezultatai**). Kiekviena eilutė (testavimo kriterijus) įvertinta balu. Galutinis rezultatas gaunamas kiekvieną, padauginus iš svorio (svarbos) ir gautus skaičius sudėjus. Svarba buvo parinkta pagal reikšmingumą vartotojui.

Žemiau pateikiama lentelė (4.3 Lentelė), kurioje palyginamas egzistuojančių sistemų ir mūsų realizuotos sistemos naudingumas vartotojui.

4.3 Lentelė. Egzistuojančių sistemų naudingumo palyginimas su realizuota sistema

| Programa ir jos vertinimas | Svarba | AcceMoney Lite | Microsoft Money | Butas 2006 | Namų buhalteris | Intuit Quicken | Moneydance | AFPVS |
|--|------------|----------------|-----------------|-------------|-----------------|----------------|-------------|-------------|
| Diegimas ir pagalba | 22% | 7,55 | 3,14 | 8,23 | 7,32 | 3,68 | 3,50 | 9,46 |
| Lietuviška versija | 10% | 10 | 1 | 10 | 10 | 1 | 1 | 9,8 |
| Spausdintas vartotojo vadovas | 4% | 1 | 1 | 1 | 1 | 1 | 1 | 10 |
| Įdiegimas | 3% | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Užimama vieta kietajame diske | 1% | 8 | 5 | 10 | 10 | 5 | 8 | 10 |
| Pagalba programoje | 2% | 5 | 5 | 9 | 1 | 9 | 5 | 8 |
| Pagalba internete | 1% | 4 | 5 | 9 | 5 | 9 | 5 | 5 |
| Išdiegimas | 1% | 10 | 5 | 10 | 10 | 5 | 10 | 10 |
| Programos valdymas ir galimybės | 23% | 7,61 | 7,52 | 8,35 | 7,61 | 9,78 | 8,04 | 7,95 |
| Valdymo patogumas | 10% | 8 | 8 | 8 | 10 | 10 | 7 | 8 |
| Mygtukų bei pranešimų aiškumas | 5% | 5 | 8 | 10 | 8 | 9 | 7 | 9 |
| Galimybė vesti apskaitą įvairiomis valiutomis | 3% | 10 | 1 | 10 | 1 | 10 | 10 | 2 |
| Duomenų apsauga slaptažodžiu | 3% | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Galimybė naudotis programa keliems vartotojams | 2% | 5 | 10 | 1 | 1 | 10 | 10 | 10 |
| Banko sąskaitos apskaita | 18% | 2,33 | 5,67 | 1 | 1 | 5,89 | 5,89 | 1,00 |
| Galimybė gauti duomenis iš mūsų el. bankininkystės sistemos | 10% | 1 | 3 | 1 | 1 | 3 | 3 | 1 |
| Kelių skirtingų sąskaitų apskaitą, galima vesti? | 4% | 2 | 10 | 1 | 1 | 10 | 10 | 1 |
| Sąskaitos įplaukų bei išlaidų rodymas | 4% | 6 | 8 | 1 | 1 | 9 | 9 | 1 |
| Namų biudžeto planavimo funkcijos | 22% | 4,68 | 6,59 | 8,09 | 6,05 | 6,59 | 6,59 | 7,32 |
| Ataskaitos apie mūsų įplaukas / išlaidas bei finansinę būklę | 3% | 6 | 8 | 10 | 6 | 8 | 7 | 8 |
| Išlaidų kategorijų skaičius | 3% | 10 | 10 | 10 | 3 | 10 | 10 | 9 |
| Priminimai apie išsiskolinimus bei reguliarius ir atliktus mokėjimus | 3% | 5 | 8 | 5 | 1 | 8 | 9 | 4 |
| Skolininkų ir skolintojų sąrašas | 3% | 10 | 10 | 10 | 10 | 10 | 10 | 9 |
| Perspėjimas apie viršytą nustatytą išlaidų limitą | 3% | 1 | 10 | 1 | 1 | 10 | 10 | 1 |

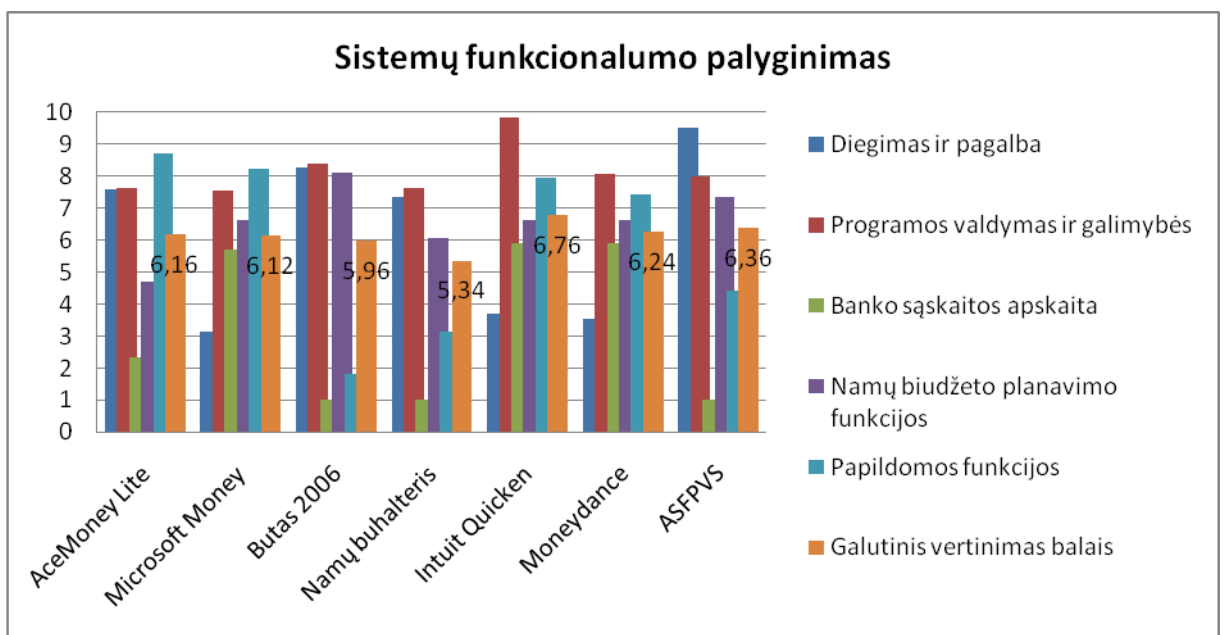
| | | | | | | | | |
|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Atsiskaitymų už paslaugas (elektrą, dujas, vandenį) skaičiuoklė | 7% | 1 | 1 | 10 | 10 | 1 | 1 | 10 |
| Papildomos funkcijos | 15% | 8,67 | 8,20 | 1,80 | 3,13 | 7,93 | 7,40 | 4,41 |
| Vertybinių popierių bei kitų investicijų apskaita | 5% | 10 | 10 | 1 | 1 | 10 | 10 | 1 |
| Galimybė išskelti duomenis į kitą formatą | 4% | 5 | 10 | 4 | 9 | 9 | 7 | 7 |
| Galimybė kuri savo pačių ataskaitas | 3% | 10 | 10 | 1 | 1 | 10 | 1 | 6 |
| Paskolų ir kitos skaičiuoklės | 3% | 10 | 1 | 1 | 1 | 1 | 10 | 5 |
| Galutinis vertinimas balais | 100% | 6,16 | 6,12 | 5,96 | 5,34 | 6,76 | 6,24 | 6,36 |

Svorinis kriterijų įvertinimas – galutinis rezultatas gaunamas kiekvieną kriterijų padauginus iš svorio (svarbos) ir gautus skaičius sudėjus, formulės (4,5):

$$Svorinis_kriteriju_įvertinimas = \sum (svorio_reikšmė \times kriterijaus_reikšmė). \quad (4)$$

$$Galutinis_įvertinimas = \sum (svorinis_kriteriju_įvertinimas). \quad (5)$$

Žemiau pateiktame grafike (4.3 Pav.), parodomi visų analizuojamų asmeninių finansų surinkti svoriniai kriterijų įvertinimai.

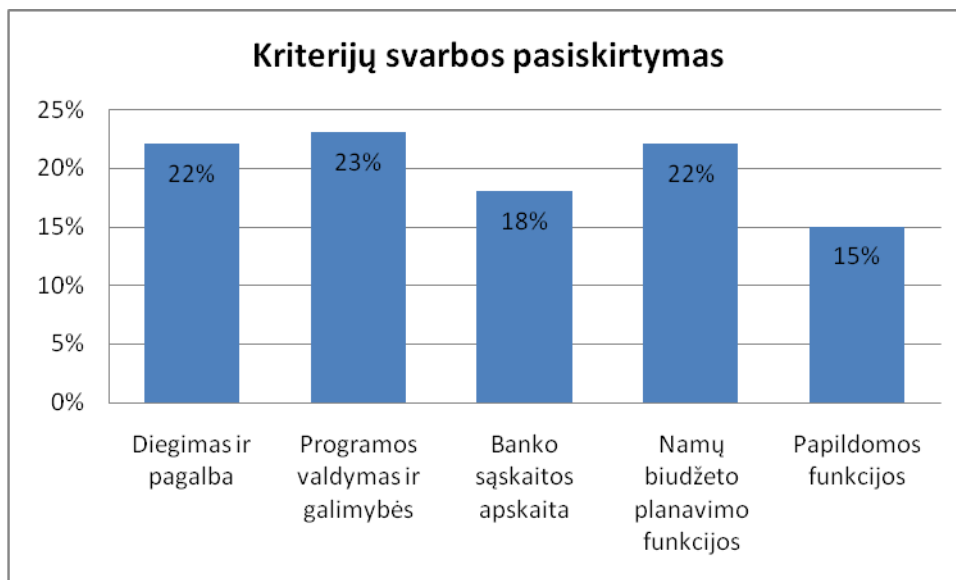


4.3 Pav. Asmeninių finansų tvarkymo sistemų įvertinimų palyginimas

Remiantis gautais palyginimo rezultatais galima teigti, kad visos sistemos išlaiko labai panašų funkcionalumo lygį. Skirtumai tarp galutinių rezultatų yra labai maži. Tačiau skirtingose funkcijų grupėse gaunami rezultatai kai kur akivaizdžiai skiriasi (4.3 Pav.). Taip yra dėl to, kad kai kuriose sistemose yra nerealizuotos tam tikros funkcijos (pvz. Banko sąskaitos apskaita). Mūsų ASFPVS rezultatai ir parodo, kad silpniausia vieta, funkcionalumo atžvilgiu, ir yra banko sąskaitos apskaitos bei papildomų funkcijų nebuvimas.

Nepaisant to, pateiktame grafike matyti, kad mūsų realizuota sistema turi vieną iš didžiausių įvertinimų (6,36 balo). Didesnį įvertinimą turi tik viena sistema: „Intuit Quicken“ (6,76). Kaip ir minėjome, visos įvertintos sistemos, savo bendru funkcionalumu yra gana panašios, t.y. jos visos atlieka savo darbą vidutiniškai gerai.

Žemiau pateiktame grafike (4.4 Pav.), matyti svarbiausi bei didžiausią įtaką turėję vertinimo kriterijai.



4.4 Pav. Kriterijų svarbos pasiskirstymas

Pateiktame grafike matyti, kad didžiausią įtaką rezultatui turėjo programos valdymo paprastumas ir jos galimybės (23%). Taip pat, nemažą įtaką sudarė ir „Diegimas ir pagalba“ su „Namų biudžeto planavimo funkcijos“ (22%). Kalbant apie pastarąją, verta paminėti, kad tai viena iš svarbiausių asmeninių finansų planavimo ir valdymo sistemų funkcijų. Šioje, randasi pagrindinės finansų valdymo funkcijos. Sugrįžtant prie „Diegimo ir pagalba“, verta pastebėti, kad šioje kategorijoje, didžiausią įtaką rezultatai turėjo lietuviškos sąsajos buvimas/nebuvimas. Tai tik įrodo, ką mes akcentavome viso darbo metu, kad produkto pritaikymas Lietuvos rinkai yra būtinas, norint sistemai konkuruoti šioje rinkoje.

Reikia pastebėti, kad nors didžiosios asmeninių finansų planavimo programos (*Microsoft, Intuit ir kt.*) turi tikrai daug įvairių funkcijų ir galimybių, tačiau jas vertinant pagal (4.4 Pav.) pateiktus penkis pagrindinius kriterijus, jų naudingumas susilygino su lietuviškas sąsajas turinčiomis sistemomis. Taip atsitiko, dėl tų pačių aukščiau paminėtų priežasčių.

Nustatyta, kad lietuviškos asmeninių finansų sistemos neturi galimybių bendrauti su mūsų el. bankininkystės sistemomis, todėl ir banko sąskaitų apskaitos funkcijų jose nėra. Analizuojant užsieniškąs programas, nustatyta, kad jose atitinkamai trūksta atsiskaitymų už paslaugas (elektrą, dujas, vandenį) skaičiuoklės.

Todėl, norint kad produktas vartotojams būtų kuo naudingesnis, reikia, kad jis pasižymėtų tokiomis pagrindinėmis savybėmis:

- Lietuviška versija;
- Aiškia vartotojo sąsaja;
- Patogumu valdyti;
- Galimybe gauti duomenis iš mūsų el. bankininkystės sistemos;
- Turėtų atsiskaitymų už paslaugas (elektrą, dujas, vandenį) skaičiuoklę.

4.5. Tolimesnio sistemos tobulinimo ir plėtojimo galimybės

Atsižvelgiant į tai, kad asmeninių finansų planavimo ir valdymo sistema anksčiau ar vėliau bus reikalinga ateityje, galime iškelti tam tikrus naujus vartotojo reikalavimus ar prioritetinės tobulinimo galimybes:

- **Vartotojo sąsajos tobulinimas.**

Gavus atsiliepimus iš sistemą bandžusių vartotojų, vartotojo sąsaja pasirodė šiek tiek per „gili“, t.y. turinti nevieną žemyn einantį lygį. Pvz. norint pasiekti ir įvesti naują atsiskaitymą už komunalines paslaugas, reikia eiti: „*Mokesčiai->Atsiskaitymo knygelė->Naujas įrašas*“. Taip sukūrus naują įrašą, reikia vėl pakilti vienu lygiu aukščiau, kad pamatyti tą sukurtą įrašą. Taigi, patobulinimo tikslas būtų pataisyti esamą sąsają, sumažinant lygių skaičių. Tai galima atlikti pasinaudojus nukreipimo funkcijomis, pvz. po naujo įrašo sukūrimo, vartotojas būtų automatiškai perkeliamas vienu lygiu aukščiau. Tokiu būdu vartotojams nereikėtų spaudinėti papildomai sugrįžimo į ankstesnį lygį mygtukų. Tokie pakeitimai neturėtų įtakos sistemos architektūrai.

- **Patogesnę atsiskaitymo knygelę.**

Išanalizavus vartotojo darbo su atsiskaitymo knygele išpročius, galimas patobulinimas būtų šios knygelės struktūros atvaizdavimo pagerinimas. Dabar vartotojas vienu metu gali pildyti tik vieną šios knygelės įrašo tipą. Atlikus pakeitimus šioje dalyje, vartotojas galėtų patogiau pildyti visą atsiskaitymo knygelę, nepriklausomai nuo to ar tai dujos, elektra ar vanduo. Tačiau, tokie pakeitimai įneštų pasikeitimų sistemos architektūroje, reikėtų perprojektuoti ir pakeisti duomenų bazės lenteles, pakeisti už atsiskaitymo knygelės apdorojimą atsakingas klases. Šiuo klausimu reikėtų atlikti detalesnę analizę, iškeliant klausimą ar būtinai to reikia?

- **Bendradarbiaujančios sistemos versija.**

Kadangi, mūsų sistema veikia kaip interneto puslapis, tai vartotojams neturintiems prieigos prie interneto, sistema yra nepasiekiamas. Todėl užsakovas vėliau pateikė naują

reikalavimą sistemai: „Sinchronizacija su kita finansų programa“. Tačiau, artėjant šio projekto realizavimo pabaigos terminui ir neturint bendradarbiavimo su kita sistema sąsajos specifikacijos, šis reikalavimas nebuvo įtrauktas į pagrindinius reikalavimus. Todėl ateityje, siekiant patobulinti ir palaikyti šią sistemą, reikėtų įgyvendinti šį reikalavimą. Tą atlikus, vartotojai galėtų ne tik vesti savo finansų apskaitą internetinėje sistemoje, bet ir pasinaudoti alternatyvia kompiuteryje veikiančia programa.

- **Sistemos planavimo galimybių išplėtimas.**

Nors pagrindinė asmeninių finansų planavimo ir valdymo sistemų užduotis yra parodyti vartotojui kur jis išleidžia daugiausia pinigų, visgi pagrindinė problema lieka, kaip tas pajamas susiplanuoti. Mūsų sistemos tikslas buvo ne pakeisti žmogų planuojant savo pajamas, bet jam padėti tai padaryti. Nors esama sistema akivaizdžiai (grafiškai) parodo kur dingsta turėti pinigai, tačiau žvelgiant į sistemos perspektyvas, reikėtų vartotojui pateikti platesnes biudžeto planavimo galimybes, tokias kaip: pirkinų planavimas, išėjimo į pensiją planavimas ar vaiko studijų planavimas. Tokie sistemos patobulinimai gana aukštai pakeltų dabartinės sistemos galimybes. Tačiau tam reikia gana nemažų investicijų, o siekiant kad tokia sistema atsipirktų reikia peržvelgti ir iš naujo įvertinti sistemos gyvavimo ciklą.

- **Perėjimas prie „Ruby On Rails 2“ karkaso versijos.**

Šio projekto realizavimo metu buvo išleista naujesnė programavimo kalbos karkaso versija. Vienas iš pagrindinių patobulinimų naujoje versijoje yra saugumo pagerinimas bei duomenų validavimo funkcijų papildymas. Kadangi, mūsų sistemoje duomenų patikrinimas vyksta dažnai, tai perėjimas prie šios versijos ne tik kad užtikrintų geresnį sistemos saugumą bet ir palengvintų sistemoje įvedamų duomenų patikrinimą. Be kita ko, perėjus ant naujesnės karkaso versijos atsiveria didesnės sistemos palaikomumo pagerinimo galimybės.

5. IŠVADOS

1. Analitinėje darbo dalyje buvo sudarytas bendras asmeninių finansų planavimo ir valdymo sistemos vaizdas, išnagrinėtos pagrindinės struktūrizuotų ir saugių sistemų projektavimo aspektai, aptartas „Modelis – Vaizdas – Valdiklis“ šablonas, taip pat aptartos galimos projektavimo problemos ir jų sprendimo būdai.

2. Išanalizavus užsakovo iškeltus reikalavimus sistemai, nustatyti penkti probleminiai sistemos realizavimo uždaviniai: universalios ir paprastos sistemos struktūros sukūrimas, finansų valdymas, informacijos įvedimas/duomenų dubliavimas, vartotojų valdymas ir saugumas, integracija su administravimo posisteme. Šiems probleminiams uždaviniams išspręsti buvo pateikti jų galimi sprendimai.

3. Palyginus plačiausiai naudojamas internetines platformas, skirtas įvairių internetinių sistemų realizavimui, taip pat atsižvelgus į sistemai keliamus vartotojo reikalavimus ir numatomus poreikius jų realizavimui, pasirinkta sistemos kūrimo platforma – „Ruby On Rails“.

4. Remiantis suformuluotais projektiniais, techniniais bei probleminių uždavinių sprendimais, realizuota asmeninių finansų planavimo ir valdymo sistema. Jos architektūra suskirstyta į tris pagrindinius lygius: atvaizdavimo, valdymo ir priėjimo prie duomenų lygius. Sistemos elementai išdėstyti pasinaudojus „Modelis-Vaizdas-Valdiklis“ projektavimo šablonu.

5. Remiantis ISO/IEC 9126 standarto rekomenduojamais kriterijais buvo atliktas sistemos kokybės vertinimo tyrimas. Jo metu gautas galutinis įvertinimo balas lygus 7,7. Rezultatas rodo, kad sistema yra geros kokybės. Išnagrinėjus atskirus vertinimo kriterijus pastebėta, kad prasčiausiai įvertintas sistemos palaikomumas. Atsižvelgiant į tai, tobulinant sistemą siūloma peržiūrėti ir įvertinti jos palaikomumą.

6. Išanalizavus sistemos vartotojų pageidavimus bei atsiliepimus, gauti egzistuojančių sistemų funkcijų naudingumo svoriniai įvertinimai. Remiantis šiais įvertinimais buvo atliktas sukurtos asmeninių finansų planavimo ir valdymo sistemos naudingumo tyrimas. Sudarius naudingumo palyginimo su kitomis sistemomis grafiką, nustatyta, kad mūsų sukurta sistema yra viena iš geresnių pagal naudingumą teikiamą vartotojui. Taip pat įvardinti penkti kriterijai, turintys didžiausią įtaką vartotojų apsisprendimui.

7. Sistemos eksperimentinio tyrimo metu sistemos vartotojų apklausa parodė, kad vartotojai yra patenkinti realizuotomis sistemos funkcijomis. Vartotojų nuomonių skirtumai buvo nedideli. Ypač teigiamai vartotojai įvertino funkcijų teikiamą naudą ir pagalbą tvarkant asmeninius finansus.

8. Atlikus sistemos vertinimą pastebėti jos trūkumai ir tobulinimo kryptys. Turbūt svarbiausias trūkumas – sistemos planavimo galimybių išplėtimas vartotojui, padaryta išvada, kad šiam trūkumui pašalinti reikėtų daugiau investuoti į sistemą.

9. Asmeninių finansų planavimo ir valdymo sistema sėkmingai atiduota užsakovui UAB „Univelas“. Šio dokumento prieduose pateikiamas „Programų sistemos įdiegimo aktas“. Veikiančią internetinės sistemos versiją galima išbandyti apsilankius adresu:

www.dev-uni.com

6. LITERATŪRA

- [1] Lietuvos Statistikos departamentas, *Informacinių technologijų naudojimas namų ūkiuose*, 2008, vasaris [žiūrėta 2008-03-09]. Prieiga per internetą: <<http://www.std.lt/lt/news/view/?id=1505>>.
- [2] Microsoft MSDN bibliotekos katalogas [interaktyvus]. *Layered Application* [žiūrėta 2008-03-20]. Prieiga per internetą: <<http://msdn.microsoft.com/en-us/library/ms978678.aspx>>.
- [3] Reade, C. *Elements of Functional Programming*. Boston, MA, USA 1989: Addison-Wesley Longman Publishing Co., Inc., 600 p. ISBN 0201129159.
- [4] Microsoft MSDN bibliotekos katalogas [interaktyvus]. *Model-View-Controller*. [žiūrėta 2008-03-18], Prieiga per internetą: <<http://msdn2.microsoft.com/en-us/library/ms978748.aspx>>.
- [5] OWASP Foundation, *A Guide to Building Secure Web Applications and Web Services*, [žiūrėta 2008 04 08]. Prieiga per internetą: <http://www.owasp.org/index.php/Guide_Table_of_Contents>.
- [6] Štuikys, V. *Programų priežiūra ir tobulinimas* [skaidrės], 2008, [žiūrėta 2008-04-10]. Prieiga per internetą: <<http://www.soften.ktu.lt/~ziber/t120m013/?FMod=T120M013>>.
- [7] Pakula A. *Namų buhalterijos programos*. Computer Bild Lietuva, 2008-07, Nr. 59, p. 24-27.
- [8] Varanauskienė, J. *Asmeninių finansų valdymo principai* [žiūrėta 2008 03 01]. Prieiga per internetą: <www.fps.lt/lt/naudinga_zinoti/finansu_planavimas/finansu_valdymas_principai>.
- [9] Elmlblad, S. *What is Financial Software and How Does It Manage Personal Finances?*, [žiūrėta 2008-04-14]. Prieiga per internetą: <<http://financialsoft.about.com/od/softwaretitle1/p/whyfinsoftware.htm>>.
- [10] Paras, S. *Building Customer Trust With Secure Ecommerce*, 2005, liepa [žiūrėta 2008 04 23]. Prieiga per internetą: <<http://www.theukwebdesigncompany.com/articles/article.php?article=1872>>
- [11] Thomas, D.; ir Heinemeier Hansson, D. *Agile Web Development with Rails*. The Pragmatic Bookshelf, 2005 m., ISBN 0-9766940-0-X, p. 1-7, p. 427-440
- [12] Fowles, D. *Top 3 Personal Finance Software* [žiūrėta 2008 03 29]. Prieiga per internetą: <<http://financialplan.about.com/od/software/tp/MoneySoftware.htm>>.
- [13] Butleris, R. *Reikalavimo specifikavimo oracle case terpėje plėtra*. [žiūrėta 2008-03-30]. Prieiga per internetą: <<http://www.leidykla.vu.lt/inetleid/inf-mok/19/str6.html>>.

- [14] ISO/IEC 9126, *International Standart. Information Technology –Software Product Evaluation – Quality characteristics and guidelines for ther use*, 1991.
- [15] Talandis, T. *Elektroninės paslaugos Lietuvoje. Kas naudinga?* [žiūrėta 2008 03 04].
Prieiga per internetą: <<http://www.smn.lt/files/e-paslaugos.pdf>>.
- [16] Danis, C.; Jones, L.; Thompson, R.; ir Levy, S. *Issues and Concerns in Internet Based Financial Applications: An Example from an Auto Finance System*, 1999 [žiūrėta 2008 04 04], p. 1-10. Prieiga per internetą:
<<http://csdl.computer.org/comp/proceedings/hicss/1999/0001/05/00015003.PDF>>.

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

7.1 Lentelė. Terminų ir santrumpų žodynas

| Santrumpa, terminas | Paaiškinimas |
|--|---|
| BLL (angl. Business Logic Layer) | – veiklos logikos projektavimo sluoksnis |
| UI (angl. <i>User Interface</i>) | – vartotojo sąsaja |
| DAL (angl. Data Access Layer) | – priėjimo prie duomenų projektavimo sluoksnis |
| MVC (angl. Model View Controller) | – modelis-vaizdas-valdiklis šablonas |
| RoR (angl. Ruby on Rails) | – programavimo kalbos karkasas |
| AJAX (angl. Asynchronous JavaScript“ and XML) | – interaktyvių interneto puslapių-programų kūrimo metodas, technologija |
| SSL (angl. Secure Sockets Layer) | – šifravimu (kriptografija) paremtas protokolas, kuris užtikrina saugų bendravimą internetu |
| DES (angl. Data Encryption Standard) | – metodas duomenims užšifruoti |
| OS (angl. Operating System) | – operacinė sistema |
| DB (angl. <i>Database</i>) | – duomenų bazė |
| ISO (angl. International Organization for Standardization) | – tarptautinė standartų organizacija |
| CMM (angl. Capability Maturity Model) | – galimybių brandos modelis |
| UML (angl. Unified Modelling Language) | – unifikuota modeliavimo kalba UML |
| WEB 2 | – taip įvardijamos dabartinės interneto tendencijos (pvz.: wiki, blogger, youtube) |

8. PRIEDAI

1 priedas. Apklauso rezultatai

| Programa ir jos vertinimas | Svarba | 1 | 2 | 3 | 4 | 5 | Vidurkis |
|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Diegimas ir pagalba | 22% | 9,59 | 9,55 | 9,55 | 9,14 | 9,50 | 9,46 |
| Lietuviška versija | 10% | 10 | 10 | 10 | 9 | 10 | 9,8 |
| Spausdintas vartotojo vadovas | 4% | 10 | 10 | 10 | 10 | 10 | 10 |
| Įdiegimas | 3% | 10 | 10 | 10 | 10 | 10 | 10 |
| Užimama vieta kietajame diske | 1% | 10 | 10 | 10 | 10 | 10 | 10 |
| Pagalba programoje | 2% | 8 | 8 | 7 | 8 | 7 | 7,6 |
| Pagalba internete | 1% | 5 | 4 | 6 | 5 | 5 | 5 |
| Išdiegimas | 1% | 10 | 10 | 10 | 10 | 10 | 10 |
| Programos valdymas ir galimybės | 23% | 8,17 | 7,52 | 7,87 | 8,17 | 8,00 | 7,95 |
| Valdymo patogumas | 10% | 9 | 8 | 8 | 9 | 8 | 8,4 |
| Mygtukų bei pranešimų aiškumas | 5% | 9 | 8 | 9 | 9 | 9 | 8,8 |
| Galimybė vesti apskaitą įvairiomis valiutomis | 3% | 1 | 1 | 2 | 1 | 3 | 1,6 |
| Duomenų apsauga slaptažodžiu | 3% | 10 | 10 | 10 | 10 | 10 | 10 |
| Galimybė naudotis programa keliems vartotojams | 2% | 10 | 10 | 10 | 10 | 10 | 10 |
| Banko sąskaitos apskaita | 18% | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| Galimybė gauti duomenis iš mūsų el. bankininkystės sistemos | 10% | 1 | 1 | 1 | 1 | 1 | 1 |
| Kelių skirtingų sąskaitų apskaitą galima vesti? | 4% | 1 | 1 | 1 | 1 | 1 | 1 |
| Sąskaitos įplaukų bei išlaidų rodymas | 4% | 1 | 1 | 1 | 1 | 1 | 1 |
| Namų biudžeto planavimo funkcijos | 22% | 7,55 | 7,00 | 7,14 | 7,36 | 7,55 | 7,32 |
| Ataskaitos apie mūsų įplaukas / išlaidas bei finansinę būklę | 3% | 8 | 8 | 7 | 9 | 8 | 8 |
| Išlaidų kategorijų skaičius | 3% | 10 | 9 | 8 | 10 | 9 | 9,2 |
| Priminimai apie įsiskolinimus bei reguliarius ir atliktus mokėjimus | 3% | 3 | 2 | 4 | 5 | 4 | 3,6 |
| Skolininkų ir skolintojų sąrašas | 3% | 10 | 8 | 9 | 8 | 10 | 9 |
| Perspėjimas apie viršytą nustatytą išlaidų limitą | 3% | 1 | 1 | 1 | 1 | 1 | 1 |
| Atsiskaitymų už paslaugas (elektrą, dujas, vandenį) skaičiuoklė | 7% | 10 | 10 | 10 | 9 | 10 | 9,8 |
| Papildomos funkcijos | 15% | 4,07 | 4,67 | 4,80 | 3,93 | 4,60 | 4,41 |
| Vertybinių popierių bei kitų investicijų apskaita | 5% | 1 | 1 | 1 | 1 | 1 | 1 |
| Galimybė iškelti duomenis į kitą formatą | 4% | 5 | 8 | 7 | 6 | 7 | 6,6 |
| Galimybė kuri savo pačių ataskaitas | 3% | 7 | 6 | 7 | 5 | 7 | 6,4 |
| Paskolų ir kitos skaičiuoklės | 3% | 5 | 5 | 6 | 5 | 5 | 5,2 |
| Galutinis vertinimas balais | 100% | 6,44 | 6,25 | 6,38 | 6,28 | 6,46 | 6,36 |

2 priedas. Sistemos įdiegimo dokumentas

SISTEMOS ĮDIEGIMO AKTAS

Šiuo aktu Paslaugos gavėjas priima, o Paulius Janušaitis perduoda Sistemos gavėjo naudojimui sukurtą Asmeninių finansų planavimo ir valdymo sistemą, ir atlieka šios paslaugos įdiegimo darbus pagal 2007-10-01 sutartį Nr. 1407-14-13.

1.SISTEMOS GAVĖJAS

| | | | |
|--|----------------------|---|--|
| fizinio asmens vardas, pavardė/ juridinio asmens pavadinimas | Giedrius Kurlavičius | fizinio asmens nuolatinė gyvenamoji vieta / juridinio asmens buveinės adresas | Žemgulio 8/Savanorių 192, Kaunas LT-44151, Lithuania |
| Asmuo ryšiams | Giedrius Kurlavičius | Telefono numeris | +370 686 26634 |
| Faksas | +370 37 761703 | El. Paštas | g.kurlavicius@univelas.com |

2. SISTEMOS ĮDIEGIMO ADRESAS

| |
|---|
| http://www.dev-uni.com |
|---|

3. PERDUOTA IR ĮDIEGTA SISTEMA

| Nr. | Sistemos pavadinimas | Tipas |
|-----|--|--------------------------------|
| 1 | ASMENINIŲ FINANSŲ PLANAVIMO IR VALDYMO SISTEMA | Internetinis puslapis/portalas |
| | | |
| | | |
| | | |

4. DARBUS ATLIKO IR ĮDIEGTĄ SISTEMĄ PERDAVĖ

| | |
|-----------------------------------|---------------|
| Vardas, pavardė, pareigos | Parašas, data |
| PAULIUS JANUŠAITIS, KTU STUDENTAS | |
| | |

5. SISTEMOS GAVĖJAS:

Pasirašęs asmuo patvirtina, kad jam perduota sistema, nurodyta šio akto 3 punkte, veikia tvarkingai, ir dėl šios sistemos įdiegimo bei veikimo jokių pretenzijų jis neturi.

| | |
|--|---------------|
| Vardas, pavardė, pareigos | Parašas, data |
| GIEDRIUS KURLAVIČIUS, PROJEKTŲ VADOVAS | |
| | |