|  |
| --- |
| VILNIAUS KOLEGIJA  ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS    **LEISTA GINTI**  **Elektronikos ir informatikos fakulteto**  **prodekanė**  **\_\_\_\_\_\_\_\_\_\_\_ dr. Loreta Savulionienė**  **2017-01-20** |
| **UAB „COMMERCIAL ENGINEERING TECHNOLOGIES“ KLIENTŲ INFORMACIJOS VALDYMO SISTEMOS KŪRIMAS**   |  | | --- | | BD 653E10002 IS13A |   **BAIGIAMASIS DARBAS** | |

STUDENTAS MANTAS BRAZDAUSKIS

2017-\_\_-\_\_

VADOVĖ IRINA LIUŠICYNA

2017-\_\_-\_\_

RECENZENTAS (-Ė) VARDAS PAVARDĖ

2017-\_\_-\_

2017

|  |
| --- |
| VILNIAUS KOLEGIJA / UNIVERSITY OF APPLIED SCIENCES  FACULTY OF ELECTRONICS AND INFORMATICS    **AUTHORIZED BY**  **Vice Dean of Electronics and Informatics**  **Faculty**  **\_\_\_\_\_\_\_\_\_\_\_ dr. Loreta Savulionienė**  **2017-01-20** |
| **UAB “COMMERCIAL ENGINEERING TECHNOLOGIES” CLIENT INFORMATION MANAGEMENT SYSTEM DESIGN**   |  | | --- | | 653E10002 IS13A |   **FINAL PROJECT** | |

UNDERGRADUATE MANTAS BRAZDAUSKIS

2017-\_\_-\_\_

SUPERVISOR IRINA LIUŠICYNA 2017-\_\_-\_\_

REVIEWER VARDAS PAVARDĖ

2017-\_\_-\_\_

2017

Vilniaus kolegija

Elektronikos ir informatikos fakultetas Informacinių sistemų katedra

Vilnius Kolegija / University of Applied Sciences Faculty of Electronics and Informatics Department of Information Systems

Studijų programa: Informacijos sistemos

Studijų programos valstybinis kodas: 653E10002

Data: 2015-01-17

Baigiamojo darbo tema Uab „Commercial Engineering Technologies“ Klientų Informacijos Valdymo Sistemos Kūrimas  
Diplomantas **Mantas Brazdauskis**Vadovė **Irina Liušicyna**

The Theme of Final Project Uab “Commercial Engineering Technologies” Client Information Management System Design

Undergraduate **Mantas Brazdauskis**

Adviser **Irina Liušicyna**

Lietuvių kalba:

Darbo tikslas ir uždaviniai:

Baigiamojo darbo tikslas – sukurti naują informacijos sistemą skirtą ryšių su klientais valdymui.

Darbo tikslui pasiekti suformuluoti šie uždaviniai:

* Išnagrinėti įmonės veiklą, siekiant nustatyti įmonėje esančių procesų problemas;
* Atlikti informacijos šaltinių analizę;
* Nustatyti funkcinius ir nefunkcinius sistemos reikalavimus;
* Parengti kūriamos sistemos realizacijos projektą;
* Parengti baigiamojo darbo ekonominį pagrindimą;

Foreign language (English)

Objectives:

The main aim is to create new information system for relations with clients managment. To reach this aim it is necessary to:

* Examine the company's activities in order to establish the company in the process problems;
* Perform an analysis of literature;
* Identify functional and non-functional system requirements;
* To prepare the framework of realization of the project;
* Develop a thesis economic justification;

Taikyti metodai:

* Objektinis projektavimas naudojant UML projektavimo kalbą;
* Atliekamų procesų modeliavimas BPMN
* Programavimas PHP programavimo kalba pasitelkiant CodeIgniter PHP karkasą;
* Klasikinis gyvavimo ciklo modelis;

Darbo dalyviai ir jo apimtis:

* Diplmantas;
* Baigiamojo darbo vadovas;
* Recenzentas;
* Anglų kalbos konsultantė;
* Techninė konsultantė;
* Kvalifikacijos komisija;
* Aiškinamasis raštas(... psl);

Darbo rezultatai:

* Išnagrinėta įmonės veikla, siekiant nustatyti įmonėje esančių procesų problemas;
* Atlikta informacijos šaltinių analizę;
* Nustatyti funkciniai ir nefunkciniai sistemos reikalavimai;
* Parengtas kūriamos sistemos realizacijos projektas;
* Parengtas baigiamojo darbo ekonominį pagrindimas;
* Atlikta sistemos realizaciją;

Išvados:

Parengtas UAB „Commercial engeneering Technologies“ klientų informacijos valdymo informacijos sistemos projektas. Taip pat projektas yra įgyvendintas naudojant PHP programavimo kalbą. Baigiamajame darbe

Applied methods:

* Object-oriented design using UML design language;
* Using processes modeling with BPMN
* Programming in PHP using CodeIgniter PHP framework;
* Classic life cycle model;

Participants and volume of the explanatory text:

* Undergraduate;
* Adviser;
* Reviewer;
* English Consultant;
* Technical Consultant;
* Qualification Commision;
* Explanatory Note (... psl);

Results:

* analyzed the activities of the company in order to establish the company process problems;
* Analysied of information sources;
* Identified functional and non-functional system requirements;
* Prepared the system implementation project;
* Prepared thesis economic justification;
* Created system realization;

Conclusions:

Prepared UAB „Commercial Engeneering Technologies“ customer information management system project. Also, the project is implemented using the PHP programming language. The final work in the economic

esančiame ekonominiame pagrindime teigiama jog projektas atsipirks per 16 mėn.

justification states that the project will pay for itself within 16 months.

**TURINYS**

[1. Įvadas 5](#_Toc471270746)

[1.1. Kilmė 5](#_Toc471270747)

[1.2. Baigiamojo darbo tikslas 5](#_Toc471270748)

[1.3. Prielaidos 5](#_Toc471270749)

[1.4. Išorinės priklausomybės 5](#_Toc471270750)

[1.5. Baigiamojo darbo Gantt‘o diagrama 5](#_Toc471270751)

[1.6. Pagrindimas 8](#_Toc471270752)

[1.7. Išteklių poreikis 8](#_Toc471270753)

[1.8. Pasirinktų metodų ir priemonių pagrindimas 9](#_Toc471270754)

[2. PROFESINĖS VEIKLOS KOMPETENCIJŲ SĄRAŠAS 11](#_Toc471270755)

[3. Užduoties formulavimas 13](#_Toc471270756)

[3.1. Baigiamojo darbo objektas 13](#_Toc471270757)

[3.2. Baigiamojo darbo tikslai 13](#_Toc471270758)

[3.3. Užduoties formulavimas 13](#_Toc471270759)

[4. Informacijos šaltinių analizė 14](#_Toc471270760)

[5. Užduoties analizė 16](#_Toc471270761)

[5.1. Įmonės kaip orginazacijos nagrinėjimas 16](#_Toc471270762)

[5.2. Probleminė sritis 17](#_Toc471270763)

[5.2.1. Brangus palaikymo mokestis 17](#_Toc471270764)

[5.2.2. Ataskaitų generavimas AdWords 18](#_Toc471270765)

[5.2.3. Sąskaitų generavimas 18](#_Toc471270766)

[5.2.4. Sutarčių generavimas 18](#_Toc471270767)

[5.3. CodeIgniter PHP karkasas kas tai? 19](#_Toc471270768)

[5.4. MVC programavimo principas 19](#_Toc471270769)

[5.5. Alternatyvos 20](#_Toc471270770)

[5.5.1. Perkant jau sukurtą ryšių su klientais informacijos sistemą 21](#_Toc471270777)

[5.5.2. Integruoti AdWords į dabar naudojamas sistemas 21](#_Toc471270778)

[6. Projektinė dalis 22](#_Toc471270779)

[6.1. Funkcinių ir nefunkcinių reikalavimų nustatymas 22](#_Toc471270780)

[6.2. Ryšių su klientais informacijos sistemos use case diagrama 23](#_Toc471270781)

[6.3. Sistemos veiklos diagramos 23](#_Toc471270782)

[6.5. Sistemos klasių diagramos aprašymas 36](#_Toc471270783)

[6.5.1. Sistemos klasių diagrama 36](#_Toc471270784)

[6.6. Sistemos dinaminis modelis 40](#_Toc471270785)

[6.6.1. Sistemos būsenų diagramos 47](#_Toc471270786)

[6.7. Duomenų bazės architektūra 49](#_Toc471270787)

[6.7.1. Duomenų bazės loginis modelis 49](#_Toc471270788)

[6.7.2. Loginio modelio esybės 49](#_Toc471270789)

[6.7.3. Ryšiai tarp esybių 52](#_Toc471270790)

[6.8. Sistemos prototipas 53](#_Toc471270791)

[7. EKONOMINIS PAGRINDIMAS 54](#_Toc471270792)

[7.1. Baigiamojo darbo administravimo išlaidos 54](#_Toc471270793)

[7.2. Sistemos sukūrimo kaštai 54](#_Toc471270794)

[7.3. Sistemos palaikymo kaštai per menėsį 54](#_Toc471270795)

[7.4. Sistemos nauda įmonei 55](#_Toc471270796)

[7.5. Sistemos atsiperkamumas 55](#_Toc471270797)

[8. Išvados ir pasiūlymai 57](#_Toc471270798)

[9. informacijos šaltinių sąrašas 57](#_Toc471270799)

# Įvadas

## Kilmė

Viena iš pagrindinių baigiamojo darbo temos pasirinkimo priežasčių yra siekis iškelti naujus iššukius sau. Ryšių su klientais informacijos sistemos kūrimas yra vienas iš tokių iššukių. Tai yra pirmas mano kaip programuotojo darbas susietas, kuriant sistemą nuo pamatų.

## Baigiamojo darbo tikslas

Baigiamojo tikslas yra sukurti UAB „Commercial Engeneering Technologies“ ryšių su klientais sistemą, turinčia integraciją su AdWords reklamos servizu. Šis tikslas iškeltas, nes įmonei reikalinga atnaujinta ryšių su klientais sistema, kuri nebūtų kelios sistemos.

## Prielaidos

Pagrindinis mano tikslas, kuriant ryšių su klientais informacijos sistemą yra tai jog noriu patobulinti savo programavimo žinias, šioje srityje dirbu jau du metus, todėl manau, kad tikrai sugebėsiu sukurti šią informacijos sistemą.

Sukurti šią sistemą man yra iššukis, kadangi niekad anksčiau neporgramavau MVC principu parašytu karkasu, tik keičiau kodą. Kiekvienas iššukis mane stumia tolyn kurti ir įrodyti jog galiu tai padaryti be didelės kitų žmonių pagalbos.

## Išorinės priklausomybės

Pagrindinė Baigiamojo darbo išorinė priklausomybė yra įmonėje dirbantys vadybininkai, nes nuo jų priklauso tiek sistemos dizainas, tiek sistemos naudotojo sąsaja.

## Baigiamojo darbo Gantt‘o diagrama

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Outline Number | Task Name | Duration | Start | Finish | Predecessors | Resource Names |
| **1** | **Baigiamasis darbas** | **47 days** | **Mon 11/7/16** | **Tue 1/10/17** |  |  |
| **1.1** | **Projektinės dalies ruošimas** | **7 days** | **Mon 11/7/16** | **Tue 11/15/16** |  | **Diplomantas** |
| 1.2 | BD vadovo konsultacija | 2 hrs | Tue 11/15/16 | Wed 11/16/16 | 2 | Diplomantas;Baigiamojo darbo vadovas |
| **1.3** | **Klientų modulio programavimas** | **3 days** | **Mon 11/14/16** | **Wed 11/16/16** | **9** | **Diplomantas** |
| 1.3.1 | Duomenu bazės derinimas | 1 hr | Mon 11/14/16 | Mon 11/14/16 |  | Diplomantas |
| 1.3.2 | Klientų funkcijų programavimas | 2 days | Mon 11/14/16 | Tue 11/15/16 |  | Diplomantas |
| 1.3.3 | Dizaino tvarkymas pagal nurodymus | 8 hrs | Wed 11/16/16 | Wed 11/16/16 |  | Diplomantas |
| 1.3.4 | BD vadovo konsultacija | 2 hrs | Wed 11/16/16 | Wed 11/16/16 |  | Diplomantas;Baigiamojo darbo vadovas |
| **1.4** | **Užduočių modulio programavimas** | **2 days** | **Thu 11/17/16** | **Fri 11/18/16** | **10** | **Diplomantas** |
| 1.4.1 | Duomenu bazės derinimas | 1 hr | Thu 11/17/16 | Thu 11/17/16 |  | Diplomantas |
| 1.4.2 | Užduočių modulio funkcijų programavimas | 2 days | Thu 11/17/16 | Fri 11/18/16 |  | Diplomantas |
| 1.4.3 | Dizaino tvarkymas pagal nurodymus | 8 hrs | Fri 11/18/16 | Fri 11/18/16 |  | Diplomantas |
| 1.4.4 | BD vadovo konsultacija | 2 hrs | Fri 11/18/16 | Fri 11/18/16 |  | Diplomantas;Baigiamojo darbo vadovas |
| 1.5 | Pirmoji baigiamojo darbo peržiūra | 0.5hr | Mon 11/21/16 | Mon 11/21/16 | 15 |  |
| 1.6 | BD vadovo konsultacija | 2 hrs | Mon 11/21/16 | Mon 11/21/16 | 20 | Diplomantas |
| **1.7** | **AdWords modulio programavimas** | **5 days** | **Mon 11/21/16** | **Fri 11/25/16** | **21** | **Diplomantas** |
| 1.7.1 | Duomenų bazės paruošimas | 1 hr | Mon 11/21/16 | Mon 11/21/16 |  | Diplomantas |
| 1.7.2 | Adwords API paruošimas | 1 day | Mon 11/21/16 | Mon 11/21/16 |  | Diplomantas |
| 1.7.3 | Adwords modulio funkcijų programavimas | 3 days | Tue 11/22/16 | Thu 11/24/16 |  | Diplomantas |
| 1.7.4 | Dizaino tvarkymas pagal nurodymus | 1 day | Fri 11/25/16 | Fri 11/25/16 |  | Diplomantas |
| 1.7.5 | BD vadovo konsultacija | 2 hrs | Fri 11/25/16 | Fri 11/25/16 |  | Diplomantas;Baigiamojo darbo vadovas |
| **1.8** | **Sąskaitų modulio programavimas** | **5 days** | **Mon 11/28/16** | **Fri 12/2/16** | **22** | **Diplomantas** |
| 1.8.1 | Duomenų bazės paruošimas | 1 hr | Mon 11/28/16 | Mon 11/28/16 |  | Diplomantas |
| 1.8.2 | Sąskaitų modulio funkcijų tvarkymas | 3 days | Mon 11/28/16 | Wed 11/30/16 |  | Diplomantas |
| 1.8.3 | Dizaino tvarkymas pagal nurodymus | 2 days | Thu 12/1/16 | Fri 12/2/16 |  | Diplomantas |
| 1.8.4 | BD vadovo konsultacija | 2 hrs | Fri 12/2/16 | Fri 12/2/16 |  | Diplomantas;Baigiamojo darbo vadovas |
| **1.9** | **Testavimas** | **4 days** | **Mon 11/28/16** | **Thu 12/1/16** | **28** | **Diplomantas** |
| 1.9.1 | Klientų modulio testavimas | 1 day | Mon 11/28/16 | Mon 11/28/16 |  | Diplomantas |
| 1.9.2 | Užduočių modulio testavimas | 1 day | Tue 11/29/16 | Tue 11/29/16 |  | Diplomantas |
| 1.9.3 | AdWords API modulio testavimas | 1 day | Wed 11/30/16 | Wed 11/30/16 |  | Diplomantas |
| 1.9.4 | Sąskaitų modulio programavimas | 1 day | Thu 12/1/16 | Thu 12/1/16 |  | Diplomantas |
| 1.9.5 | BD vadovo konsultacija | 2 hrs | Thu 12/1/16 | Thu 12/1/16 |  | Diplomantas;Baigiamojo darbo vadovas |
| **1.10** | **Sistemos testavimas su vadovu** | **4 days** | **Mon 12/5/16** | **Thu 12/8/16** | **33** | **Diplomantas** |
| 1.10.1 | Klientų modulio testavimas | 1 day | Mon 12/5/16 | Mon 12/5/16 |  | Diplomantas |
| 1.10.2 | Užduočių modulio testavimas | 1 day | Tue 12/6/16 | Tue 12/6/16 |  | Diplomantas |
| 1.10.3 | AdWords API modulio testavimas | 1 day | Wed 12/7/16 | Wed 12/7/16 |  | Diplomantas |
| 1.10.4 | Sąskaitų modulio programavimas | 1 day | Thu 12/8/16 | Thu 12/8/16 |  | Diplomantas |
| 1.10.5 | BD vadovo konsultacija | 2 hrs | Mon 12/5/16 | Mon 12/5/16 |  | Diplomantas;Baigiamojo darbo vadovas |
| 1.11 | Sistemo klaidų taisymas po testavimo | 2 days | Fri 12/9/16 | Sun 12/11/16 | 39 | Diplomantas |
| 1.12 | BD vadovo konsultacija | 2 hrs | Mon 12/12/16 | Mon 12/12/16 | 45 | Diplomantas;Baigiamojo darbo vadovas |
| 1.13 | Antroji baigiamojo darbo peržiūra | 0.5hr | Mon 12/12/16 | Mon 12/12/16 | 46 |  |
| 1.14 | Klaidų taisymas po antrosios peržiūros | 20 days | Mon 12/12/16 | Fri 1/6/17 | 47 | Diplomantas |
| 1.15 | Trečioji baigiamojo darbo peržiūra | 0.5hr | Fri 1/6/17 | Fri 1/6/17 | 48 | Diplomantas |
| 1.16 | Baigiamojo darbo taisymas po trečiosios peržiūros | 2 days | Fri 1/6/17 | Mon 1/9/17 | 49 | Diplomantas |
| 1.17 | BD vadovo konsultacija | 2 hrs | Mon 1/9/17 | Tue 1/10/17 | 50 | Diplomantas;Baigiamojo darbo vadovas |
| 1.18 | Gynimas Informacinių sistemų katedros posėdyje | 1 day | Tue 1/10/17 | Tue 1/10/17 | 51 | Diplomantas |

**Užduočių vykdymo kelias pateikiamas 1 Priede**

## Pagrindimas

Kaip baigiamojo darbo rezultatas bus pateikiami šie objektai:

* UAB „Commercial Engeneering Technologies“ ryšių su klientais informacijos sistemos projektas
* UAB „Commercial Engeneering Technologies“ ryšių su klientais informacijos sistemos projekto realizacija

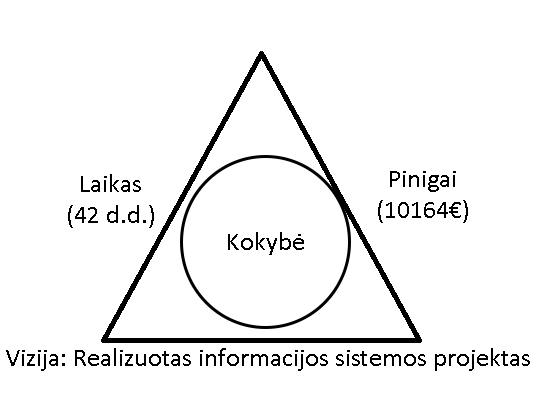
## Išteklių poreikis

Šiame skyriuje bus pateikiami visi reikalingi baigiamojo darbo projekto ištekliai.

Žmogiškieji ištekliai: Diplomantas, baigiamojo darbo vadovas, recenzentas, techninis konsultantė, recenzentas.

Materialieji: „Microsoft Windows 10“, interneto prieiga, „Microsoft Office 2013“, „Microsoft Outlook“, „Visual Paradigm Community Edition“

Taip pat pateikiamas baigiamojo darbo sėkmės trikampis – tai faktoriai, įtakojantys baigiamojo darbo eigą ir rezultatus (1 pav.).

  
pav. 1. Sėkmės trikampis

## Pasirinktų metodų ir priemonių pagrindimas

Pasirinkti baigiamojo darbo metodai:

1. Procesų modeliavimas BPMN – procesų modeliavimas BPMN pasirinktas tam, jog būtų aiškiai parodoma esamų sistemų probleminė sritis, taip pat papildomas darbas darbuotojams;
2. Projektavimas UML kalba - projektavimas objektiškai UML kalba pasirinktas tam jog būtų sukurti sistemos aprašai, taip pat būtų aiškus sistemos vaizdas, kaip ji turi veikti.

Pasirinktos darbo priemonės:

1. PHP programavimo kalba – galimos alternatyvos šiai programavimo kalbai: C++ arba JAVA. PHP programavimo kalba pasirinkta dėl to jog įmonėje visi kuriami projektai yra kūriami šia kalba. Programuojant PHP programavimo kalba siekiama sukurti informacijos sistemą.
2. MySQL duomenų bazė – galimos alternatyvos MsSQL duomenų bazė. MySQL duomenų bazė pasirinkta numatant jog sistema veiks Linux serverio pagrindu, ko pasekoje MsSQL duomenų bazė tiesiog neveiks. MySQL duomenų bazė saugos naudotojų, sąskaitų, ataskaitų, sutarčių, klientų informaciją.
3. „Microsoft Windows 10“ - operacinė sistema pasirinkta ruošti baigiamajam darbui, nes kolegija išduoda šią programinę įrangą.
4. „Microsoft Office 2013“ – paketas naudojamas ruošti baigiamajam darbui, taip pat išduodamas kolegijos.
5. „Microsoft Outlook“ – pasirinkta priemonė siųsti laiškams bendradarbiaujant su baigiamojo darbo vadove, technine konsultante taip pat anglų kalbos konsultante.
6. „Visual Paradigm Community Edition“ – pasirinkta priemonė projektuoti UML diagramas, taip pat ši priemonė pasirinkta, dėl to jog buvo naudojama mokantis projektavimo kalbos kolegijoje.

# PROFESINĖS VEIKLOS KOMPETENCIJŲ SĄRAŠAS

Šiame skyriuje bus pateikiamos profesinės veiklos kompetencijos, kurios bus įrodytos baigiamuoju darbu.

lentelė 1. Profesinės veiklos kompetencijų sąrašas

|  |  |  |
| --- | --- | --- |
| **Kompetencija** | **Studijų programos rezultatai** | **Įrodymas** |
| 1.Instrumentinė kompetencija: gebėjimas surasti, atrinkti, analizuoti informaciją. | 1.1.Gebėti taikyti šiuolaikines informacijos paieškos ir atrankos pagal suformuluotus požymius sistemas. | Atlikta paieška iš kolegijos prenumeruojamų duomenų bazių papildomiems straipsniams ir knygoms rasti (56 psl.) |
| 1.2. Atlikti skirtingų informacijos šaltinių analizę konkrečiu aspektu | Atlikta informacijos šaltinių analizė(13 psl.) |
| 2.Sisteminė kompetencija: gebėjimas mokytis savarankiškai | 2.1.Gebėti savarankiškai tobulinti savo kvalifikaciją, apibrėžiant tobulintinas kompetencijas ir panaudojant šiuolaikines informacijos technologijas. | Baigiamojo darbo realizacijai naudota PHP programavimo kalba ir CodeIgniter Karkasas. Viskas išmokta savarankiškai. |
| 3.Sisteminė kompetencija: gebėjimas taikyti įgytas žinias praktinėje veikloje | 3.1. Gebėti nustatyti, kokių sričių žinios yra būtinos vykdant konkrečią praktinę užduotį | Baigiamajame darbe nurodyta užduoties analizė (17-21 psl.) |
| 3.2.Gebėti pasirinkti reikalingas žinių taikymo priemones ir metodus. | Aprašytas pasirinktų priemonių pagrindimas. (9-10 psl.) |
| 5.Kurti ir eksploatuoti organizacijų IS | 5.1. Analizuoti organizacijos dalykinės veiklos sritį ir nustatyti IS kūrimo tikslus | Išanalizuota įmonės ir jos naudojimos IS (16 – 17 psl.) |
| 5.2. Parengti reikalavimų organizacijos IS specifikaciją | Parengti funkciniai ir nefunkciniai reikalavimai (22 psl.) |
| 5.3. Parengti organizacijos IS projektą | Paruoštas IS projektas (22 – 52 psl) |
| 5.4. Valdyti organizacijos IS diegimą ir eksploatavimą | Paruošta IS diegimo instrukcija (3 priedas) |
| 5.5. Konsultuoti, mokyti organizacijos personalą IS naudojimo klausimais | Paruošta IS naudojimosi instrukcija (4 priedas) |
| 5.6. Projektuoti ir realizuoti atitinkamoje programinėje aplinkoje programų struktūras, atitinkančias IS reikalavimus | Realizuotas projektas PHP programavimo kalba http://crm.brazdauskis.eu |
| 7. Modeliuoti, administruoti organizacijų informacinių sistemų duomenų bazes | 7.1.Išskirti loginio sprendimo modelio objektus, jų atributus, tarpusavio ryšius, įvertinti ir parinkti modelio organizavimo programines priemones. | Sukurtas informacijos sistemos duomenų bazės loginis modelis (48 – 52 psl.) |

# Užduoties formulavimas

## Baigiamojo darbo objektas

Baigiamojo darbo objektas – įmonės ryšiai su klientais ir komunikacija su jais.

## Baigiamojo darbo tikslai

Baigiamojo darbo tikslas – sukurti naują informacijos sistemą skirtą ryšių su klientais valdymui.

Darbo tikslui pasiekti suformuluoti šie uždaviniai:

* Išnagrinėti įmonės veiklą, siekiant nustatyti įmonėje esančių procesų problemas;
* Atlikti informacijos šaltinių analizę;
* Nustatyti funkcinius ir nefunkcinius sistemos reikalavimus;
* Parengti kūriamos sistemos realizacijos projektą;
* Parengti baigiamojo darbo ekonominį pagrindimą;
* Atlikti sistemos realizaciją;

## Užduoties formulavimas

Šiame skyriuje pateikiamas baigiamojo darbo tikslų detalizaivimas.

Atlikus įmonės veiklos, nustatoma probleminė sritis ir taip nurodoma kokie procesai bus pagerinti, kuriant naują informacijos sistemą. Naudojant BPMN modelį nustatomos procesų problemos.

Atlikus literatūros analizę, galima nustatyti kokiomis priemonėmis bus atliekamas visas projektas.

Funkciniai ir nefunkciniai reikalavimai nustato ką ir kaip sistema veiks

Parengiamas sistemos realizacijos projektas: UML kalba suprojektuojamas statinis ir dinaminis sistemos modeliai.

Parengiamas loginis duomenų bazės modelis, kur nurodoma, kokie duomenys bus sistemoje.

Sistemos realizacija – sukuriama nauja sistema, sistemos realizacija atliekama PHP programavimo kalba ir MySQL duomenų baze.

# Informacijos šaltinių analizė

1. Professional PHP – Šią knygą galima suskirstyti į kelis skirsnius. Susipažinimą su PHP programavimo kalba, PHP programavimo kalbos objektiškai orietuota puse, UML projektavimo kalbos pagrindu, tokius kaip klasių ir sekų diagramos, Model-View-Controller programavimo principą, Taip pat apie UML projektavimo kalbos pradmenis.

Ši knyga padėjo įsigilinti į PHP programavimo kalbos ir MVC principo faktus ir informacija

Knyga buvo naudojama programuojan informacijos sistemą, taip pat aprašant PHP programavimo kalbą ir MVC programavimo principus.

1. CodeIgniter dokumentacija – Šioje dokumentacijoje yra detaliai aprašoma PHP karkaso naudojimosi instrukcija, ją sudaro: Įvadas, Instaliacija, Padėjėjų biblioteka, Bibliotekos, Klasių informacija, Duomenų bazės pagalba.

Codeigniter dokumentacija naudota projekto realizacijai atlikti .

1. Codeigniter 1.7 Professional Development, Griffiths, Adam – Šioje knygoje yra aprašomi CodeIgniter programavimo principai, kurie leidžia suprasti kaip veikia karkasas. Knygoje yra nemažai pavyzdžių kaip reikėtų saugiai ir taisyklingai aprašyti karkaso modulius, taip juos padarant saugiais ir nepriieinamus norintiems juos nulaužti. Ši knyga yra tinkama tiek pradedančiajam programuoti karkasais tiek pažengusiam.  
   Šioje knygoje pateikiama informacija apie Web Servizus ir apie tai kaip reikia paimti duomenis iš jų. Šios knygos informacija buvo naudojama realizuojant sistemą ir perteikiant išorinių duomenų paėmimą iš tokių kaip Google servizų.
2. Just Enough Web Programming with XHTML, PHP, and MYSQL – Šis straipsnis pateikia informaciją kaip sukurti internetinį puslapį naudojant html, php ir mysql. Taip pat jame aprašomi kaip reikėtų taisyklingai parodyti svetainės dizainą css pagalba. Taip pat straipsnyje užsiminima apie Turinio Valdymo Sistemas.   
   Straipsnio informacija panaudota realizuojant sistemą
3. Role of UML Class Diagram in Object-Oriented Software Development – Straipsnis aprašantis pagrindinius apsektus, projektuojant informacinę sistemą iš programuotojo pusės. Straipsnyje yra pateikiama informacija kaip reikėtų suprojektuoti klasių diagramas taip jog programuotojas pamatęs šią diagramą galėtų sukurti informacijos sistemą tiksliai pagal funkcinius reikalavimus.   
   Straipsnio informacija buvo naudojama projektuojant klasių diagramą UML projektavimo kalba.
4. PHPWord dokumentacija – Dokumentacijoje yra pateikiami metodai kaip naudoti PHPWord biblioteką.  
   Pagrindiniai skyriai naudojami bibliotekoje: Sekcijos, Teksto elementai, Nuotraukos, Lentelės, Šablonai

Pagrindinis dokumentacijos skyrius, kuris buvo naudotas baigiamojo darbo realizacijai: Šablonai

Šablonų skyriuje yra aprašoma kaip sugeneruoti naują Word dokumentą jau iš turimo šablono, kaip priskirti kintamuosius, ir juos išsaugoti naujus Word dokumentus.

1. AdWords API dokumentacija – Šioje dokumentacijoje yra aprašoma, kaip reikia integruoti AdWords sistemą į jau naudojamą PHP sistemą. Dokumentacijoje yra pateikiama biblioteka, kurią įdiegus į sistemą reikalingas autorizacijos „žetonas“, kuris leidžia prisijungti prie API ir tada pasiimti visus duomenis, kurie reikalingi ataskaitų generavimui sistemoje.

Šis šaltinis buvo naudojamas tam jog būtų realizuotas pagrindinis informacijos sistemos sprendimas. Tai yra AdWords integracija į sistemą tam jog sutrumpinti darbuotojų darbo laiką.

# Užduoties analizė

## Įmonės kaip orginazacijos nagrinėjimas

Pati organizacija susidaro iš dviejų nedidelių skyrių, tai Vadybininkai ir gamybos skyrius. Organizacija orientuojasi į atliekamų paslaugų kokybę, o ne kiekybę. Šiuo metu organizacijoje dirba 15 darbuotojų. Organizacija už savo kokybę yra užsitarnavę Google Premier Partner vardą, kas suteikia didesnes galimybes naudojantiems AdWords reklamą.  
Šiuo metu įmonei reikalinga nauja sistema pakeisti senas 2 naudojamas sistemas, kurios reikalauja didelio finansavimo. Taip pat reikalinga sistema, kuri valdytų sąskaitas ir sutartis vienoje vietoje. Taigi todėl yra kuriamas nauja ryšių su klientais sistema, kurioje bus įtraukiama AdWords aplikacijų programavimo sąsaja, kuri leidžia pasiimti visus kampanijos skaičius ir juos paversti grafikais, atlikti ataskaitas.

  
pav. 2. UAB "Commercial Engeneering Technologies" Organigrama.

Taigi abi sistemos yra perkamos iš kitų tiekėjų(kas mėnesį mokant tam tikrą mokestį, priklausant nuo pasirinktų modulių kiekio ir naudotojų kiekio)

Taigi pirmiausia buvo išnagrinėta Lexitainformacijos Sistema

Šioje sistemoje vadyba būtent ir dirba su klientais, tai yra Informacijos Sistema‘as, kuriame yra nurodomi visi esamų klientų kontaktai ir taip pat susisiekimas su kiekvienu klientu, per Informacijos Sistema galima teikti vadovui visą informacija apie tai kaip buvo susisiekiama su klientu, ar su juo buvo susisiekta ar ne ir taip pat ar yra pardavimai ar ne. Taip pat Informacijos Sistema‘e galima susikurti atskirą pažymimą sąrašą, tam kad būtų nepamirštamos visos užduotys.

Kitas Informacijos Sistema naudojamas įmonėje nėra atskleidžiamas dėl konfendincialumo sutarties.

Šiame Informacijos Sistema tiesiog yra generuojamos visos sąskaitos siunčiamos klientui. Viskas nėra automatizuota, tai yra vykstant kiekvienam pardavimui reikalingą informacija reikia užpildyti pačiam vadybininkui, kas prailgina laiką pateikiant visą informacija klientui.

Nei vienas Informacijos Sistema nėra susieta viena su kitu. Klientams parduodami produktai taip pat neturi jokių kodų. Kas neleidžia generuoti vieningų sutarčių ir vieningų sąskaitų. Tai būtų galima išspręsti jei būtų vientisa Informacijos Sistema, be atskirų sistemų, kas išleistų įmonei mažiau pinigų ir laiko naudojant dviejus Informacijos Sistema.

## Probleminė sritis

Pagrindinės problemos naudojant 2 dabartines sistemas yra:

1. Pakankamai Brangus palaikymo mokestis.
2. Generuojant ataskaitas reikia kreiptis į AdWords administratorių
3. Generuojant sąskaitas reikalingas duomenų perkėlimas iš vienos sistemos į kitą
4. Generuojant sutartis reikalingas vadovo parašas ant kiekvienos sutarties ir po to siųsti sutartį klientui.

Užduoties analyzėje pateiksiu visus argumentus dėl ko manau, kad kiekvienas iš aukščiau paminėtų teiginių yra problema.

### Brangus palaikymo mokestis

Lexita CRM naudojamas naudotojų apmokestinimas, kuris kainuoja kiekvieną mėnesį – 30€ kiekvienam naudotojui. Skaičiuojant tai kiekvienas mėnuo šios sistemos kainuoja apie 450€ įmonei

Taip pat yra naudojamas kitas CRM, kurio palaikymo mokestis yra 300€ kiekvieną mėnesį.

Kuriant naują informacijos sistemą įmonėje nebereikėtų mokėti mėnesinio mokesčio informacijos sistemų kūrėjams. Reikalingas būtų tik skiriamas laikas sukurti naujai sistemai(darbo valandos) ir taip pat serverio palaikymo mokestis, kuris kiekvieną mėnesį kainuotų ne daugiau nei 8€. Kadangi įmonė savo domeną turi už jį nereikėtų papildomų išlaidų.

### Ataskaitų generavimas AdWords

Kaip matome iš pateikto BPMN diagramos (pav. 3) galima teikti jog yra naudojamas dviejų žmonių laikas atlikti vienam darbui. Sistemos sukūrimo atveju reikėtų tik vadybininko, duomenims peržiūrėti, kai yra generuojama ataskaita.

  
pav. 3. AdWords kampanijos ataskaita.

### Sąskaitų generavimas

Generuojant sąskaitas (pav. 4) reikalingas duomenų paėmimas iš kito CRM tam kad būtų galima sugeneruoti sąskaitas.



pav. 4. Sąskaitų generavimas

### Sutarčių generavimas

Šioje diagramoje (pav. 5) galime matyti jog kol sutartis yra pasirašoma užtrunkama nemažai laiko, tai galima išspręsti pakeitus sutarties sąlygas ir taip leisti skenuotiems pasirašytiems dokumentams būti nusiustiems el. Paštu. Tada sistema galėtų sugeneruoti tokį dokumentą ir nusiųsti klientui.

  
pav. 5. Sutarčių generavimas.

## CodeIgniter PHP karkasas kas tai?

CodeIgniter yra PHP karkasas paremtas „Model-Controller-View“ arba kitaip MVC principu. CodeIgnitor iš kitų PHP karkasų išsiskiria savo paprastumu ir lengvumu.

**CodeIgniter yra aplikacijų kūrimo karkasas**

Šis karkasas yra skirtas kurti sistemoms nuo pat pradžių. Šio karkaso tikslas yra kuo greičiau sukurti reikalinga programinę įrangą, nenaudojant didelių resursų. Dėl MVC principo ir pernaudojamų funkcijų. Taip pat šis karkasas turi labai paprastą priėjimą prie savo bibliotekų, kurios jau yra paruoštos naudojimui. Tuo remdamiesi galime greičiau sukurti mums reikalingas aplikacijas.

**CodeIgniter yra visiškai nemokamas**

CodeIgniter licenzija yra visiškai nemokama ir atvirojo kodo.

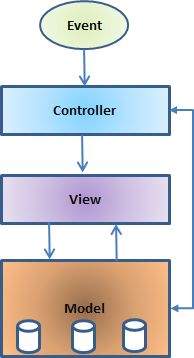
**CodeIgniter yra lengvas**

Bazinė šio karkaso versija užima vos kelis megabaitus, kas labai stipriai skiriasi nuo kitų PHP karkasų. Kas leidžia sutelkti daugiau resursų ten kur jų reikia, ypač jeigu yra naudojamas ribotos vietos serveris. Taip pat šis karkasas skiriasi nuo kitų karkasų tuo, jog visos bibliotekos yra įkraunamos tik tada kai jos iš tikro reikalingos. Dėl šios priežasties viskas yra užkraunama žymiai greičiau.

## MVC programavimo principas

PHP kalba galima rašyti skriptus ir taip pat programuoti pritaikant principus, kurie palengvina darbą programuotojui ir vėliau tirinėjantiems sistemą žmonėms. Skriptų programavimas pasižymi tuo jog, kodas dažniausiai yra labai didelis ir užimantis daug vietos serveryje, taip pat nėra MVC būdingo kodo panaudojimumo. Skriptų rašymas pradedančiojo PHP programuotojo įrankis, vėliau išmokus šią programavimo kalbą, programuotojai persikelia į MVC principu paremtus karkasus.

Pagrindinis MVC tikslas struktūrizuoti kodą ir tuo pačiu padidinti kodo panaudojimumą.

  
pav. 6. MVC principas – diagram (šaltinis: <https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm>)

**Modelis –** Skirtas atiduoti ir apdoroti duomenis kontroleriui. Visos duomenų migracijos. Duomenų bazė nebūtinai gali būti duomenų ir lentelių kratinys. Tai gali būti XML ar CSV failas kuriame yra saugomi duomenys. Modelis dažniausiai jau turi užklausų rašymo variklį, kuris pagreitina programuotojo programavimo darbą.

**Kontroleris –** Skirtas apdoroti informacijai gautai iš įvykio ir iš duomenų bazės, visas programinės įrangos funkcionalumas yra aprašomas būtent kontroleryje, nesvarbu ar tai būtų užklausos apdorojimas, ar tai būtų duomenys iš modelio. Kontrolerio funkcijos gali būti naudojamos daug kartų be apribojimų.

**View(Atvaizdas) –** Skirtas parodyti apdorotiems duomenims. Pagrindinis View tikslas yra parodyti informaciją, kurią būtų patogu matyti pagrindiniam sistemos naudotojui, kuris neturi nieko bendro su programavimu, tai yra stilius, animacijos, filtrai, laukai.

Paprasčiausias būdas paaiškinti MVC principą tiesiog sulyginus su mums natūraliais dalykais, pavyzdžiui Pica. Mes įvykdome įvykį užsakinėdami picą. Tada picų kepėjas priėmęs mūsų užsakymą jį apdoroją(tai būtų mūsų kontroleris), taip jis naudodamas modelius: picos paplotį, igridientus, krosnį . Sukuria mums vaizdą, kurį matome galutiniame produkte, tai yra pagaminta garuojanti pica(vaizdas).

## Alternatyvos



### Perkant jau sukurtą ryšių su klientais informacijos sistemą

Pagrindinė problemą renkantis šią alternatyvą būtų rasti ryšių su klientais informacijos sistemą, kuri turėtų integruotą AdWords valdymo sistemą. Kadangi AdWords API dar yra pakankamai naujas būdas pasiimti duomenis iš AdWords kampanija tai šiuo metu nėra gerai veikiančių Ryšių su klientais informacijos sistemų.  
Taip pat perkant jau sukurtą ryšių su klientais informacijos sistemą įkainiai priklauso arba nuo vartotojų skaičiaus arba yra nustatytas mėnesinis mokestis.

Taip pat perkant ryšių su klientais informacijos sistemą susiduriame su vienu didžiausiu minusų, tai – šios informacijos sistemos bus neįmanoma redaguoti, nes kodas yra slepiamas nuo kliento.

### Integruoti AdWords į dabar naudojamas sistemas

Pagrindinė dalis dėl ko yra nenaudojama ši alternatyva yra tai, jog toks įprogramavimas kainuotų papildomus pinigus, kurie yra investuojami jau į dabartinę informacijos sistemą.

Taip pat Informacijos Sistemai tinkamai suprogramuoti reikia programuotojo, kuris būtų susidūręs su būtent AdWords API galimybėmis ir būtų susipažinęs, kokie duomenys yra būtini klientui, tai aišku galima nustatyti per funkcinius reikalavimus, bet taip pat kaip ir prieš tai minėtoje informacijos sistemoje šių informacijos sistemų kodas yra uždaras, kas atima galimybę toliau tobulinti informacijos sistemą.

# Projektinė dalis

## Funkcinių ir nefunkcinių reikalavimų nustatymas

**Būsimos CRM funkciniai reikalavimai:**

1. CRM turi turėti klientų kontaktų saugojimą
2. CRM turi turėti funkciją skambinti tiesiogiai iš programos
3. CRM turi turėti sąskaitų generavimo funkciją
4. CRM turi paimti duomenis iš AdWords kampanijų
5. CRM turi imti duomenis iš PageSpeed API
6. CRM turi turėti individualius numerius kiekvienai paslaugai
7. CRM turi turėti ataskaitų generavimo funkciją
8. CRM turi turėti galimybę susidaryti užduočių sąrašą
9. CRM turi turėti individualius prisijungimus
10. CRM turi leisti darbuotojams pildyti užduotis
11. CRM turi leisti prisegti failus prie užduočių
12. CRM užduotys turi turėti kategorijas
13. Registraciją prie CRM gali vykdyti tik direktorius arba administratorius

**Būsimo CRM nefunkciniai reikalavimai:**

1. Serveris turi palaikyti MySql DBVS
2. Serveris turi palaikyti PHP 5.5 arba aukštesnę versiją.
3. Serverio užkrovimas po užklausos 0.5s maksimaliai
4. Serveris turi palaikyti CodeIgniter 2.0 arba aukštesnę sistemą
5. Serverio maksimali kaina mėnesiui neturi viršyti 8€
6. Maksimalus programinės įrangos užimamas Production state turėtų būti ne daugiau kaip 1GB
7. Sistema turi blokuoti Google, Yandex, Yahoo paieškos variklių skenavimą

## Ryšių su klientais informacijos sistemos use case diagrama

Šiame skyriuje yra patalpinta sistemos Use Case diagrama, kuri nurodo kokias sistema turi atlikti funkcijas jog ji atitiktų funkcinius reikalavimus.

  
Pav. 7. USE CASE diagrama.

## Sistemos veiklos diagramos

Šiame skyriuje bus nurodomos veiklos diagramos bei jas aprašančios lentelės.

  
Pav. 8. Panaudos atvejis "Naujo naudotojo registracija"

Lentelė Nr. 2. Naujo naudotojo registracija

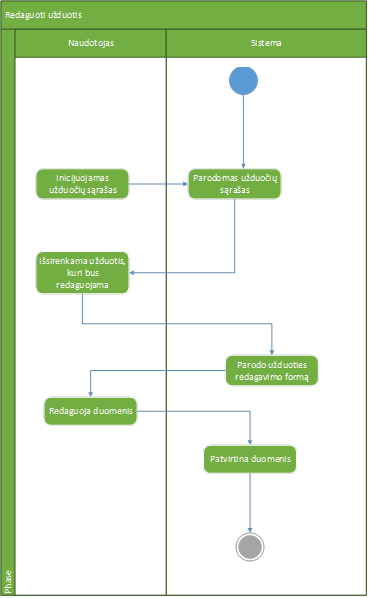
|  |  |
| --- | --- |
| **Pavadinimas** | Naujo naudotojo registracija |
| **ID** | UC02 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas užregistruoti naują naudotoją |
| **Naudotojas (-ai)** | Direktorius |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja naujo naudotojo sukūrimą 2. Parodoma naujo naudotojo registravimo forma 3. Suvedami duomenys 4. Išsiunčiamas el. laiškas naujam naudotojui |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Užregistruotas naujas naudotojas |
| **Papildoma informacija** | Informacija apie naudotoją  +Prisijungimas  +Slaptažodis |

  
Pav. 9. Panaudos atvejis "Naujos užduoties kūrimas"

Lentelė Nr. 3. Naujos užduoties kūrimas

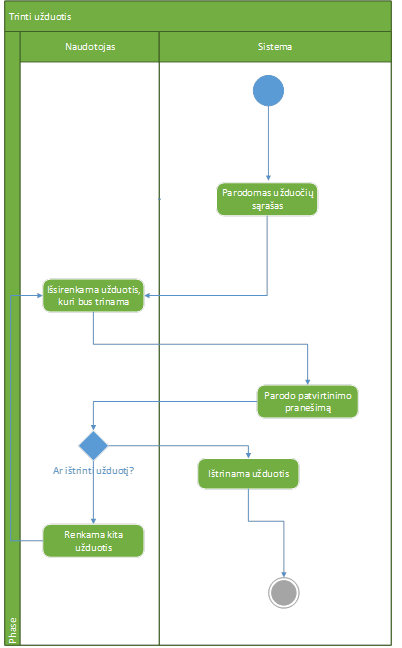
|  |  |
| --- | --- |
| **Pavadinimas** | Naujos užduoties kūrimas |
| **ID** | UC03 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas sukurti naują užduotį |
| **Naudotojas (-ai)** | Direktorius, naudotojas |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja naujos užduoties kūrimo formą 2. Sistema parodo naujos užduoties kūrimo formą 3. Suvedami duomenys 4. Sukūriama nauja užduotis |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Sūkuriama nauja užduotis |
| **Papildoma informacija** | Informacija apie uždotį  +id  +Užduoties pavadinimas  +užduoties aprašymas |

  
Pav. 10. Panaudos atvejis "Redaguoti užduotis"



Lentelė Nr. 4. Redaguoti užduotis

|  |  |
| --- | --- |
| **Pavadinimas** | Redaguoti užduotį |
| **ID** | UC04 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas redaguoti užduočiai |
| **Naudotojas (-ai)** | Direktorius, Naudotojas |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja užduoties redagavimą 2. Sistema parodo užduoties redagavimo formą 3. Keičiami duomenys 4. Užduotis pakeista |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Užduotis redaguota |
| **Papildoma informacija** |  |

  
Pav. 11. Panaudos atvejis "Trinti užduotis"

Lentelė Nr. 5. Trinti užduotis

|  |  |
| --- | --- |
| **Pavadinimas** | Trinti užduotį |
| **ID** | UC05 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas trinti užduotims |
| **Naudotojas (-ai)** | Direktorius, Naudotojas |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja užduoties trinimą 2. Sistema ištrina užduotį |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Užduotis ištrinta |
| **Papildoma informacija** |  |

  
Pav. 12. Panaudos atvejis "Generuoti sutartį"

Lentelė Nr. 6. Generuoti sutartis

|  |  |
| --- | --- |
| **Pavadinimas** | Generuoti sutartį |
| **ID** | UC06 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas generuoti sutartims |
| **Naudotojas (-ai)** | Direktorius, Naudotojas |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja sutarties generavimą 2. Sistema sugeneruoja naują sutartį ir išsiunčia ją klientui |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Sutartis sugeneruota |
| **Papildoma informacija** |  |

  
Pav. 13. Panaudos atvejis "Generuoti sąskaitas"

Lentelė Nr. 7. Generuoti sąskaitas

|  |  |
| --- | --- |
| **Pavadinimas** | Generuoti sąskaitas |
| **ID** | UC07 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas sąskaitų generavimui |
| **Naudotojas (-ai)** | Direktorius, Naudotojas |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja sąskaitų generavimą 2. Parodomas sutarčių sąrašas 3. Pasirenka sutartį 4. Sugeneruoja sąskaita ir išsiunčia klientui el. paštu |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Sąskaita sugeneruota |
| **Papildoma informacija** |  |

  
Pav. 14. Panaudos atvejis "Pridėti failą prie užduoties"

Lentelė Nr. 8. Pridėti failą prie užduoties

|  |  |
| --- | --- |
| **Pavadinimas** | Pridėti failą prie užduoties |
| **ID** | UC09 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas failų pridėjimui |
| **Naudotojas (-ai)** | Direktorius, Naudotojas |
| **Priešsąlygos** | Turi būti atidarytas puslapis |
| **Pagrindinis scenarijus** | 1. Naudotojas pasirenka užduotis 2. Parodomos užduotys 3. Paspaudžiama pridėti failą 4. Parodoma failo pridėjimo forma 5. Pridedamas failas |
| **Alternatyvūs scenarijai** |  |
| **Posąlygos** | Failas pridėtas prie užduoties |
| **Papildoma informacija** |  |

  
 Pav. 15. Panaudos atvejis "Peržiūrėti PageSpeed informaciją"

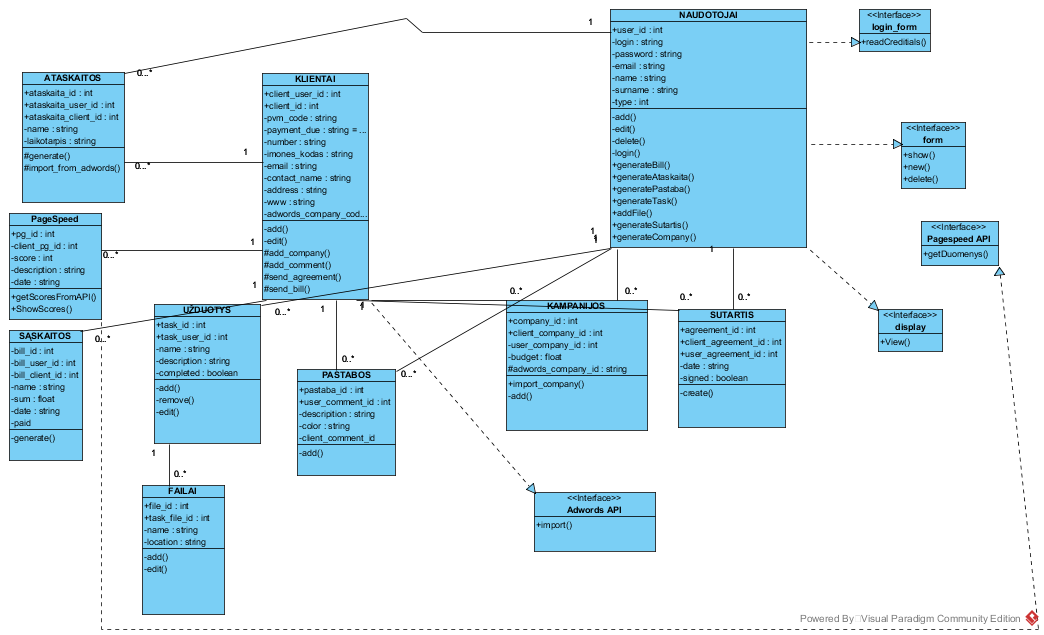
Lentelė Nr. 9. Peržiūrėti PageSpeed informaciją

|  |  |
| --- | --- |
| **Pavadinimas** | Peržiūrėti PageSpeed informaciją |
| **ID** | UC10 |
| **Trumpas aprašymas** | Panaudos atvejis skirtas aprašyti PageSpeed informacijos parodymui |
| **Naudotojas (-ai)** | Direktorius, Naudotojas |
| **Priešsąlygos** | Turi būti prisijungta prie CRM |
| **Pagrindinis scenarijus** | 1. Naudotojas pasirenka jog nori kad būtų rodoma kliento PageSpeed informacija 2. Tikrinama ar klientui įvestas puslapis 3. Parodoma PageSpeed informacija |
| **Alternatyvūs scenarijai** | 3a. Parodoma jog klientui nėra priskirta nei viena interneto svetainė. |
| **Posąlygos** |  |
| **Papildoma informacija** | + PageSpeed API duomenys |

## Sistemos klasių diagramos aprašymas

### Sistemos klasių diagrama

Šiame skyriuje bus nurodoma klasių diagrama. Kuri nurodo sistemos architektūrą ir vėliau programuotojams padės sukurti sistmą.

  
pav. 16. UML Class Diagrama

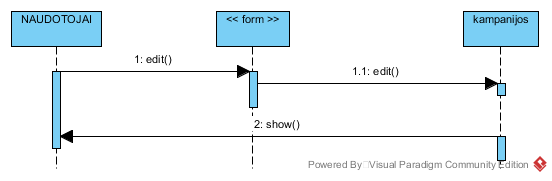
**Didesnė UML diagrama galima pamatyti 2 priede.**

Lentelė Nr. 10. Klasių diagramos klasės, atributai, operacijos

|  |  |  |
| --- | --- | --- |
| **Eil. Nr.** | **Klasė** | **Komentaras** |
|  | Pav. 17. Klasė klientai | **Atributai:**  Client\_user\_id – naudotojo id skirtas nustatyti prie kurio naudotojo yra priskirtas klientas  Client\_id – kliento id generuojamas automatiškai  Pvm\_code – kliento įmonės kodas  Payment\_due – nustatytas terminas per kurį reikia klientui susimokėti(numatytas 15)  Number – kliento telefono numeris  Imones\_kodas – kliento imones kodas  Email – kliento elektroninis paštas  Contact name – kontaktinio asmens vardas pavardė  Address – įmonės buveinės adresas  www – įmonės internetinis adresas  adwords\_company\_code – Jei yra reklamuojamasi AdWords čia įrašomas kampanijos id pagal kurį vėliau bus generuojamos ataskaitos.  **Operacijos:**  Add() – pridėti naują klientą  Edit() – keisti jau esamą klientą  add\_comment() – skambinant klientui pridėti komentarą  send\_agreement() – nusiųsti klientui sutartį el. Paštu  send\_bill() – nusiųsti klientui sąskaitą el. Paštu |
|  | Pav. 18. Klasė naudotojai | **Atributai:**  user\_id – naudotojo id  login – naudotojo prisijungimo vardas  password – naudotojo prisijungimo slaptažodis  email – naudotojo elektroninis paštas  name – naudotojo Vardas  surname – naudotojo pavardė  type – naudotojo tipas (administratorius arba paprastas vartotojas)  **Operacijos:**  Add() – pridėti naudotoją  Edit() – keisti naudotoją  Login() – naudotojo prisijungimas |
|  | Pav. 19. klasė ataskaitos | **Atributai:**  Ataskaita\_id – ataskaitos id  Ataskaita\_user\_id – naudotojo id, kuriam priklauso ataskaita  Ataskaita\_client\_id – kliento id, kuriam ataskaita yra generuojama  Name – ataskaitos pavadinimas  Laikotarpis – laikotarpis, kuriam ataskaita yra generuojama  **Operacijos:**  Generate() – ataskaitos generavimas  Import\_from\_adwords() – AdWords informacijos importavimas |
|  | Pav. 20. Klasė PageSpeed | **Atributai:**  Pg\_id – pagespeed duomenų id  Client\_pg\_id – klientui kuriam priskirtas PageSpeed rezultatas  Score – PageSpeed skiriamas balas  Description – PageSpeed klaidų sąrašas  Date – data kada atliktas testas  **Operacijos:**  GetScoresFromAPI() – gaunama informacija iš API  showScores() – parodoma informacija |
|  | Pav. 21. Klasė sąskaitos | **Atributai:**  Bill\_id – sąskaitos id  Bill\_user\_id – naudotojo id, kuris sugeneravo sąskaitą  Bill\_client\_id – kliento id, kuriam priskirta sąskaita  Name – sąskaitos pavadinimas  Sum – sąskaitos suma  Date – sąskaitos išrašymo data  Paid – sąskaitos būsena(sumokėta ar nesumokėta)  **Operacijos:**  Generate() – sąskaitos generavimo operacija |
|  | Pav. 22. Klasė užduotys | **Atributai:**  Task\_id – užduoties id  Task\_user\_id – naudotojo sukūrusio užduotį id  Name – užduoties pavadinimas  Description – užduoties pavadinimas  Completed – ar užduotis pabaigta  **Operacijos:**  Add() – sukurti naują užduotį  Edit() – redaguoti esamą užduotį  Delete() – Ištrinti esamą užduotį |
|  | Pav. 23. Klasė Failai | **Atributai:**  File\_id – failo id  Task\_file\_id – užduoties id, kuriai yra priskirtas failas  Name – failo pavadinimas  Location – failo lokacija serveryje  **Operacijos:**  Add() – pridėti failą |
|  | Pav. 24. Klasė pastabos | **Atributai:**  Pastaba\_id – pastabos id  User\_comment\_id – naudotojo, kuris pakomentavo id  Description – komentaras  Color – komentaro spalva  **Operacijos:**  Add() – pridėti komentarą |
|  | Pav. 25. Klasė kampanijos | **Atributai:**  Company\_id – kampanijos id  Client\_company\_id – kliento kuriam priskirta kampanija id  User\_company\_id – naudotojas, kuris prižiūri kampanija  Budget – kampanijos biudžetas  Adwords\_company\_id – kampanijos id  **Operacijos:**  Import\_company() – kampanijos duomenų paėmimas iš AdWords API  Add() – kampanijos pridėjimas |
|  | Pav. 26. klasė sutartys | **Atributai:**  Agreement\_id – sutarties id  Client\_agreement\_id – kliento kuriam priskirta sutartis id  User\_agreement\_id – naudotojas, kuris sugeneravo sąskaitą  Date – data kada buvo sugeneruota sutartis  Signed – ar pasirašyta sutartis  **Operacijos:**  Create() – sutarties generavimas |

## Sistemos dinaminis modelis

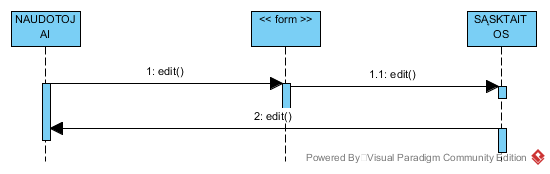
Šiame skyriuje bus nurodomos diagramos suijusios su veiklos diagramomis tik viskas bus rodoma visiškai iš sistemos pusės, tai yra kaip Sistema bus valdoma.



pav. 27. Keisti kampanijas

Lentelė Nr. 11. KeistiKampanijas

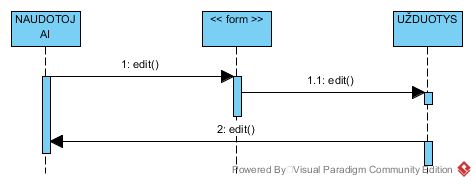
|  |  |
| --- | --- |
| Panaudos atvejis: | KeistiKampanijas |
| Trumpas aprašymas | Panaudos atvejis skirtas pakeisti Kampanijas |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja kampanijos redagavimo veiskmą 2. Formos intefeicas parodo formą kampanijos keitimui 3. Kampanija yra pakeičiama |
| Posąlyga |  |
| Alternatyvus srautas |  |



pav. 28. Keisti sąskaitas

Lentelė Nr. 12. keistiSąskaitas

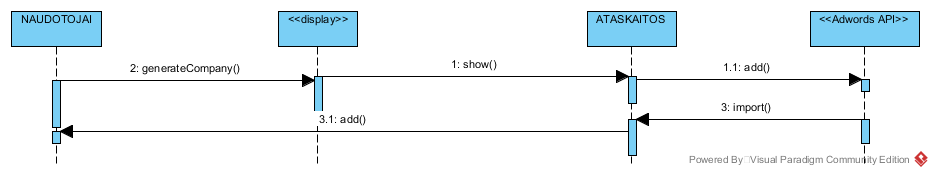
|  |  |
| --- | --- |
| Panaudos atvejis: | KeistiSąskaitas |
| Trumpas aprašymas | Panaudos atvejis skirtas pakeisti sąskaitas |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja sąskaitos redagavimo veiskmą 2. Formos intefeicas parodo formą sąskaitos keitimui 3. Kampanija yra pakeičiama |
| Posąlyga |  |
| Alternatyvus srautas |  |



pav. 29. keisti užduotis

Lentelė Nr. 13. KestiUžduotis

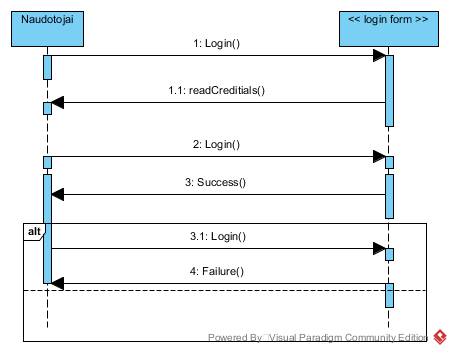
|  |  |
| --- | --- |
| Panaudos atvejis: | KeistiUžduotis |
| Trumpas aprašymas | Panaudos atvejis skirtas pakeisti užduotims |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja užduoties redagavimo veiskmą 2. Formos intefeicas parodo formą užduoties keitimui 3. Užduotis yra pakeičiama |
| Posąlyga |  |
| Alternatyvus srautas |  |



pav. 30. Parodyti ataskaitas

Lentelė Nr. 14. ParodytiAtaskaitas

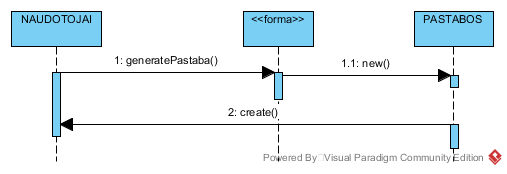
|  |  |
| --- | --- |
| Panaudos atvejis: | ParodytiAtaskaitas |
| Trumpas aprašymas | Panaudos atvejis skirtas Parodyti duomenims iš AdWords API naudojant tuos duomenis |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja kliento kampanijos parodymą 2. Interfeisas kreipiasį į ataskaitų klasę 3. Klasė kreipiasi į Adwords API interfeisą ir gauna informaciją apie klientą 4. Parodoma visa informacija apie AdWords kampanija |
| Posąlyga |  |
| Alternatyvus srautas |  |



pav. 31. Prisijungti

Lentelė Nr. 15. Prisijungti

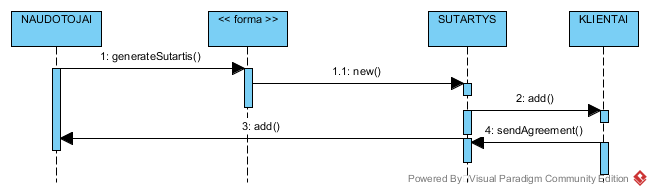
|  |  |
| --- | --- |
| Panaudos atvejis: | Prisijungti |
| Trumpas aprašymas | Panaudos atvejis skirtas naudotojo prisijungimui |
| Aktorius | Naudotojas |
| Priešsąlyga |  |
| Pagrindinis srautas | 1. Naudotojas inicijuoja prisijungimą 2. Naudotojas suveda visą prisijungimo informaciją 3. Informacija patikrinama 4. Naudotojas prisijungia |
| Posąlyga |  |
| Alternatyvus srautas | 1. Naudotojo informacija yra netinkama 2. Parodomas pranešimas jog informacija nėra tinkama |



pav. 32. Sukurti pastabą

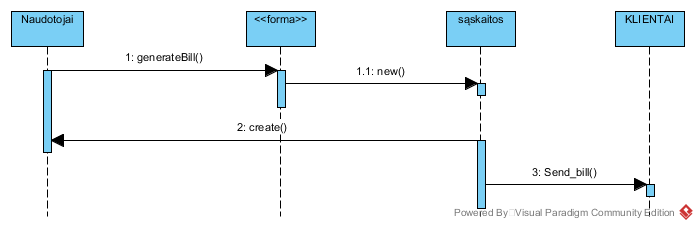
Lentelė Nr. 16. sukurtiAtaskaitą

|  |  |
| --- | --- |
| Panaudos atvejis: | sukurtiAtaskaitą |
| Trumpas aprašymas | Panaudos atvejis skirtas sukurti ataskaitai |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja pastabos sukūrimą 2. Parodoma pastabos sukūrimo forma 3. Sukūriama nauja pastaba |
| Posąlyga |  |
| Alternatyvus srautas |  |

  
pav. 33. sukurti sutartį

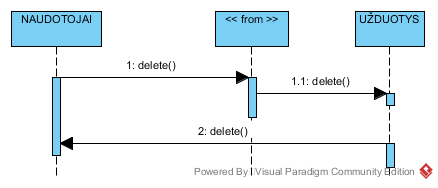
Lentelė Nr. 17. sukurtiSutartį

|  |  |
| --- | --- |
| Panaudos atvejis: | Sukurti sutartį |
| Trumpas aprašymas | Panaudos atvejis skirtas sukurti sutarčiai |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja sutarties sukūrimą 2. Parodoma sutarties sukūrimo forma 3. Sukūriama nauja sutartį 4. Išsiunčiamas el. Laiškas klientui su sutartimi |
| Posąlyga |  |
| Alternatyvus srautas |  |

  
pav. 34. sukurtiSąskaitą

Lentelė Nr. 18. sukurtisąskaitą

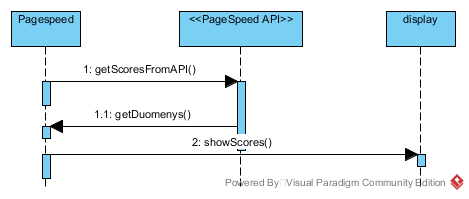
|  |  |
| --- | --- |
| Panaudos atvejis: | Sukurti sąskaitą |
| Trumpas aprašymas | Panaudos atvejis skirtas sukurti sąskaitai |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja sąskaitos sukūrimą 2. Parodoma sąskaitos sukūrimo forma 3. Sukūriama nauja sąskaita 4. Išsiunčiamas el. Laiškas klientui su sąskaita |
| Posąlyga |  |
| Alternatyvus srautas |  |



pav. 35. Trinti užduotis

Lentelė Nr. 19. Trinti Užduotis

|  |  |
| --- | --- |
| Panaudos atvejis: | Sukurti užduotis |
| Trumpas aprašymas | Panaudos atvejis skirtas trinti užduotim |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja užduoties trinimą 2. Užduotis yra ištrinama |
| Posąlyga |  |
| Alternatyvus srautas |  |

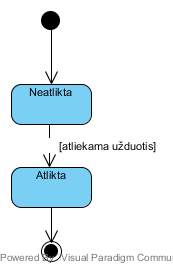
  
pav. 36. Parodyti PageSpeed API informaciją

Lentelė Nr. 20. Parodyti PageSpeed informacija

|  |  |
| --- | --- |
| Panaudos atvejis: | Parodyti PageSpeed informaciją |
| Trumpas aprašymas | Panaudos atvejis skirtas parodyti informacijai iš pageSpeed API |
| Aktorius | Naudotojas |
| Priešsąlyga | Naudotojas turi būti prisijungęs |
| Pagrindinis srautas | 1. Naudotojas inicijuoja Score gavimą 2. Sistema kreipiasi į API su internetinio puslapio adresu 3. API gražina puslapio taškus ir taisytinas vietas 4. Sistema parodo duomenis |
| Posąlyga |  |
| Alternatyvus srautas |  |

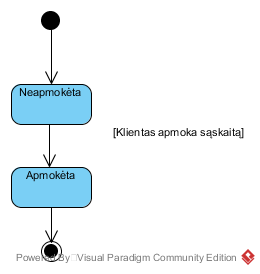
### Sistemos būsenų diagramos

Šiame skyriuje bus aparašomos dviejų klasių būsenų diagramos



Pav. 37. Užduoties klasės būsenos

Būsenų diagramoje (pav. 37): kai yra sukuriama nauja klasė, šios klasės būsena yra Neatlikta, kai jau užduotis yra atliekama ši užduotis pasidaro atlikta ir po 14d. Išsitrina.



Pav. 38. sąskaitos klasės būsenos

Būsenų diagramoje (Pav. 37): kai sugeneruojama nauja sąskaita, šios klasės būsena tampa neapmokėta, vos tik klientas apmoka sąskaitą, sąskaitos būsena tampa apmokėta.

## Duomenų bazės architektūra

Šiame skyriuje bus pateikiama sistemos duomenų bazės loginis modelis. Taip pat skyriuje bus pateikiamas esybių bei jų ryšių aprašymas.

### Duomenų bazės loginis modelis

Šiame skyriuje parodomas duomenų bazės loginis modelis (Pav. 16)

  
pav. 39. Duomenų bazės realiacinis modelis.

### Loginio modelio esybės

Šiame skyriuje bus aprašomos duomenų bazės esybės.

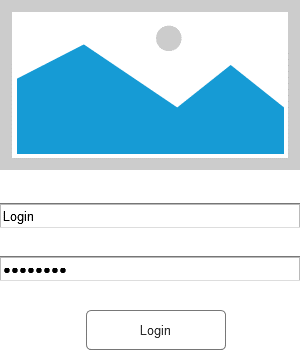
Lentelė Nr. 21. Esybės ir atributai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pavadinimas | Atributai | Duomenų tipai | Privalomumas | Apribojimai |
| KLIENTAI | Client\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Client\_user\_id | Sveikieji Skaičiai | Privalomas | Iki 100 ženklų |
| Imones\_kodas | Sveikieji Skaičiai | Neprivalomas | Iki 20 ženklų |
| Name | Tekstas | Privalomas | Iki 256 ženklų |
| Contact\_name | Tekstas | Privalomas | Iki 80 ženklų |
| Address | Tekstas | Privalomas | Iki 300 ženklų |
| www | Tekstas | Neprivalomas | Iki 50 ženklų |
| Adwords\_company\_code | Tekstas | Neprivalomas | Iki 30 ženklų |
| Email | Tekstas | Privalomas | Turi būti El. paštas |
| Number | Sveikieji skaičiai | Privalomas | Iki 9 ženklų |
| Pvm\_code | Tekstas | Privalomas | Iki 20 ženklų |
| Payment\_due | Sveikieji skaičiai | Privalomas | Iki 2 ženklų |
| NAUDOTOJAI | User\_id | Sveikieji skaičiai | Privalomas | Generuojamas automatiškai |
| Login | Tekstas | Privalomas | Iki 20 ženklų |
| Password | Tekstas | Privalomas | Iki 50 ženklų |
| Email | Tekstas | Privalomas | Turi būti El. paštas |
| Name | Tekstas | Privalomas | Iki 20 ženklų |
| Surname | Tekstas | Privalomas | Iki 30 ženklų |
| UŽDUOTYS | Task\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Task\_user\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Name | Tekstas | Privalomas | Iki 50 ženklų |
| Task\_user\_id\_pri | Sveikieji skaičiai | Privalomas | Iki 20 ženklų |
| Description | Tekstas | Privalomas | Iki 1000 ženklų |
| FAILAI | File\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Task\_file\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Name | Tekstas | Privalomas | Iki 50 ženklų |
| location | Tekstas | Privalomas | Iki 50 ženklų |
| PASTABOS | Pastaba\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| User\_comment\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Description | Tekstas | Privalomas | Iki 150 ženklų |
| Date | Data | Privalomas | Iki 15 ženklų |
| PAGESPEED | Pg\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Client\_pg\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Score | Skaičius | Privalomas | Iki 5 ženklų |
| Description | Tekstas | Privalomas | Iki 1000 ženklų |
| Date | Tekstas | Privalomas | Iki 15 ženklų |
| KAMPANIJOS | Company\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Client\_company\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| User\_company\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Budget | Tekstas | Privalomas | Iki 50 ženklų |
| Adwords\_comapny\_id | Tekstas | Privalomas | Iki 50 ženklų |
| ATASKAITOS | Ataskaita\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Ataskaita\_user\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Ataskaita\_client\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Name | Tekstas | Privalomas | Iki 50 ženklų |
| laikotarpis | Tekstas | Privalomas | Iki 50 ženklų |
| SUTARTYS | Agreement\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Client\_agreement\_id | Sveikieji Skaičiai | Privalomas | Iki 50 ženklų |
| User\_agreement\_id | Sveikieji Skaičiai | Privalomas | Iki 50 ženklų |
| Date | Data | Privalomas | Iki 30 ženklų |
| File\_name | Tekstas | Privalomas | Iki 50 ženklų |
| Path | Tekstas | Privalomas | Iki 150 ženklų |
| SĄSKAITOS | Bill\_id | Sveikieji Skaičiai | Privalomas | Generuojamas automatiškai |
| Bill\_user\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Bill\_client\_id | Sveikieji Skaičiai | Privalomas | Iki 20 ženklų |
| Name | Tekstas | Privalomas | Iki 50 ženklų |
| File\_name | Tekstas | Privalomas | Iki 50 ženklų |
| path | Tekstas | Privalomas | Iki 50 ženklų |
| paid | Sveikieji Skaičiai | Privalomas | Iki 150 ženklų |

### Ryšiai tarp esybių

* Kiekvienas NAUDOTOJAS gali peržiūrėti vieną ar daugiau UŽDUOČIŲ
* Kiekviena UŽDUOTIS privalo turėti vieną NAUDOTOJĄ
* Kiekvienas NAUDOTOJAS gali pasirašyti vieną ar daugiau SUTARČIŲ
* Kiekviena SUTARTIS privalo būti priskita vienam NAUDOTOJUI
* Kiekvienas NAUDOTOJAS gali sukurti viena ar daugiau SASKAITŲ
* Kiekviena SĄSKAITA turi būti sugeneruota NAUDOTOJO
* Kiekvienas NAUDOTOJAS gali generuoti viena ar daugiau ATASKAITŲ
* Kiekviena ATASKAITA privalo būti sugeneruota NAUDOTOJO
* Kiekvienas NAUDOTOJAS gali sudaryti vieną ar daugiau KAMPANIJŲ
* Kiekviena KAMPANIJA privalo būti priklausoma NAUDOTOJUI
* Kiekvienas NAUDOTOJAS gali užrašyti vieną ar daugiau PASTABŲ
* Kiekviena PASTABA provalo būti priskirta NAUDOTOJUI
* Kiekvienas NAUDOTOJAS gali peržiūrėti viena ar daugiau UŽDUOČIŲ
* Kiekviena UŽDUOTIS privalo turėti vieną NAUDOTOJĄ
* Kiekviena UŽDUOTIS gali valdyti vieną ar daugiau FAILŲ
* Kiekvienas FAILAS privalo būti priskirtas vienai UŽDUOČIAI
* Kiekvienas Pagespeed privalo būti Priklausyti vienam klientui
* Kiekvienas KLIENTAS gali turėti vieną ar daugiau pagespeed

## Sistemos prototipas

  
pav. 40. Sistemos prototipo pagrindidinis langas

Sistemos prototipą galima peržiūrėti nuorodoje: <http://3i0t71.axshare.com>

# EKONOMINIS PAGRINDIMAS

## Baigiamojo darbo administravimo išlaidos

Baigiamajam darbui buvo skirta XXX,XX €

Baigiamojo darbo atlikimo kaina:

* 3,125 € \* 42 d.d. \* 8 h = 1050€.

Atliekame skaičiavimus jog vidutinis valandinis atlyginimas į rankas yra 3,125€ (mėnesinis atlyginimas 500€)

Baigiamojo darbo visas laikas: 42 darbo dienos

Viena darbo diena sudaro 8 valandas

Kitos išlaidos:

* Spausdinimas: 120 lapų \* 0,1 EUR/lap. = 12€.

Skaičiuojame jog baigiamasis darbas užims 120lapų, skaičiuojamas nespalvotas spausdinimas 120 lapų.

Bendri baigiamojo darbo kaštai: 289,62 € + 1050 € + 12 € = 1351,62 €.

## Sistemos sukūrimo kaštai

Sistemai sukurti reikalingi programavimo darbai. Kadangi įmonėje, kurioje atliekami programavimo darbai įkainis yra 42,35€(su PVM), tai sistemos kaina priklauso nuo darbuotojo laiko sugaišto programuojant sistemą.

Sistema buvo sukurta per mėnesį, taip pat pridedame jog Sistema buvo testuojama dar 2 savaites(tai įeina į programavimo darbus)

Taigi turime jog Sistema buvo programuojama apie 30 darbo dienų tai yra: 30 d.d.\*8h = 240 valandų.

Taigi turime jog sistemos sukūrimo kaštai buvo tokie: 42,35 €/h \* 240h = 10 164€

## Sistemos palaikymo kaštai per menėsį

Sistemos palaikymo kaštai yra tokie:

* Serveris sistemai 7.26€/mėn. (serverio planas universalus serveriai.lt)
* Taip pat yra taikomas sistemos palaikymo mokestis 150€/mėn (kiekvieną mėnesį yra peržiūrima ar neatsiranda naujų klaidų sistemoje, taip pat padaromos atsarginės duomenų bazės kopijos, atsižvelgiama į sistemos naudotojų pageidavimus ir atnaujinimus)

## Sistemos nauda įmonei

Sukurta informacijos Sistema leis darbuotojam sutaupyti apie 2h į mėnesį darbo laiko AdWords kampanijos peržiūrai(kadangi Sistema jau pateikia šiuos duomenis).

Taip pat kiekvieną mėnesį darbuotojai sutaupys laiko generuodami sąsakaitas klientams, nebereikės imti duomenų iš vienos sistemos ir juos perkelti į kitą, nes viskas bus vienoje vietoje, tai sutaupys apie 15min nuo darbo dienos.

Taip pat nebereikės įmonei mokėti kitų sistemų palaikymo mokesčių, kur vienos sistemos palaikymo mokestis yra: 450€, o kitos 300€. Jau matome jog įmonė sutaupo 300+450 = 750€/mėn.

Taip pat sistema sumažina direktoriaus darbo apkrova, nes sutartys yra siunčiamos jau pasirašytos.

## Sistemos atsiperkamumas

Sistema sukūrimo kaina: 10 164€

Sistemos kaina mėnesiui: 7.26€ + 150€ = 157.26€

Sistemos sutaupymas kiekvieną mėnesį: 750€ + 2\*3.125€ + 15/60\*8\*20\*3.125 = 881.25€

Sistemos atsiperkamumo diagrama:

Lantelė Nr. 1. Atsiperkamumo lentelė

|  |  |  |
| --- | --- | --- |
| Mėnuo | Balansas | Įplaukos |
| 1 | -10164 | 723,99 |
| 2 | -9440,01 | 723,99 |
| 3 | -8716,02 | 723,99 |
| 4 | -7992,03 | 723,99 |
| 5 | -7268,04 | 723,99 |
| 6 | -6544,05 | 723,99 |
| 7 | -5820,06 | 723,99 |
| 8 | -5096,07 | 723,99 |
| 9 | -4372,08 | 723,99 |
| 10 | -3648,09 | 723,99 |
| 11 | -2924,1 | 723,99 |
| 12 | -2200,11 | 723,99 |
| 13 | -1476,12 | 723,99 |
| 14 | -752,13 | 723,99 |
| 15 | -28,14 | 723,99 |
| 16 | 695,85 | 723,99 |
| 17 | 1419,84 | 723,99 |
| 18 | 2143,83 | 723,99 |
| 19 | 2867,82 | 723,99 |
| 20 | 3591,81 | 723,99 |
| 21 | 4315,8 | 723,99 |
| 22 | 5039,79 | 723,99 |
| 23 | 5763,78 | 723,99 |
| 24 | 6487,77 | 723,99 |

pav. 41. Sistemos atsiperkamumo grafikas

Kiekvienas stulpelis žymi kiokia yra suma kiekvieną mėnesį sistemos atžvilgiu. Sistema atsipirks po 16 mėn.

Sistemos Balansas yra žymimas oranžine spalva, o įplaukos pilka spalva.

# Išvados ir pasiūlymai

Išnagrinėta įmonės veikla, įmonės procesai. Įmonėje nustatytos trys problemos: daug laiko užtrunkama generuoti sąskaitos, per daug laiko užtrunkama generuoti ataskaitoms, per daug laiko užtrunkama generuoti sutartims. Dauguma darbo buvo daroma rankiniu būdu, taip pat buvo nustatytas nereikalingų dokumentų – sutarčių be parašų saugojimas.

Atlikus informacijos šaltinių analizę buvo pasirinktas CodeIgniter karkasas dėl savo paprastumo ir greito žinių įsisavinimo - dėl paprastos dokumentacijos. Taip pat išnagrinėjus informacijos šaltinius buvo pasirinktos ir kitos priemonės projektuoti būsimai sistemai.

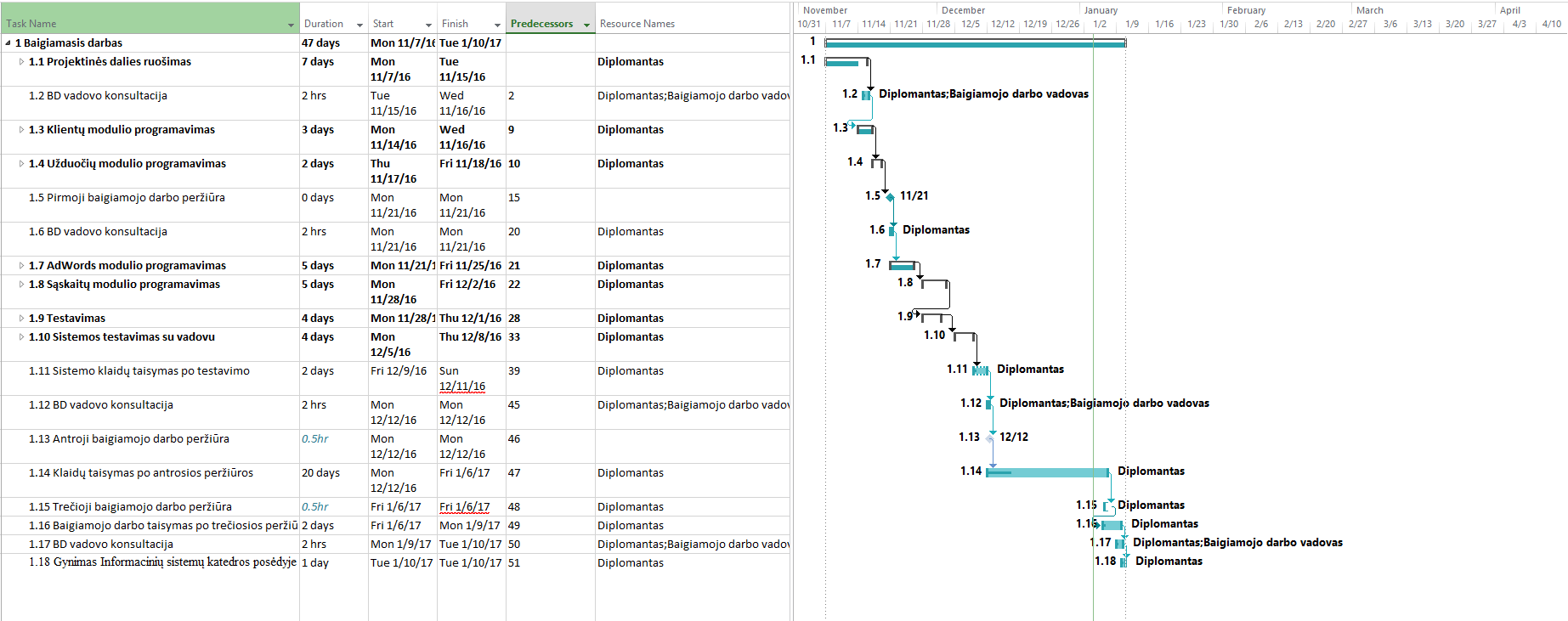
Atlikus reikalingos sistemos analizę buvo nustatyti sistemos funkciniai ir nefunkciniai reikalavimai. Buvo sėkmingai sudarytas naujos sistemos projektas. Sudaryti statinis ir dinaminis sistemos modeliai.

Sėkmingai realizuotas sistemos projektas. Atitinkantis visus funkcinius ir nefunkcinius reikalavimus. Sukurtos diegimo ir naudojimosi instrukcijos padedančios naudotojams greičiau pradėti naudoti naująją sistemą.

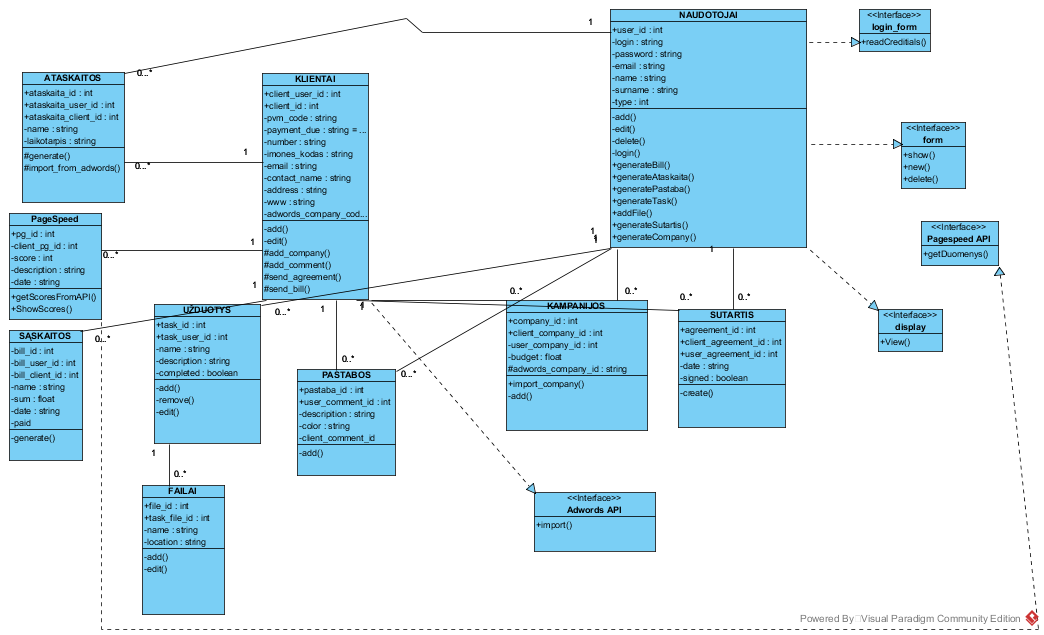
# informacijos šaltinių sąrašas

1. Griffiths, Adam. *Codeigniter 1.7 Professional Development*. Birmingham : Packt Publishing. 2010, 60p. ISBN 9781849510905;
2. Lecky-Thompson, Ed; Eide-Goodman,Heow; Nowicki, Steven D.; Cove, Alec. *Professional PHP.* 2008,621 p. ISBN 0-7645-7282-2;
3. Lecky-Thompson, Guy W. *Just Enough Web Programming with XHTML, PHP, and MYSQL.* Australia : Course PTR. 2008, 30 p. ISBN 9781598634815;
4. Nikiforova, Oksana; Sejans, Janis; Cernickins, Antons. *Role of UML Class Diagram in Object-Oriented Software Development*. Computer Science (1407-7493). 2011, Vol. 47, p65-74. 10p, ISSN 1407-7493
5. *CodeIgniter Documentation* [interaktyvus]. CodeIngiter Framework [Žiūrėta 2016-12-11] Prieiga per internetą: < <https://www.codeigniter.com/user_guide/> >
6. *PHPWord Documentation* [interaktyvus]. [žiūrėta 2016-12-11] Preiga per internetą: <<https://phpword.readthedocs.io/en/latest/>>
7. *Google AdWords API documentation* [interaktyvus]. Google Inc. [žiūrėta 2016-12-11] Preiga per internetą: < <https://developers.google.com/adwords/api/docs/guides/start/> >

**1 priedas**



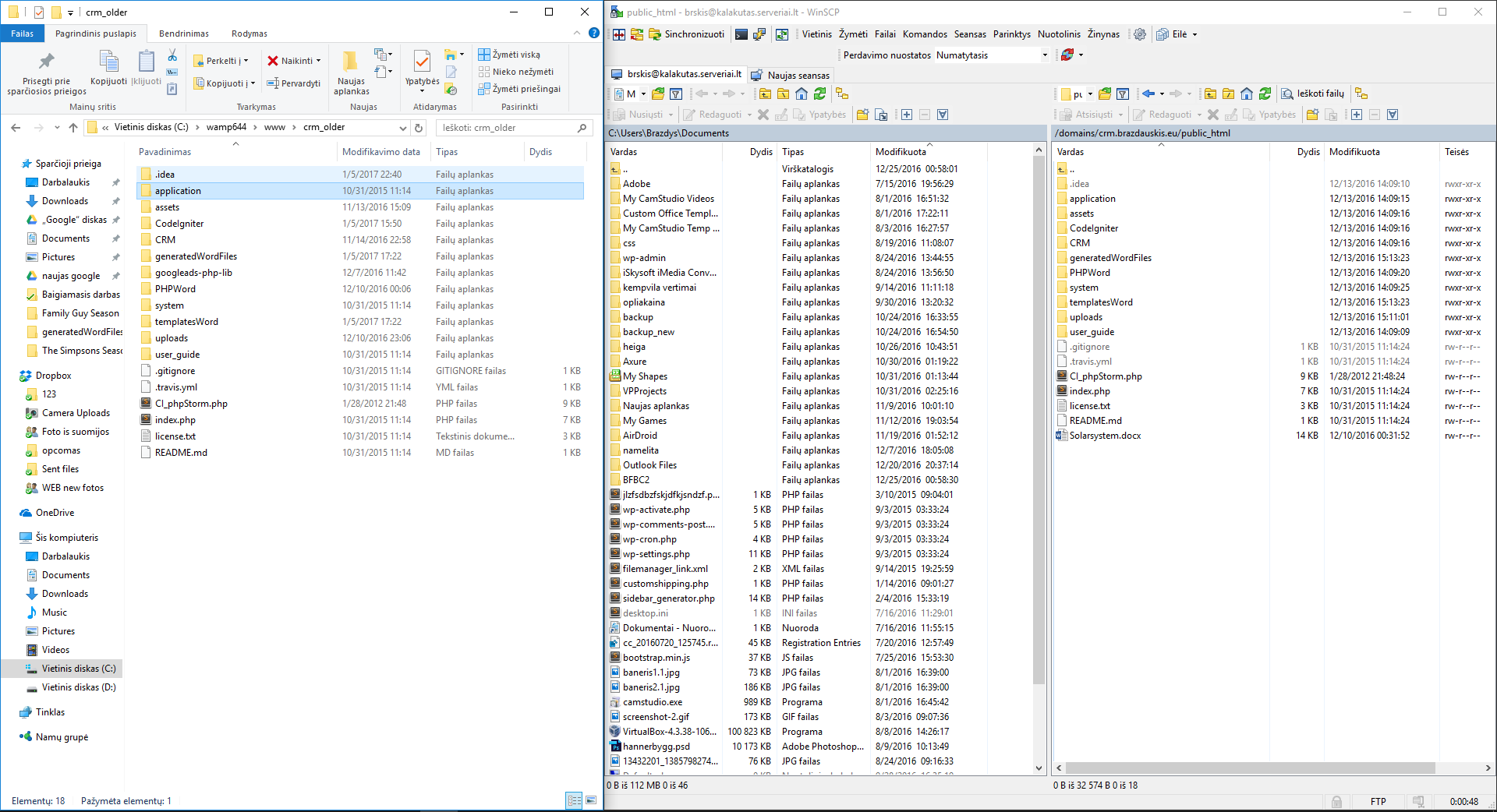
**2 priedas**

****

**3 priedas**

**Informacijos sistemos diegimo instrukcija**

1. Sukeliame visus failus į FTP serverį



1. Tada nueiname į failą \application\config\database.php

$db['default']['username'] = 'root';

$db['default']['password'] = '';

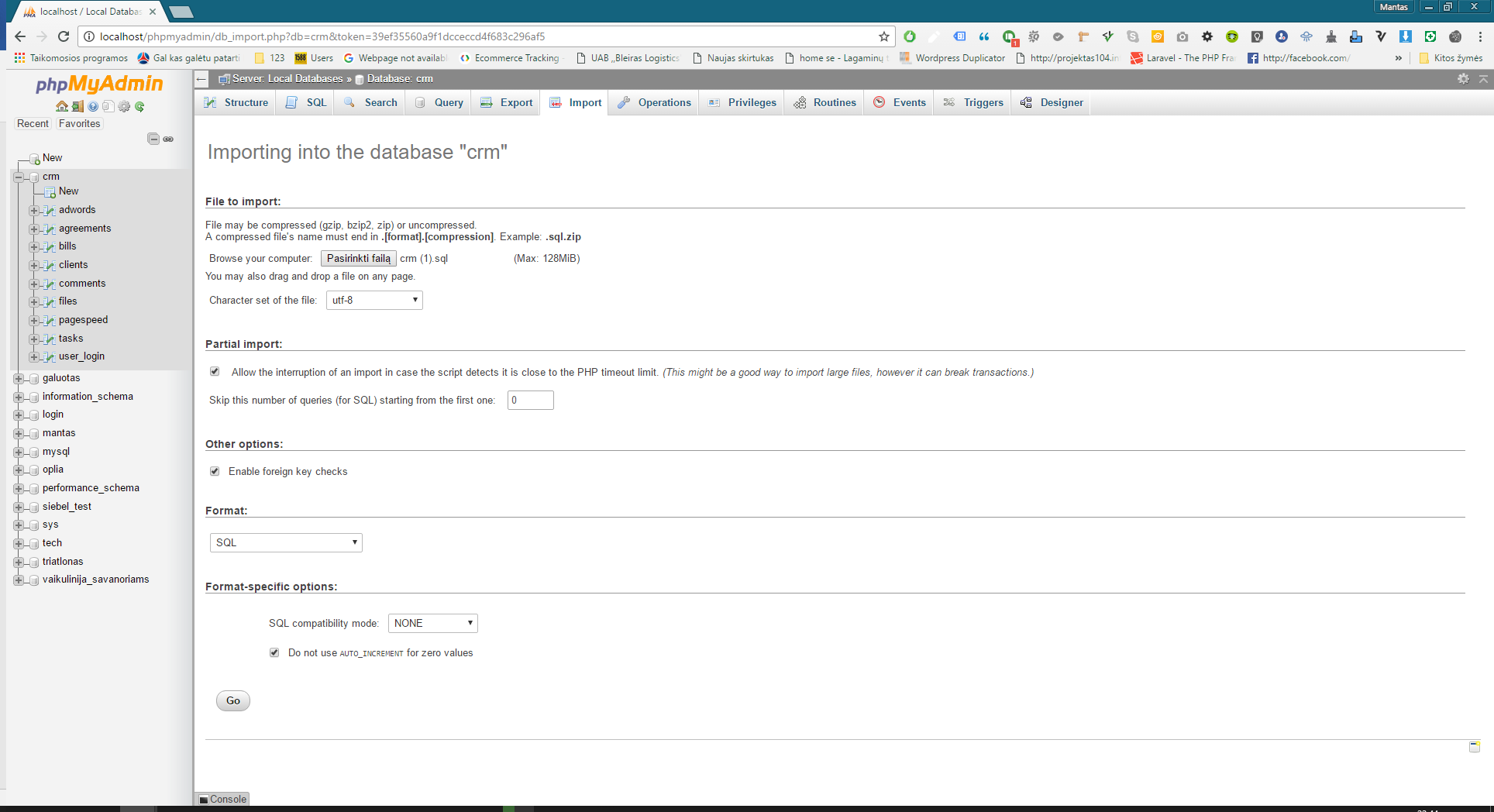
Pakeičiame šiuos prisijungimus į savo duomenų bazės prisijungimus

1. Taip pat config.php faile keičiame:

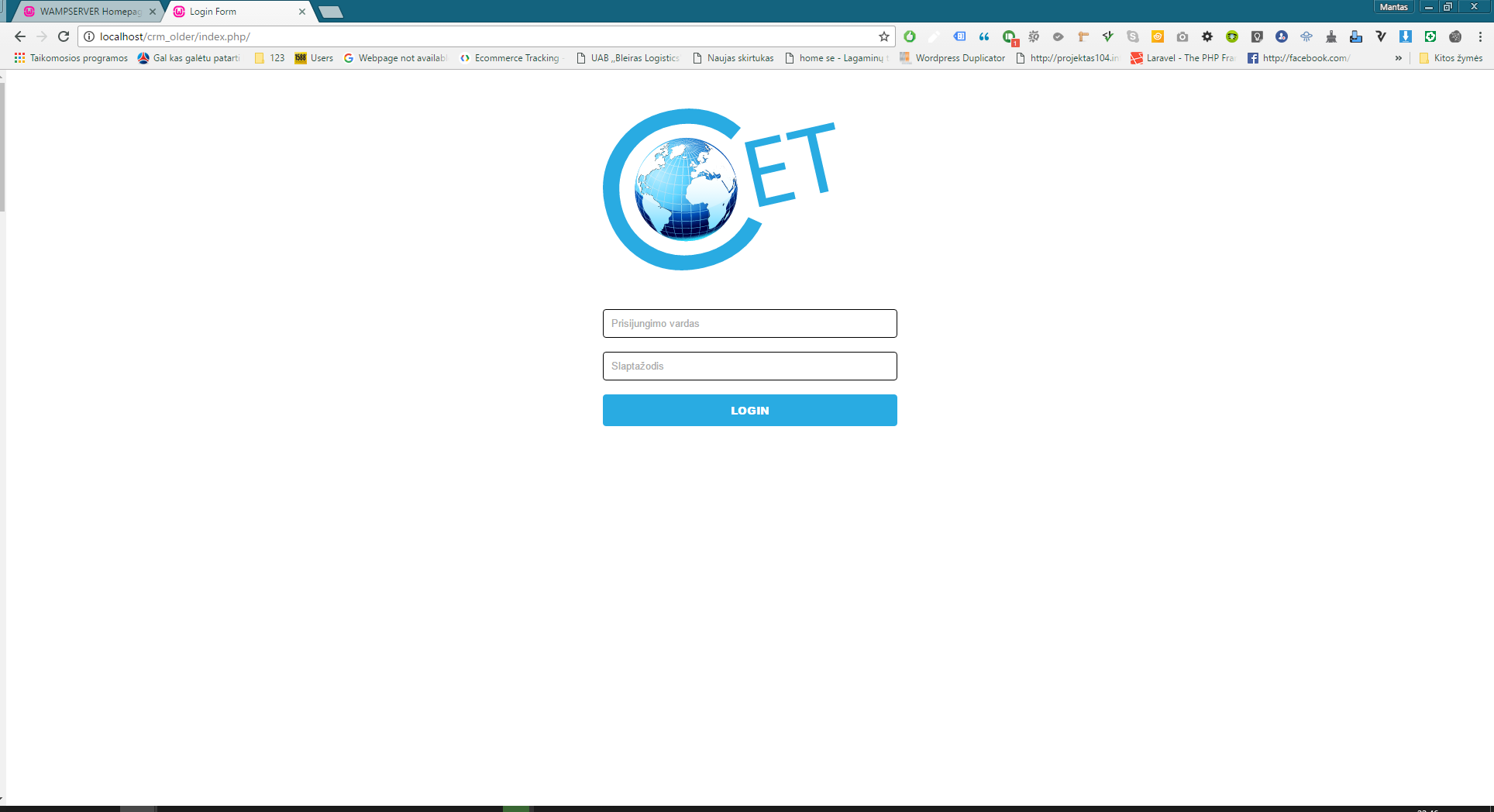
$config['base\_url'] = 'http://localhost/crm\_older/';

Pavyzdžiui šiuo atveju: $config['base\_url'] = 'http://crm.brazdauskis.eu/ ';

1. Suimportuojame duomenų bazę naudodami phpmyadmin panelę



1. Viskas galime paleisti CRM



**3 priedas. Naudotojo instrukcija**

1. **Klientų modulis**

Prisijungus prie sistemos mes matome štai tokį langą: