**Profesionālās izglītības kompetences centrs**

**„RĪGAS VALSTS TEHNIKUMS”**

DATORIKAS NODAĻA

Izglītības programma: “Programmēšanā”

**Kvalifikācijas darbs**

Laika pārvaldības sistēmā

Paskaidrojošais raksts lpp

Audzēknis: D. Ponomarenko

Vadītājs J.Matvejeva

Normu kontrole O.Sabanska

**Rīga**

**2019./2020. m.g.**

**ANOTĀCIJĀ**

Izveidojamās sistēmas izstrādes process ir aprakstīts dokumentācijā. Sistēma ļauj SimBASE lietotājiem izmantot lietotāja interfeisu kā iztērētā darba laika ierakstu Tika izmantots Bootstrap, PHP programmēšanas valoda un SimBASE ārējā sistēma – SIA “Simourg” produkts.

Kvalifikācijas darba paskaidrojuma raksts ietver sevī ievadi, darba definīciju, prasību specifikāciju, izvēlēto tehnoloģiju aprakstu, programmatūras izstrādei, projektēšanai un modelēšanai, datu struktūras aprakstu, atzinumu un pielikumus. Kvalifikācijas darba ieviešanā aprakstīta aktuāla veiktā darba ierakstu uzskaites problēma. Prasību specifikācijā ietilpst funkcionālās un nefunkcionālās prasības, kā arī ieejas, izvades un ārējās informācijas apraksts. Darba definīcijā norādīti uzdevumi, kurus sistēma veiks. Attiecībā uz izvēlētajām tehnoloģijām izstrādei, uzskaitīts, kādi programmēšanas tehnologi bija nepieciešami realizācijai un kādiem mērķiem tie tika izmantoti. Programmatūras modelēšanas un projektēšanas daļā, kā arī sistēmas arhitektūra ar attēliem, kā arī ER diagramma ar relāciju aprakstu. Lietotāja rokasgrāmatā norādītas sistēmas minimālās sistēmas prasības, sistēmas instalācija un palaišana, kā arī programmas apraksts, kurā uzskatāmi tiek skaidrots sistēmas funkcionālis un tas, kā to pareizi lietot. Atzinumā aprakstīts, kas tika realizēts, kā arī nākotnes programmas attīstības plāns. Testa piemērā apraksta gadījumus, kad sistēma nosūta lietotājam kļūdas tekstu ar vizuāliem attēliem.

Kvalifikācijas darbs sastāv no 53 lapaspuse, kura ietilpst 30 attēli, 1 tabulu un 10 pielikumi. Pielikumi sevi ietver ER diagrammu, funkcionālas dekompozīcijas diagrammu, datu plūsmu diagrammas, programmas attēlus un programmas pirmkodu.

**АНОТАЦИЯ**

Процесс разработки создаваемой системы описан в документации. Система предоставляет пользователям системы SimBASE использовать пользовательский интерфейс в качестве записи потраченного рабочего времени на ту или иную задачу.

Был использован фреймворк Bootstrap, язык программирования PHP и внешняя система SimBASE – продукт компании Simourg.

Пояснительная записка квалификационной работы содержит в себе введение, постановку задания, спецификацию требований, описание выбранных технологий для разработки, проектирование и моделирование программного обеспечения, описание структуры данных, заключение и приложения. Во введении квалификационной работы описана актуальная проблема учёта записей выполненной работы. В спецификацию требований входит функциональные и нефункциональные требования, а также описание входной, выходной и внешней информации. В постановке задания указаны задачи, которые системе будет необходимо выполнять. В описании выбранных технологий для разработки, перечислено, какие технологи программирования были необходимы для реализации и для каких целей они были использованы. В части моделирования и проектирования программного обеспечения, дана архитектура системы с изображениями, а также ER диаграмма с описанием связей. В руководстве пользователя указаны минимальные системные требования аппаратуры, инсталляция системы и ее запуск, а также описание программы, в котором наглядно объясняется функционал системы и то, как правильно им пользоваться. В заключении описано, что было реализовано, а также будущий план развития программы. В тестовом примере описывается случаи, когда система отправляет пользователю ошибки в виде текста с наглядными изображениями.

Описание квалификационной работы состоит из 53 страниц, в которые входят 30 изображений, 1 таблица и 10 приложений. Приложения содержит ER диаграмму, диаграммы потоков данных и исходный код программы.

**SATURS**

[**IEVADS** 5](#_Toc41817389)

[**1.** **UZDEVUMA NOSTĀDNĒ** 6](#_Toc41817390)

[**2.** **PRASĪBU SPECIFIKĀCIJA** 7](#_Toc41817391)

[**2.1.** **Ieejas, izejas un arējas informācijas apraksts** 7](#_Toc41817392)

[***2.1.1.*** ***Iejās informācijas apraksts*** 7](#_Toc41817393)

[***2.1.2.*** ***Izejas informācijas apraksts*** 7](#_Toc41817394)

[***2.1.3.*** ***Arējas informācijas apraksts*** 8](#_Toc41817395)

[**2.2.** **Funkcionālas prasības** 10](#_Toc41817396)

[**2.3.** **Nefunkcionālas prasības** 13](#_Toc41817397)

[**3.** **RISINĀŠANAS LĪDZEKĻU IZVELĒ** 15](#_Toc41817398)

[**3.1.** **Koda redaktors “Visual Studio Code”** 15](#_Toc41817399)

[**3.2.** **Versijas kontroles sistēmā “GitHub”** 15](#_Toc41817400)

[**3.3.** **Aizmugursistēmas valoda “PHP”** 15](#_Toc41817401)

[**3.4.** **Stila ietvars “Bootstrap”** 15](#_Toc41817402)

[**3.5.** **BPM sistēmā “SimBASE”** 16](#_Toc41817403)

[**3.6.** **Modelēšanas un biznesa procesu izveides platforma “Camunda”** 16](#_Toc41817404)

[**4.** **PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA** 17](#_Toc41817405)

[**4.1.** **Sistēmas struktūras modelis** 17](#_Toc41817406)

[***4.1.1.*** ***Sistēmas arhitektūra*** 17](#_Toc41817407)

[***4.1.2.*** ***Sistēmas ER modelis*** 17](#_Toc41817408)

[**4.2.** **Funkcionālais sistēmas modelis** 20](#_Toc41817409)

[***4.2.1.*** ***Datu plūsmu modelis*** 20](#_Toc41817410)

[**5.** **LIETOTĀJA CEĻVEDIS** 22](#_Toc41817411)

[**5.1.** **Sistēmas prasības aparatūrai un programmatūrai** 22](#_Toc41817412)

[**5.2.** **Sistēmas instalācija** 22](#_Toc41817413)

[**5.3.** **Programmas apraksts** 23](#_Toc41817414)

[**5.4.** **Testa piemērs** 35](#_Toc41817415)

[**NOBEIGUMS** 41](#_Toc41817416)

[**INFORMĀCIJAS AVOTI** 42](#_Toc41817417)

[**PIELIKUMI** 43](#_Toc41817418)

[1. pielikums 44](#_Toc41817419)

[2. pielikums 45](#_Toc41817420)

[3. pielikums 46](#_Toc41817421)

[4. pielikums 53](#_Toc41817422)

**IEVADS**

Šodien galvenais resursu pārvaldības izaicinājums ir tas, ka vadītājs ne vienmēr saprot, par ko viņš maksā naudu saviem darbiniekiem? Kādu darbu veic darbinieks mēneša laikā? un kādā apjomā?

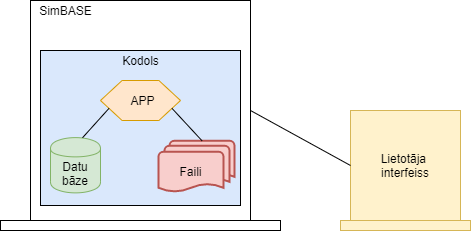
Lielisks risinājums savu darbinieku kontrolei ir gatavā SimBASE sistēma. Tā ļauj iestatīt sistēmu ar jebkuru funkcionalas grutību, piemēram, aprēķināt darbinieka algu pēc izpildītā darba. Bet jautājums ir tads, par kādu kritēriju ir aprēķināt darbinieka laiku? Par katru izpildīto darbu un tas patereto laiku, darbinieks var pārskatīt sistēmas iekšienē, ierakstot paveikto darbu komentāru, norādot laiku, par kuru viņš ir izpildījis šo uzdevumu un tipu, ar kuru būs vieglāk saprast, kādā nozarē darbinieks veic darbus.

Sistēma palīdz lietotājam redzēt savus pašreizējos uzdevumus, parādot datumu, kad uzdevums ir jānokārto, uzdevuma statuss, piemēram, nokavēts vai tas ir izpildes statusā. To varēs izmantot arī standarta interfeisā, ko sniedz pati SimBASE sistēma, gan speciālajā. Kuras funkcijās ietilps tikai atskaite par paveikto darbu. Viss pārējais funkcionālis atradīsies SimBASE sistēmā, šeit to varēs papildināt. Piemēram, izveidot papildu procesus resursu sadalei. Pievienot papildu amatus, nodaļas. Nolīgt vairāk darbinieku.

Lai izmantotu komentāru rakstīšanas mehānismu, ir ērtāk izmantot mobilās ierīces, labāk izmantot jaunu lietotāja interfeisu. Izmantojot šo interfeisu, lietotājs varētu atskaitīties, neejot uz SimBASE, kas tiek atbalstīts tikai lielajos ekrānos. Lietotājs var atrasties jebkur, strādāt mājās un pastāvīgi izmantot šo sistēmu, izmantojot īpašo tīmekļa lietojumprogrammu.

1. **UZDEVUMA NOSTĀDNĒ**

Šis kvalifikacijas darba galvenais mērķis ir izvdeidot laikas pārvaldības sistēmu ar kuru palīdzībū SimBASE lietotājiem, būt iespēja ērti reģistrēt savā darbā pavadītais laiku. Sistēma sastāv no divām daļām, galvenais būs SimBASE sistema, tas ir kodols šajā kvalifikācijas darbā, un otrs - lietotāja interfeiss, kas būs saistīts ar pamatsistēmu, izmantojot API ar nākamo arhitektūru.



1.1.att. Sistēmas arhitektūra

Laika pārvaldības sistēma tiks nodrošinātas sekojošas darbības:

* Projektus attēlošana un ar tiem saistītās aktivitātes;
* Tehniskos atbalstus attēlošana, kuros piedalās Simourg darbinieki.
* Komentārus pievienošana par veikto darbu;
* Jaunu uzdevumu pievienošana;
* Ātra sazināšanās iespēja ar citiem sistēmas lietotājiem.

1. **PRASĪBU SPECIFIKĀCIJA**
   1. **Ieejas, izejas un arējas informācijas apraksts**
      1. ***Iejās informācijas apraksts***

Sistēma tika nodrošināta šādu ieejas informācijas apstrāde:

1. Informācijā par **lietotājiem** sastāvēs no sekojošiem datiem:

* Lietotājvārds – burtu teksts ar izmēru līdz 10 rakstzīmēm.
* Parole – burtu teksts ar izmēru līdz 20 rakstzīmēm.

1. Informācijā par **komentāriem** sastāvēs no sekojošiem datiem:

* Teksts – burtu teksts ar izmēru līdz 1000 rakstzīmēm.
* Laiks – ciparu teksts ar izmēru līdz 4 rakstzīmēm.
* Komentaru tips – piedāvājumā izvelē, tādi kā ‘Testing’, ‘Support’, ‘Programming’, ‘Project mgmt’, ‘Implementation’, ‘Design’, ‘Documenting’, ‘Bugfix’, ‘Pre-project activities’.
* Datums – šodienas datums - dd.mm.yyyy formāta.

1. Informācijā par **jaunu uzdevumu** sastāvēs no sekojošiem datiem:

* Tēma - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Apraksts - burtu teksts ar izmēru līdz 500 rakstzīmēm.
* Sākuma datums - dd.mm.yyyy formāta.
* Beigu datums - dd.mm.yyyy formāta.
* Izpildītājs - piedāvājumā izvelē visas sistemas lietotajus.
* Kontrolieris - piedāvājumā izvelē visas sistemas lietotajus.
  + 1. ***Izejas informācijas apraksts***

1. Sistēmas ieeja atspoguļojums. Ja lietotājs ir sekmīgi pierakstījies, sistēma parādīs projektu sarakstu.
2. Rādīt neveiksmīgu pieteikšanos. Ja SimBASE nevarēja atrast lietotāja pēc lietotājvārdu un paroli, programma atgrieza kļūdas tekstu - “Lietotājs nav atrasts” krievu valodā.
3. Rādīt ierakstīto komentāru. Pēc tam, kad lietotājs ir ierakstījis komentāru par veikto darbu savā uzdevumā, tiek parādīts pēdējais komentārs komentāru tabulā.
4. Jauna uzdevuma validācija. Programma atgriež kļūdas tekstu, ja vismaz viens lauks nav aizpildīts. Piemēram, ja lietotājs nav norādījis sākuma datumu, programma atgriež nākamo kļūdas tekstu “Ir jānorāda sākuma datums” krievu valodā utt.

* Tēma – “Norādiet uzdevuma tēmu”;
* Apraksts - “Norādiet uzdevuma aprakstu”;
* Beigu datums — “Norādiet beigu datumu”;
* Izpildītājs – “Izvēlieties izpildītāju”;
* Kontrolieris - “Izvēlieties kontrolieri”;

1. Ja sākuma datums būs vēlāks par beigu datumu, programma atgriezs teksta kļūdu “Sākuma datums nevar būt vēlāks par beigu datumu”.
2. Pēc statusa maiņas mainās uzdevuma indikatora krāsa.
   * 1. ***Arējas informācijas apraksts***

Sistēma tika nodrošināta šādu ieejas informācijas apstrāde:

1. Informācijā par **projektiem** sastāvēs no sekojošiem datiem:

* Stāvoklis - burtu teksts ar izmēru līdz 40 rakstzīmēm.
* Nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Sākuma datums – sekundes formātā.
* Pabeigšanas datums – sekundes formātā.
* Nodošanas datums – sekundes formātā.
* Nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Vadītājs - vesels skaitlis. Lietotāja ID numurs.
* Izpildītājs - vesels skaitlis. Lietotāja ID numurs.
* Apraksts - burtu teksts ar izmēru līdz 500 rakstzīmēm.

1. Informācijā par **pasākumiem** sastāvēs no sekojošiem datiem:

* Stāvoklis - burtu teksts ar izmēru līdz 40 rakstzīmēm.
* Nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Projekts - vesels skaitlis. Projekta identifikācijas numurs.
* Vadītājs - vesels skaitlis. Lietotāja ID numurs.
* Sākuma datums – “unixtime” formātā.
* Pabeigšanas datums – “unixtime” formātā.
* Plānotais laiks - vesels skaitlis.
* Progresa indikators – vesels skaitlis no 1 līdz 100.

1. Informācijā par **tehnisku atbalstu** sastāvēs no sekojošiem datiem:

* Stāvoklis - burtu teksts ar izmēru līdz 40 rakstzīmēm.
* Nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Klients - burtu teksts ar izmēru līdz 30 rakstzīmēm.
* Apraksts - burtu teksts ar izmēru līdz 500 rakstzīmēm.

1. Informācijā par **uzdevumiem** sastāvēs no sekojošiem datiem:

* Stāvoklis - burtu teksts ar izmēru līdz 40 rakstzīmēm.
* Reģistrators - vesels skaitlis. Lietotāja ID numurs.
* Kontrolieris - vesels skaitlis. Lietotāja ID numurs.
* Tēma - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Izpildītājs - vesels skaitlis. Lietotāja ID numurs.
* Sākuma datums – “unixtime” formātā.
* Brīdinājuma datums - “unixtime” formātā.
* Pabeigšanas datums – “unixtime” formātā.
* Apraksts - burtu teksts ar izmēru līdz 500 rakstzīmēm.
* Progresa indikators – vesels skaitlis no 1 līdz 100.
* Uzdevuma laiks – hh:ss formāta.

1. Informācijā par **lietotājiem** sastāvēs no sekojošiem datiem:

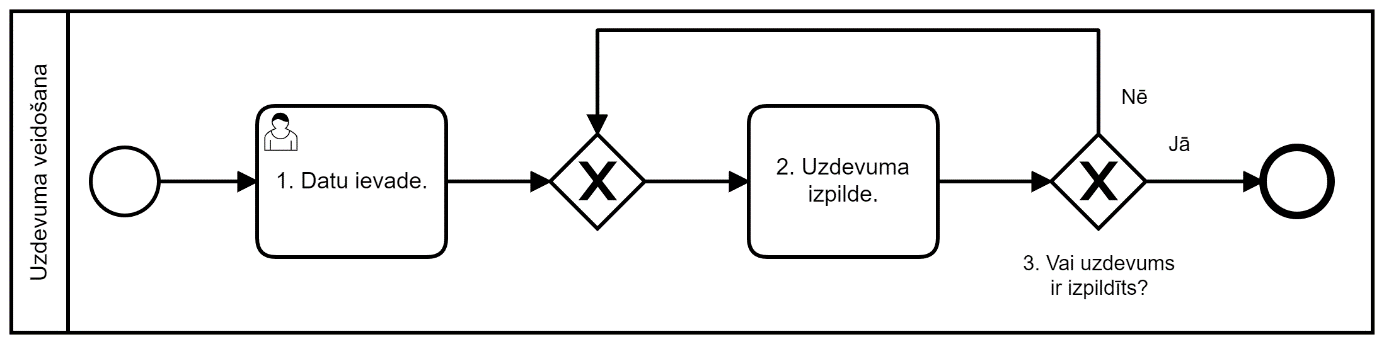
* Vārds - burtu teksts ar izmēru līdz 40 rakstzīmēm.
* Uzvārds - burtu teksts ar izmēru līdz 40 rakstzīmēm.
* Dzimums – 1 – vīrietis, 2 - sieviete.
* Amats - vesels skaitlis. Amata ID numurs.
* Nodaļa - vesels skaitlis. Nodaļa ID numurs.
* Dzimšanas datums – “idate” formātā.
* Tips – lietotāja tipa identifikācijas numurs.
* Valsts - valsts identifikācijas numurs.
* Pilsēta - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Adrese - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* E-pasta adrese - burtu teksts ar izmēru līdz 100 rakstzīmēm.
* Telefons - burtu teksts ar izmēru līdz 20 rakstzīmēm.
* Numurs SMS sūtīšanai - burtu teksts ar izmēru līdz 20 rakstzīmēm.
* Lietotāja vietnieks - vesels skaitlis. Lietotāja ID numurs.

1. Informācijā par **amatiem** sastāvēs no sekojošiem datiem:

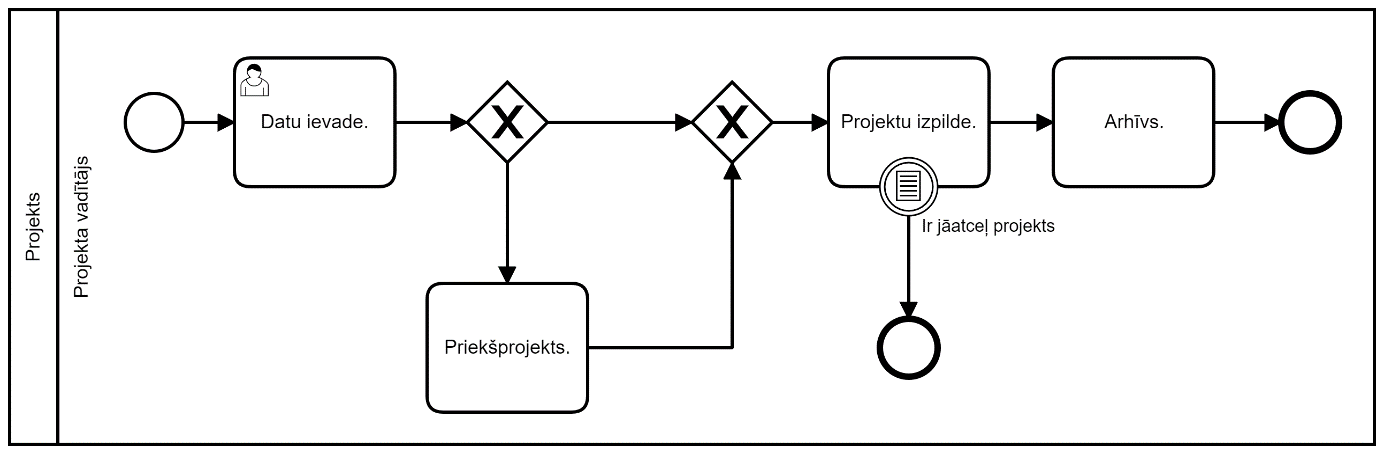
* Nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Apraksts - burtu teksts ar izmēru līdz 500 rakstzīmēm.

1. Informācijā par **nodaļām** sastāvēs no sekojošiem datiem:

* Nosaukums - burtu teksts ar izmēru līdz 50 rakstzīmēm.
* Apraksts - burtu teksts ar izmēru līdz 500 rakstzīmēm.
  1. **Funkcionālas prasības**

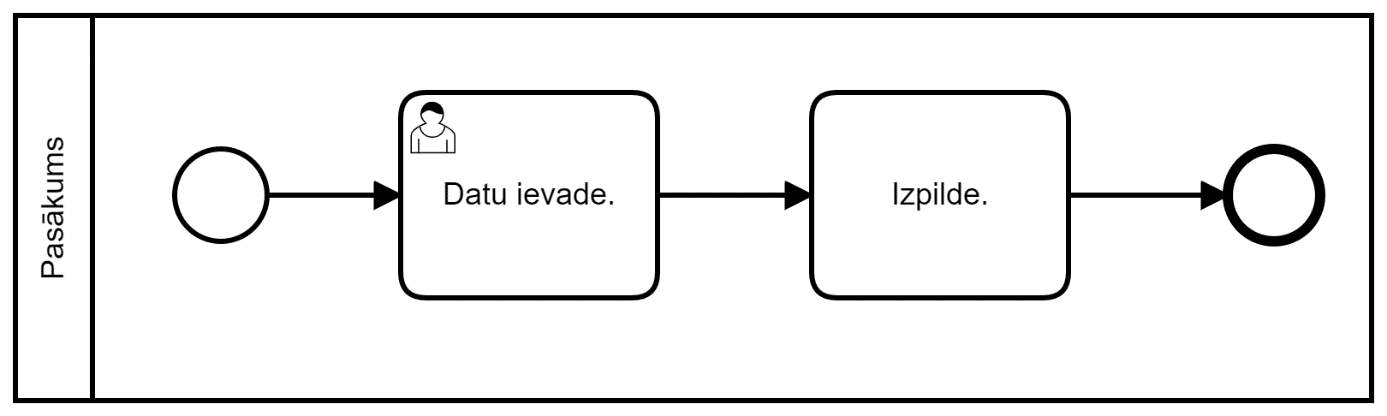
1. Iestatīt SimBASE sistēmas procesus.
   1. Sagatavot procesu “Uzdevums” SimBASE sistēmā, kā norādīts shēmā (skat. 1.1.1. att.).

1.1.1.att. Uzdevumu procesa diagramma

* + 1. Lai ievadīt datus, ir jāiestata tālāk norādītie lauki:
* Reģistrētājs - lietotājs, kurš izveidoja uzdevumu;
* Kontrolieris - lietotājs, kurš atbild par veiktā darba kvalitāti;
* Darba izpildes sākuma datums;
* Brīdinājuma datums;
* Darba izpildes beigu datums;
* Uzdevuma apraksts;
* Uzdevuma izpildes progresa indikators;
* Nepieciešamais laiks uzdevuma veikšanai.
  + 1. Kontroles stāvoklī kontrolierim ir jāizvēlas, jānosūta uzdevumu uz pabeigšanu vai jāapstiprina uzdevuma izpilde.
  1. Sagatavot procesu “Projekts” SimBASE sistēmā, kā norādīts shēmā (skat. 1.2.1. att.).

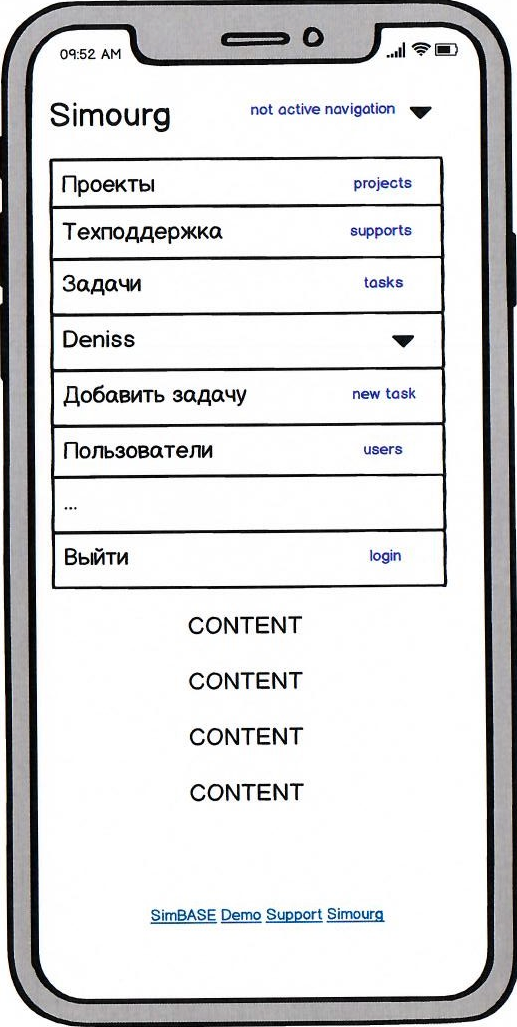
1.2.1.att. Projektu procesa diagramma

1.2.1.att. Projektu procesa diagramma

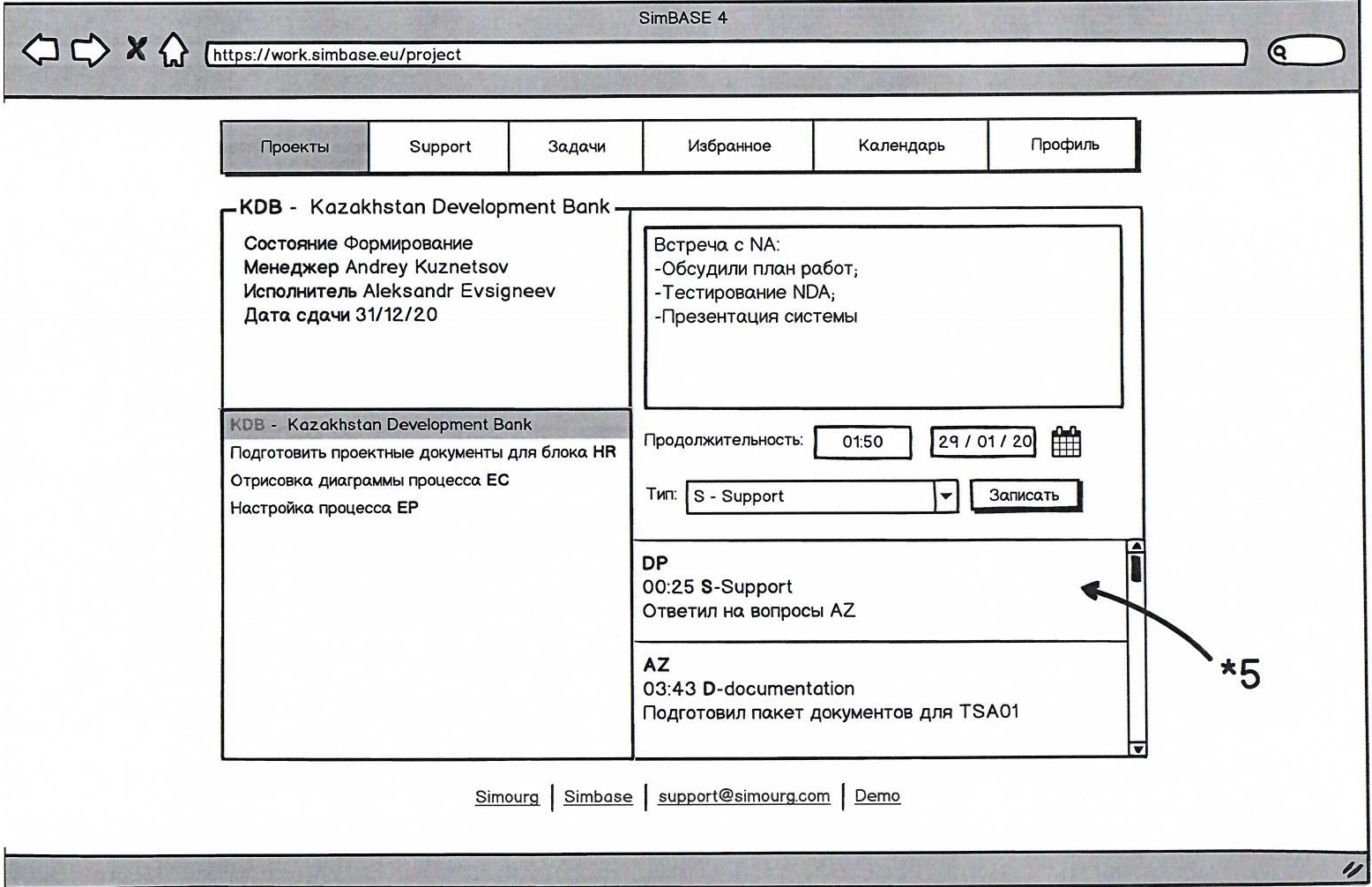
* + 1. Lai ievadīt datus, ir jāiestata tālāk norādītie lauki:
* Projekta nosaukums;
* Sākuma datums;
* Beigu datums;
* Nodošanas datums;
* Vadītājs - lietotājs, kurš atbild par projekta nodošanu;
* Izpildītājs - lietotājs, kurš atbild par projekta izpildīšānu;
* Apraksts.
  + 1. Iestatīt atsauci uz pasakumiem, kas attiecas uz konkretu projektu.
  1. Sagatavot procesu “Pasākums” SimBASE sistēmā, kā norādīts shēmā (skat. 1.3.1. att.).

1.3.1.att. Pasakuma procesa diagramma

* + 1. Lai ievadīt datus, ir jāiestata tālāk norādītie lauki:
* Pasākuma nosaukums;
* Projekta atsauce;
* Vadītājs - lietotājs, kurš atbild par pasakumu rezultatu;
* Sākuma datums;
* Beigu datums;
* Nodošanas datums;
* Plānotais laiks;
* Progresa indikators.

1. Jānodrošina lietotāja autorizācija.
   * 1. Jā lietotāja statuss ir aktīvs, sistēmai ir jānodrošina autorizācija, ievadot lietotājvardu un paroli.
     2. Pēc lietotāja autorizācijas, saglabāt viņa lietotajvārdu un paroli.
     3. Ja lietotāja statuss ir neaktīvs, tad sistēmai ir jāliedz autorizācija.
     4. Ja kāds lauks nav aizpildīts vai litotājvards un parole nesakrīt, sistēmai ir jāizvada paziņojums un saglabat lietotajvardu atbilstošaja lauka.
     5. Pēc lietotāja autorizācijas parādīt projektu lapu.
2. Jānodrošina komentāru ierakstīšanu.
   * 1. Lietotajam jābūt autorizētam sistēmā, lai pievienot komentaru.
     2. Rādīt komentāru pievienošanas formu tikai speciālajās lapās, piemēram, projektos, tehniskajā atbalstā un uzdevumos.
     3. Komentāra ierakstīšanai, nosūtīt uz SimBASE sistemu atlasītā komentāra tipa identifikācijas numurs
     4. Pievienot komentāru var tikai tad, kad obligātie ievadlauki ir aizpildīti. Ja kāds lauks nav aizpildīts, sistēmai ir jāizvada paziņojumu.
     5. Rādīt komentāru tipu kā nosaukumu.
3. Jānodrošina datus attelošānu.
   * 1. Rādīt komentāra tipu nosaukumu, nevis identifikācijas numuru.
     2. Rādīt nodaļas nosaukumu, nevis identifikācijas numuru.
     3. Rādīt amatas nosaukumu, nevis identifikācijas numuru.
     4. Rādīt lietotaja vārdu un uzvārdu, nevis identifikācijas numuru.
     5. Rādīt datumus dd.mm.yyyy, nevis nevis sekunžu vai dienu skaita formātā.
4. Jānodrošina uzdevumu pievienošanu.
   * 1. Lietotajam jābūt autorizētam sistēmā, lai pievienot uzdevumu.
     2. Lietotajam tiek pieprasītā datu ievade.
        1. Lietotajam, tiek pieprasīts ievadīt nosaukumu līdz 50 simboliem, gadījumā ja tas tiks atstats tukšs, tad pievienošanas poga netiks aktivizēta un sistema nosūtīs īsziņu par kļūdu.
        2. Lietotajam, tiek pieprasīts ievadīt uzdevuma aprakstu līdz 500 simboliem, gadījumā ja tas tiks atstats tukšs, tad pievienošanas poga netiks aktivizēta un sistema nosūtīs īsziņu par kļūdu.
        3. Lietotajam, tiek pieprasīts ievadīt sakuma datumu ar formatu dd.mm.yyyy, gadījumā ja tas tiks atstats tukšs, tad pievienošanas poga netiks aktivizēta un sistema nosūtīs īsziņu par kļūdu.
        4. Lietotajam, tiek pieprasīts ievadīt beigu datumu ar formatu dd.mm.yyyy, gadījumā ja tas tiks atstats tukšs, tad pievienošanas poga netiks aktivizēta un sistema nosūtīs īsziņu par kļūdu.
        5. Lietotajam, tiek pieprasīts izvelet izpilditaju, gadījumā ja tas tiks atstats tukšs, tad pievienošanas poga netiks aktivizēta un sistema nosūtīs īsziņu par kļūdu.
        6. Lietotajam, tiek pieprasīts izvelet kontrolieris, gadījumā ja tas tiks atstats tukšs, tad pievienošanas poga netiks aktivizēta un sistema nosūtīs īsziņu par kļūdu.
5. Izmantojot lietojumprogrammu, lietotājs var rakstīt vestules uz e-pastu, noklikšķinot uz e-pasta adresi izmantojot, piemēram, outlook vai piezvanīt, noklikšķinot uz mobilā tālruņa numura.
6. Jānodrošina lietotajas informacijas attelošanu.
   1. Rāda kontaktinformāciju, kas tiek glabāta SimBASE lietotāja kartītē.
   2. Rādīt lietotāja attēlu, kas atrodas darbinieka Personiskā lieta (process).
   3. **Nefunkcionālas prasības**
7. Sistēmas saskarnes valodai ir jābūt krievu valodai.
8. Jānodrošina tīmekļa lietojumprogrammas pielāgošanos jebkuram ekrāna izmēram, lai to varētu lietot datorā, gan planšetdatorā vai viedtālrunī.
9. Dizainā jābūt uzņēmuma SIA “Simourg” krāsām, tas iz zila un balta krasa, tekstam jābūt melnā krāsā ar salasāmu fontu.
10. Datu apstrādes pogām ir jābūt lielām.
11. Jānodrošina mūsdienīgu un patīkamu dizainu.
12. Lauku, kuri tika nepareizi aizpildīti – rādīt kļūdas tekstu sarkanā krāsā, kā brīdinājums.
13. Galvenā izvēlnē var saturēt apakšsadaļas.

7.1.att. Pasakuma procesa diagramma

1. Interfeisu ir jāsadala uz trim daļām: izvēlnes daļu, sistēmas daļu, kur tiek veiktas visas darbības un uzņēmuma kontaktinformācija.

8.1.att. Lietotāja interfeiss

1. **RISINĀŠANAS LĪDZEKĻU IZVELĒ**
   1. **Koda redaktors “Visual Studio Code”**

Šis Microsoft produkts ir ieteicis sevi visā pasaulē, jo ir ļoti ērta un saprotama saskarne. Tai ir daudz iebūvētu moduļu, kas palīdz efektīvāk un ātri strādāt ar programmas kodu. Piemēram, darbs vienlaikus ar visu failu struktūru, kas, piemēram, nav notepad + +. Atkarībā no programmēšanas valodas ir lielisks koda spīdums, kas ļauj viegli atrast apzīmogojumu vai kļūdu. Visual Studio Code teksta redaktors ir iebūvēts komandrindā, kas ļauj programmu izstrādātājiem ērti un ātri manipulēt un konfigurēt savu programmatūru.

Arī viens no šī redaktora lielajiem plusiem ir viņa pieejamība, jo studentiem tas tiek piešķirts pilnīgi bez maksas.

Tiek izmantota versija: 2020 gada janvaris 1.42 versija

* 1. **Versijas kontroles sistēmā “GitHub”**

Tiek izmantota lai veikts sistēmas versiju kontroli. Tiek izmantota versija: GitHub version 2.20.8.

* 1. **Aizmugursistēmas valoda “PHP”**

Tika izvelētā šī aizmugursistēmā valoda, lai izveidot sistēmu, kura varēs nodrošināt labu komunikāciju ar datubāzi un priekšgalsistēmu.

Ir populāra vispārējas nozīmes skriptu valoda, kas ir īpaši piemērota tīmekļa izstrādei. Ātrs, elastīgs un praktisks PHP nodrošina visu, sākot no jūsu emuāra un beidzot ar populārākajām vietnēm pasaulē. Universāla skriptu valoda, ko intensīvi izmanto tīmekļa lietojumprogrammu izstrādei. Pašlaik to atbalsta vairums mitināšanas pakalpojumu sniedzēju, un tas ir viens no līderiem starp valodām, kuras izmanto dinamisku vietņu izveidošanai.

* 1. **Stila ietvars “Bootstrap”**

Tika izvelētā, kā populārs un stabils stila ietvars lai saīsināt laiku ar stila darbiem, ka ari izveidot labu skatu priekš sistēmas un patīkamiem modāliem logiem priekš un ērtas formas aizpildīšanas, ka ari intuitīvā interfeisa.

Bezmaksas rīku komplekts vietņu un tīmekļa lietojumprogrammu izveidošanai. Ietver HTML un CSS dizaina veidnes tipogrāfijai, tīmekļa veidlapām, pogām, etiķetēm, navigācijas blokiem un citiem tīmekļa interfeisa komponentiem, ieskaitot JavaScript paplašinājumus.

Tiek izmantota versija: 2020 gada marta 4.5 versija

* 1. **BPM sistēmā “SimBASE”**

SimBASE ir jaunas paaudzes biznesa procesu (BPM system) vadības sistēma, kas ļauj automatizēt jebkura veida uzņēmumu darbību no jebkuras nozares. Būtībā tas ir konstruktors, kas ļauj radīt sarežģītības procesu, sākot no personāla vadības, beidzot ar valsts banku matemātiku.

SimBASE ir šī projekta kodols. Visi procesi un matemātika ir tieši tajā, kas ļauj samazināt slodzi lietotāja interfeisā.

Sistēmai ir jau iebūvēta datu bāze, kas tiek dinamiski paplašināta, pielāgojot sistēmu. Tas nozīmē, ka visi dati tiek glabāti sistēmā un lietotāja interfeisa izstrādātājam nav jākonstruē datu bāze, kas ievērojami ietaupa laiku plānošanas un realizācijas posmā.

Tiek izmantota versija: 2020 gada majs 441A versija

* 1. **Modelēšanas un biznesa procesu izveides platforma “Camunda”**

Šī platforma tika izmantota, lai vizualizētu procesu loģiku, kas tika konfigurēta SimBASE BPM sistēmā.

Šī platforma tika izmantota, lai vizualizētu procesu loģiku, kas tika konfigurēta SimBASE BPM sistēmā.

Šī platforma tika veikta, balstoties uz Javascript programmēšanas valodu, kas ļauj darboties pietiekami ātri ar lielajām diagrammām, salīdzinot ar tās analogiem.

Camunda ir iebūvēts eksports trim attēlu veidiem, PNG, JPEG un SVG.

Tiek izmantota versija: 2020 gada majs 7.13.0 versija

1. **PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA**
   1. **Sistēmas struktūras modelis**
      1. ***Sistēmas arhitektūra***

Lietotajs, kuram ir piekļuve sistēmai, var izmantot gan standarta interfeisu, gan lietotāja interfeisu. Standarts interfeiss (SimBASE) saglabās daudz vairāk funkciju, tas ir atkarīgs no konfigurēto procesu skaita un kvalitātes. Kad lietotāja interfeiss ir atbildīgs par laika pārvaldību.

Šī sistēmā sastāv no divam apakšsistēmām: SimBASE pusē un lietotāja interfeiss. Funkcionālas dekompozīcijas diagrammu skatīt 1. pielikuma.

***SimBASE apakšsistēmā***

SimBASE apakšsitema sastāv no vairākiem moduļiem un visiem lietotajas funkcijām, kuras apstrāda datus.

Šajā apakšsistēmā tiek konfigurēti visi biznesa procesi. Sistēma atbild par datu apstrādi un visas nepieciešamās informācijas glabāšanu. Šeit var izveidot jaunus biznes procesus, pievienot lietotājus, amatus, vienības jeb nodaļas un lietotāju tipus.

***Lietotāja interfeiss***

Lietotāja interfeiss ir atbildīgs par datu attēlošanu jaunākā formātā. Šajā apakšsistēmā var pievienot jaunus uzdevumus, rakstīt komentārus.

Šajā apakšsistēmā tiek rādīts lietotāju saraksts, aktīvais darbs, projekts un tehniskais atbalsts. Šis ir pielāgoti biznesa procesu objekti SimBASE sistēmā.

* + 1. ***Sistēmas ER modelis***

ER modeļa diagrammu skatīt 2. pielikuma.

ER diagramma sastāv no zemāk minētajām entītijām

* **“Lietotājs”** – apraksta sistēmas lietotājus. Tas atribūti kopuma sevi ietver: lietotājvārds, vārds, uzvārds, dzimums, amats, nodaļa, dzimšanas datums, lietotāja tips, valsts, pilstēta, adrese, e-pasta adrese, telefons, aprakstu un vietnieku.
* **“Amats”** – apraksta lietotajā amatu. Tas atribūti kopuma sevi ietver: nosaukums un apraksts.
* **“Nodaļa**” – apraksta nodaļu kura pašlaik atrodas darbinieks. Tas atribūti kopuma sevi ietver: nosaukums un apraksts.
* **“Lietotāja tips”** – apraksta lietotajas tipu. Tas atribūti kopuma sevi ietver: nosaukums, apraksts, lietotāja klase un autentifikācijas iestatījumi.
* **“Loma”** – apraksta lietotajas lomu. Tas atribūti sevi kopuma ietver: nosaukums un apraksts.
* **“Valsts”** – apraksta valstu kurā pašlaik dzīvo lietotājs. Tas atribūti sevi nosaukums, karogs, divburtu un trīsburtu kods.
* **“Laika josla”** – apraksta kurā laika joslā atrodas valsts. Tas atribūti sevi nosaukums un standarta laiks.
* **“Piezīme”** – apraksta lietotāja komentāru. Tas atribūti sevi kopuma ietver: autora datums, datums, komentāra teksts un komentāra tips.
* **“Piezīmes tips”** – apraksta komentāra tipu. Tas atribūti sevi kopuma ietver: Tips jeb nosaukums, apraksts un kods.
* **“Objekts”** – apraksta procesa realizāciju. Tas atribūti sevi kopuma ietver: izveidotājs, izveides laiks, numurs.
* **“Stāvoklis”** – apraksta, kādā stāvoklī atrodas objekts. Tas atribūti sevi kopuma ietver: nosaukums un apraksts.
* **“Objektu grupa”** – apraksta, kādā stāvoklī atrodas objekts Tas atribūti sevi kopuma ietver: nosaukums apraksts un attels.
* **“Process”** – apraksta biznesa procesa loģiku. Tas atribūti sevi kopuma ietver: nosaukums, apraksts, attels un noklusējuma objektu grupa.
* **“Lauks”** – apraksta, kādā stāvoklī atrodas objekts Tas atribūti sevi kopuma ietver: nosaukums, apraksts un mainīgā nosaukums.

ER diagrammas relācijas parāda kā savstarpēji saistītas divas vai vairākas entītijas

* Starp lietotāju un lomu ir attieciba daudz pret daudziem, jo vienam lietotajam var būt daudz lomas un vienai lomai var būt daudz lietotajus.
* Starp lietotāju un lietotajās tipu ir attiecība viens pret daudziem, jo vienam lietotajām ir tikai viens tips, bet vienam tipam ir daudz lietitaji.
* Starp lietotāju un amatu ir attiecība viens pret daudziem, jo vienam lietotājam ir tikai viens amats, bet vienam amatam ir daudz lietotāju.
* Starp nodaļu un lietotajam ir attiecība viens pret daudziem, jo vienā nodaļā strādā daudzi darbinieki, bet vienam lietotājam ir tikai viena nodaļa.
* Starp valstu un lietotaju ir attiecība viens pret daudziem, jo vienā valstī dzīvo daudzi lietotaji, bet viens darbinieks dzīvo vienā valstī.
* Starp laika joslu un valstu ir attiecība viens pret daudziem, jo vienā laika joslā atrodas vairākas valstis, bet viena valsts atrodas vienā laika joslā (neskaitot Krieviju (11 joslas), ASV (5), Kanāda (4), Austrālija (3), Brazīlija (3), Indonēzija (3),Meksika (2) ).
* Starp lietotaju un piezimes ir attiecība viens pret daudziem, jo vienai piezīmei ir tikai viens autors, bet vienam autoram var būt daudz piezīmes.
* Starp piezīmes tipu un piezīmei ir attiecība viens pret daudziem, jo vienai piezīmei ir tikai viens tips, bet ar vienu tipu var būt daudz piezīmes.
* Starp objektu un piezimei ir attiecība viens pret daudziem, jo viens objekts var glabās sevi daudz piezīmes un vienai piezīmei ir tikai viens objekts.
* Starp lietotāju un objektu ir attiecība viens pret daudziem, jo vienam objektam ir tikai viens radītājs, bet viens lietotājs var izveidot daudz objektu.
* Starp objektas grupu un objektu ir attiecība viens pret daudziem, jo vienam objektam ir tikai viena objektu grupa, bet vienam objektu grupai ir daudz objektus.
* Starp objektu grupu in procesu ir attiecība viens pret daudziem, jo vienam procesam ir tikai viena objekta grupa pec noklusejuma, bet vienai objektu grupai ir daudz atļautu procesu.
* Starp procesu un lauku ir attiecība daudz pret daudziem, jo viens lauks var atrasties daudzos procesos un vienam procesam ir daudz lauku.
  1. **Funkcionālais sistēmas modelis**
     1. ***Datu plūsmu modelis***

1. Datu plūsmu diagramma par lietotāja autorizāciju, skatīt 3. pielikuma 1. attēlā, tiek attēlots, kādas darbītbas tiek veiktas, kad lietotājs mēģina izveidot savienojumu ar sistēmu, izmantojot lietotāja interfeisu. Vispirms lietotājs ievada pieteikšanās informāciju (lietotājvārds un parole) un nospiež pogu “pierakstīties”. Pēc tam serveris pārbauda, vai visi lauki ir aizpildīti, ja ir aizpildīti, serveris gatavo un nosūta XML pieprasījumu SimBASE sistēmā. Kad SimBASE sistēma ir saņēmusi pieprasījumu, notiek pārbaude, vai lietotājs ir sistēmā? Ja lietotāja nav, SimBASE atgriež tukšu atbildi un serveris nosūta lietotājam ziņojumu, ka šo lietotāju nevarēja atrast. Ja lietotājs ir atrasts, SimBASE sagatavo un nosūta atbildi, serveris to apstrādā un saglabā sesijas datus un lietotājs pāriet uz projektu lapu.
2. Datu plūsmu diagramma par noliktavu pievienošanu, skatīt 3. pielikuma 2. attēlā, tiek attēlots, kādas darbības un pārbaudes tiek veiktas, kad lietotājs nonācis uz projektu lapu. Serveris, izmantojot saglabātos lietotāja datus, sagatavo un sūta XML pieprasījumu SimBASE sistēmā. Ja pēc pieprasījuma apstrādes sistēma pārbauda, vai lietotājam ir piekļuve kādam projektam? Ja nav, nosūta tukšu atbildi, ja ir nosūtīts aktīvo projektu saraksts atpakaļ uz serveri, kur pēc atbildes apstrādes tiek ģenerēta projektu lapa.
3. Datu plūsmu diagramma par noliktavu pievienošanu, skatīt 3. pielikuma 2. attēlā, tiek attēlots, kādas darbības un pārbaudes tiek veiktas, kad lietotājam ir nepieciešams jauns uzdevums. Vispirms lietotājam ir jāievada dati, šis ir darba tēma, sākuma datums, beigu datums, izpildītājs un kontrolieris. Pēc tam lietotājs nospiež pogu. Noklikšķinot uz pogas, serveris pārbauda datu aizpildīšanas pareizību, ja dati nav pareizi, serveris nosūta lietotājam kļūdas tekstu. Ja visi dati ir pareizi, serveris nosūta SimBASE pieprasījumu, lai izveidotu jaunu darbu. Simbase sistēma, apstrādā datus un izveido jaunu darba objektu. Pēc tam serveris nosūta paziņojumu lietotājam par objekta veiksmīgu izveidi
4. Datu plūsmu diagramma par noliktavu pievienošanu, skatīt 3. pielikuma 2. attēlā, tiek attēlots, kādas darbības un pārbaudes tiek veiktas, kad sistēma ģenerē lapu, ņemot vērā lietotāja saglabātos datus un interfeisa statusu. Vispirms sistēma pārbauda, vai lietotāja dati ir saglabāti, tā ir parole un lietotājvārds. Tiek pārbaudīts arī stāvoklis, ja visi dati ir saglabāti, un serveris zīmē lapu, izmantojot globālo mainīgo 'state'. Ja nav datu, lietotājs pāriet uz autorizācijas lapu.
5. Datu plūsmu diagramma par noliktavu pievienošanu, skatīt 3. pielikuma 2. attēlā, tiek attēlots, kādas darbības un pārbaudes tiek veiktas, kad lietotājs noklikšķināja uz tehniskā atbalsta lapas. Serveris ņem iepriekš saglabātos datus un ģenerē pieprasījumu, lai iegūtu aktīvus tehniskos atbalstu. Pēc tam sūta pieprasījumu SimBASE sistēmā. Apstrādā vaicājumu un ģenerē atbildi un nosūta to atpakaļ uz serveri. Serveris apstrādā atbildi un parāda tehnisko atbalstu.
6. Datu plūsmu diagramma par noliktavu pievienošanu, skatīt 3. pielikuma 2. attēlā, tiek attēlots, kādas darbības un pārbaudes tiek veiktas, kad lietotājs nonācis uz uzdevumu lapu. Serveris ņem iepriekš saglabātos datus un ģenerē pieprasījumu, lai iegūtu aktīvus tehniskos atbalstu. Pēc tam sūta pieprasījumu SimBASE sistēmā. Apstrādā vaicājumu un ģenerē atbildi un nosūta to atpakaļ uz serveri. Serveris apstrādā atbildi un parāda tehnisko atbalstu. Ja lietotājam nav neviena aktīva uzdevuma, serveris parādīs SimBASE ziņojumu par to, ka netika atrasts neviens uzdevums.
7. Datu plūsmu diagramma par noliktavu pievienošanu, skatīt 3. pielikuma 2. attēlā, tiek attēlots, kādas darbības un pārbaudes tiek veiktas, kad lietotājs vēlas rakstīt komentāru par paveikto darbu. Šim nolūkam ir jāievada dati, šis ir komentāra teksts, patērētais laiks un izpildītā darba tips. Kad datu ievade ir ievadīta, serveris pārbauda aizpildītos datus, un, ja radās kļūda, no lietotāja sūta kļūdas tekstu. Ja visi dati ir ievadīti pareizi, serveris ir gatavs pieteikties sistēmā. Simbase apstrādā vaicājumu un pievieno komentāru objektam. Pēc tam serveris pārbauda, vai objekts, kuram pievienots komentārs, ir uzdevums un, ja tas ir uzdevums, tiek parādīts pēdējais
8. **LIETOTĀJA CEĻVEDIS**
   1. **Sistēmas prasības aparatūrai un programmatūrai**

Priekš šis sistēmas izmantošanas minimālas sistēmas prasības ir attēlotas tabula zemāk, lai izmantot šis sistēmas piedāvājumus.

5.1 tabula

Tabula **“Minimālas sistēmas prasībās”**

|  |  |
| --- | --- |
| Sistēmas daļa | Minimālā prasība |
| Operētājsistēmā | Windows 10 |
| Procesors | Intel Core i-5 |
| Operatīvā atmiņa | 1GB RAM |
| Brīvas vietas uz diska | 1 GB |

Lai izmantot lietotāja interfeisu, var izvēlēties vienu no piedāvātiem pārlūkiem:

* Google Chrome 83.0.4103.61
* Firefox 76.0
* Opera 68.0.3618.125
* Microsoft Edge 83.0.478.37
* Internet Explorer 11.778.18362.0
  1. **Sistēmas instalācija**

Lai sāktu lietot SimBASE sistēmas lietotāja interfeisu, jums ir jāiegādājas SimBASE, sākot no 441A versijas. Pēc tam doties uz nākamo saiti... un lejupielādēt visu saturu.

Lai atbalstītu šo interfeisu, ir nepieciešams serveris. Varat izmantot portatīvo usbserver vai ātrāku xampp.

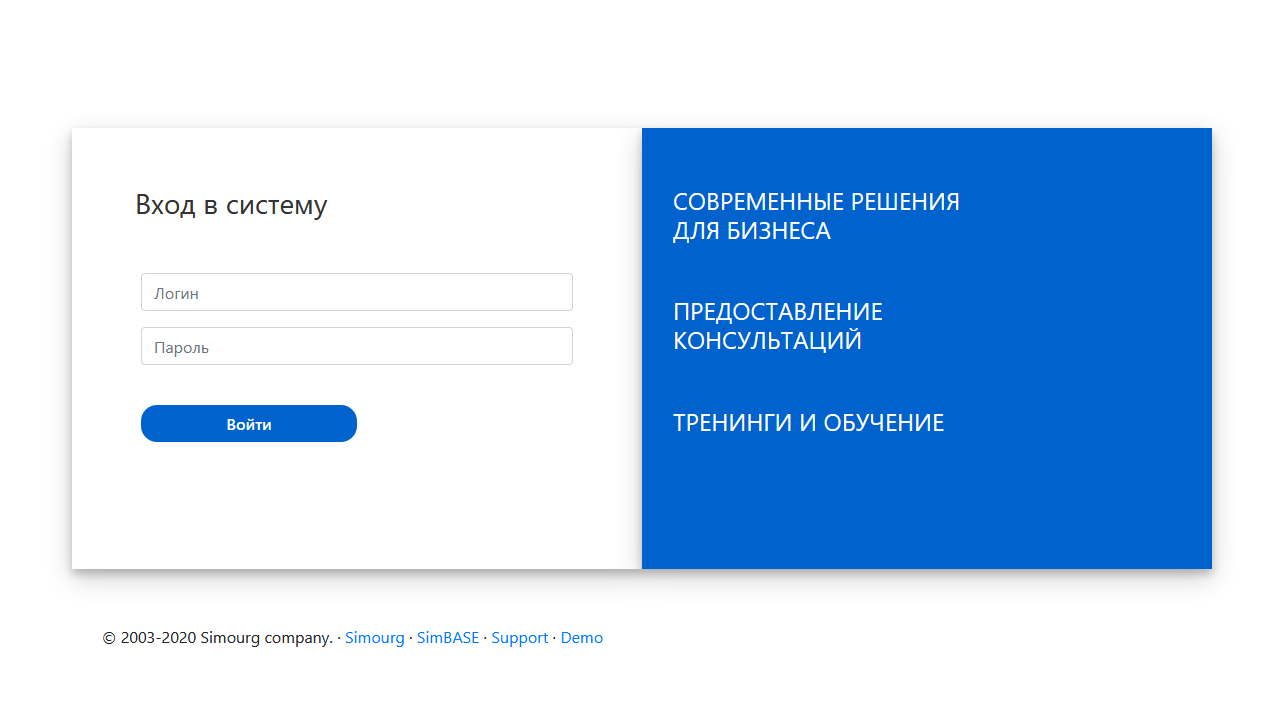
Ir jāmigrē projekts, kas tika lejupielādēts pirms servera mapes. Uz XAMPP servera tā ir “htdocs” un uz usbserver “root”.

Lai palaist interfeisu, vajadzēs izmantot vienu no tīmekļa pārlūkiem kurš ir minēts sistēmas minimālas prasības. Pēc tam pārlūkā ievadīt timekļa adresi localhost vai vai jebkurš domēns, kurā atrodas lietotāja interfeisa failu struktūra. Pēc visām iepriekšējām darbībām varat brīvi izmantot sistēmu.

* 1. **Programmas apraksts**

Kad lietotājs ir pārgājis uz saiti, tas nonāk autorizācijas lapā. Šeit ir jāievada lietotājvārds un parole. Tie paši dati tiek ievadīti SimBASE sistēmā.

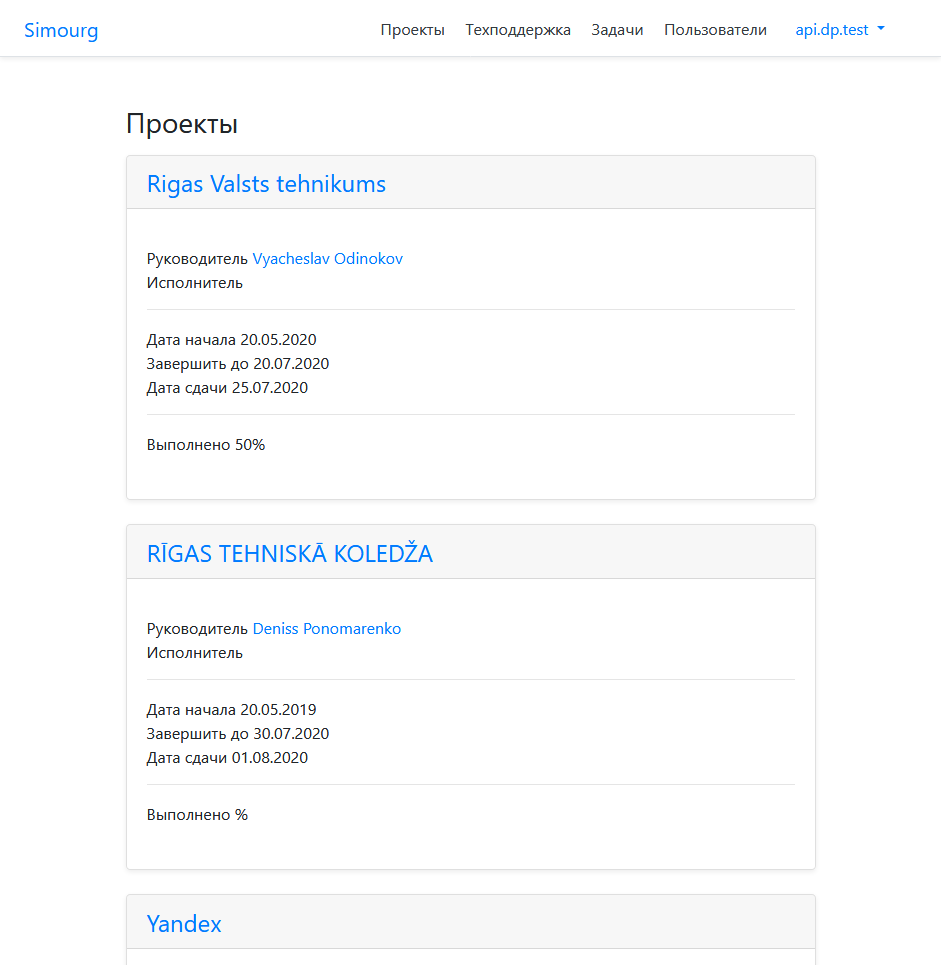
Kad esat ievadījis datus, noklikšķiniet uz pogu “Войти”.



5.3.1.att. Autorizācijas modulis

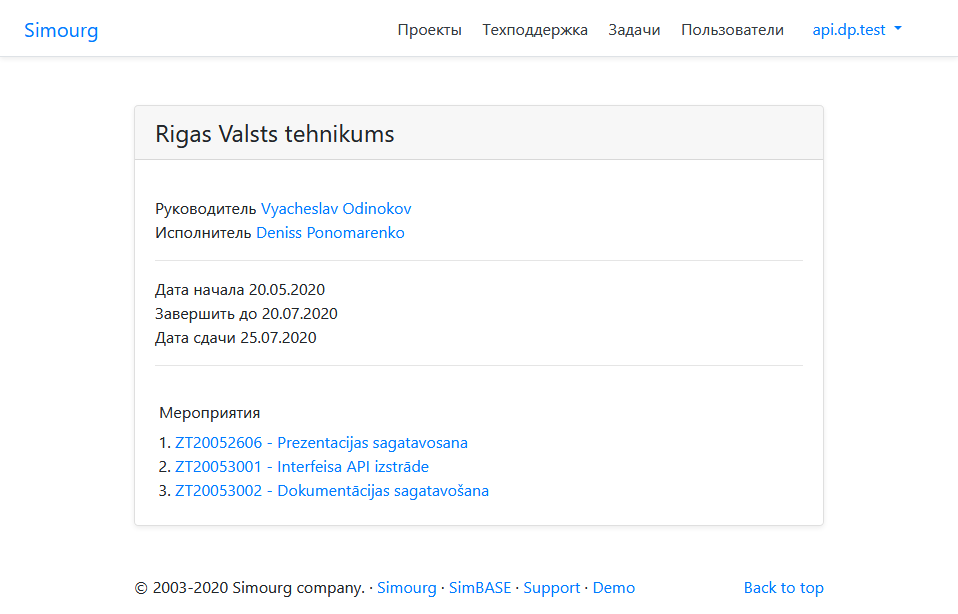
Pēc veiksmīgas autorizācijas lietotājs nokļūst projektu lapā, no kurienes var pāriet uz citām lappusēm, kuras tiek parādītas ekrāna augšdaļā.

Pirmajā lapā tiek parādīts visu aktīvo projektu saraksts. Katrs projekts ir atdalīts ar atsevišķu bloku, kartītes veidā un ietver sevī nepieciešamos datus, tas ir projekta vadītājs, izpildītājs, datumi, kā arī projekta izpildes procenti.



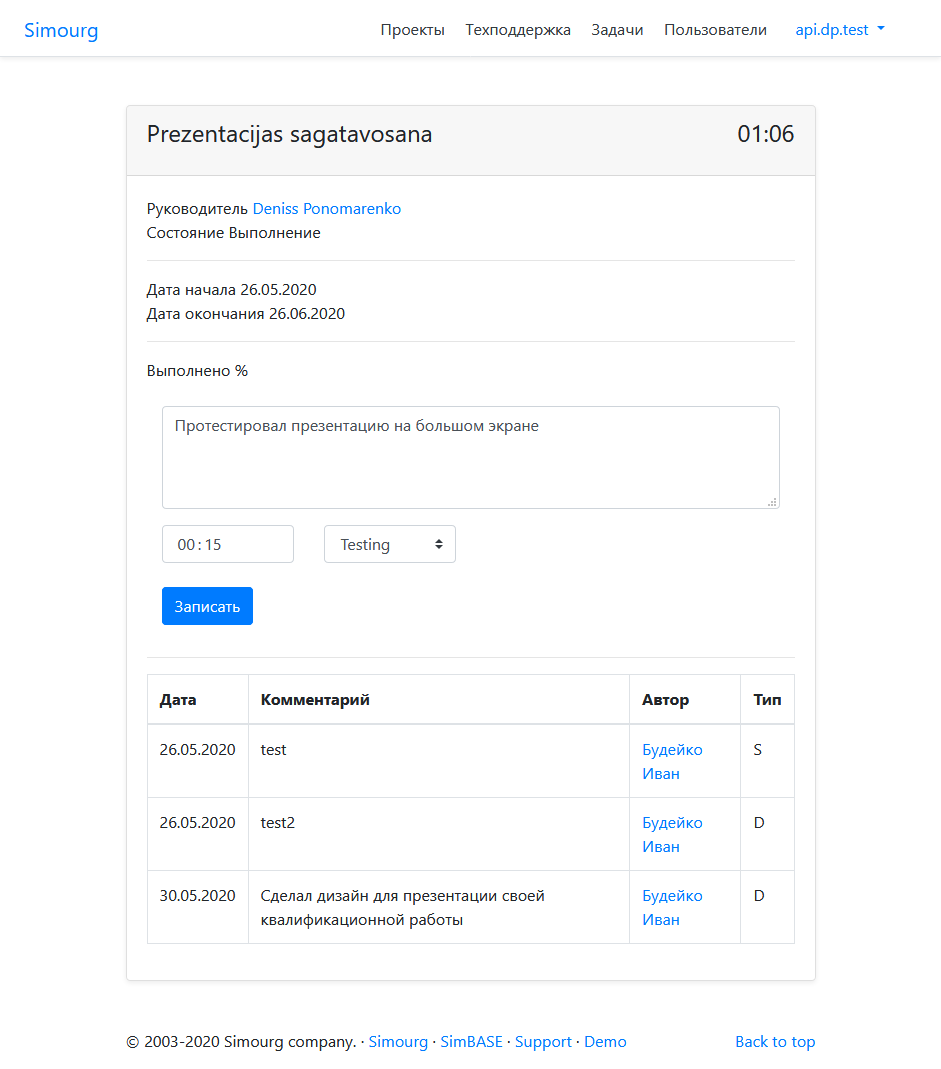
5.3.2.att. Projektu sarakstu modulis

Projekta kartītē zilā krāsā ir redzama pāreja uz lapu ar projekta datiem. Noklikšķinot uz šīs saites, lietotājs nonāk atlasītā projekta lapā.



5.3.3.att. Projektu modulis

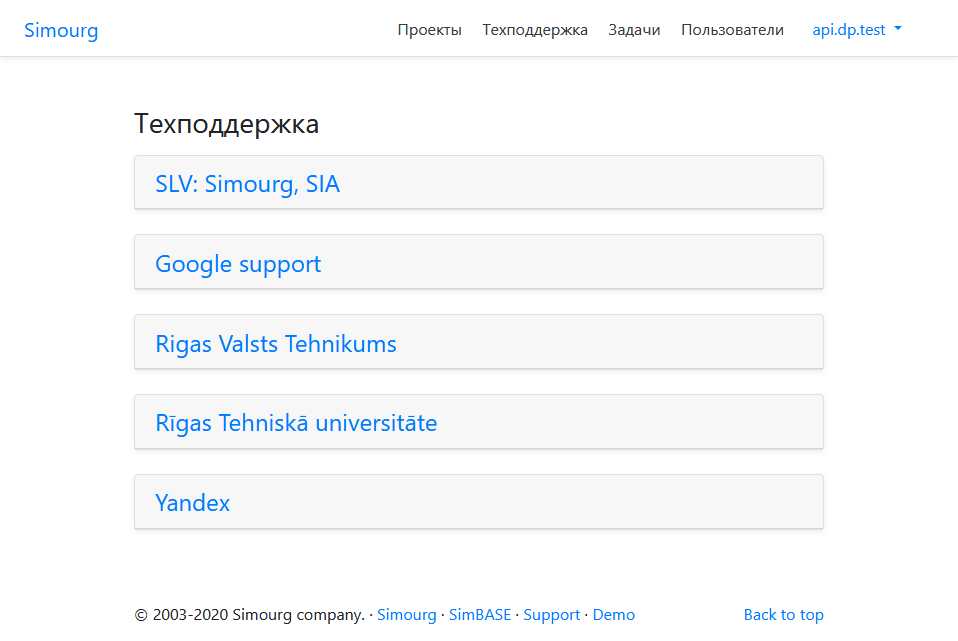
Projekta lapā rāda papildu pasākumu sarakstu, kas attiecas uz šo projektu. Katrs saraksta elements ir saite, kurā lietotājs noklikšķina uz aktivitātes lapu.



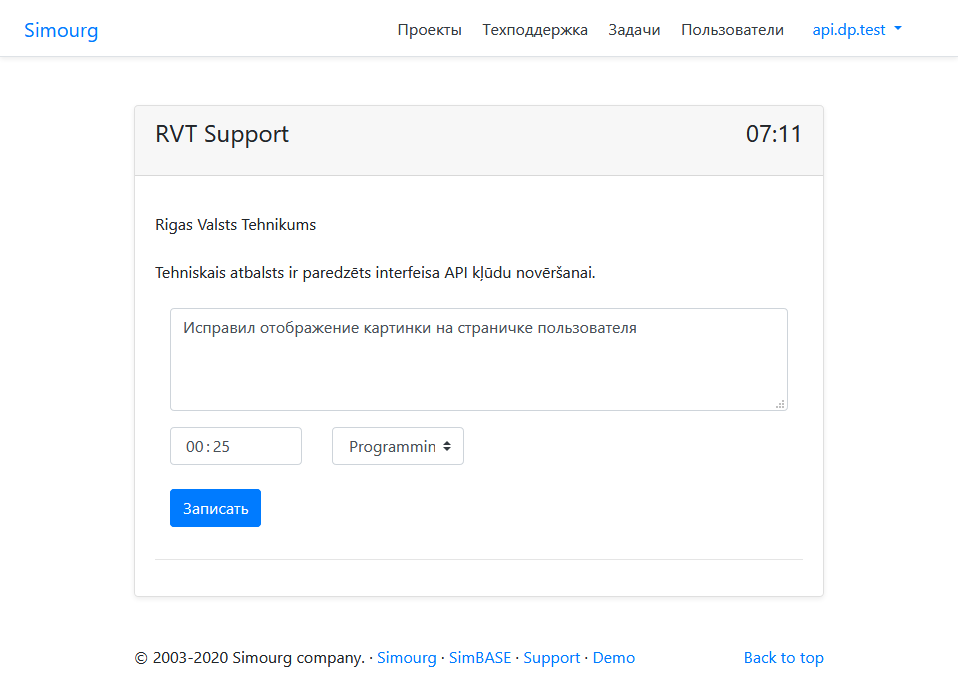
5.3.4.att. Aktivitātes modelis

Pārejot uz aktivitāti, lietotājs var ierakstīt patērēto laiku šī pasākuma ietvaros. Aprakstot veikto darbu un norādot komentāra tipu un laiku. Pēc komentāra ierakstīšanas mainās pasākumam iztērētā laika kopējais skaits, kā arī izpildes procentuālā vērtība.

Nākamais projekts ir tehniskais atbalsts. Tas ir izveidots, lai ierakstītu izpildīto darbu klienta tehniskajā atbalsta ietvaros. Šeit parasti tiek ierakstīti klienta konsultāciju darbi vai avārijas kļūdu labošana.

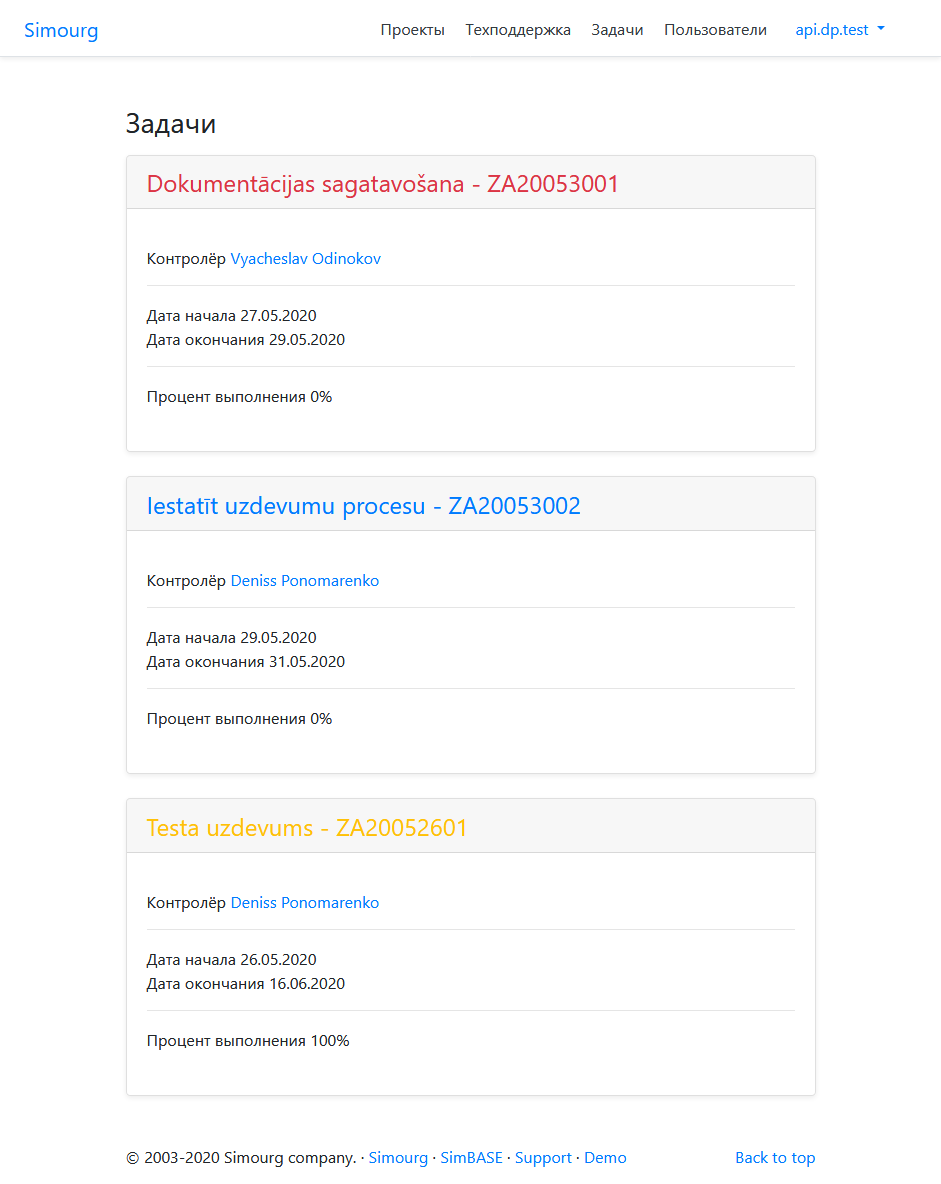


5.3.5.att. Tehniskā atbalsta modulis

Pāreja uz atsevišķu pasākuma lapu ir tāda paša kā projektām. Katram atsevišķam tehniskajam atbalstam ir zila krāsa nosaukums, uz kuru noklikšķinot, lietotājs pāriet uz tehnisko atbalstu.

5.3.6.att. Tehniskā atbalsta lapa

Pārejot uz sadaļu Uzdevumi, lietotājs var redzēt savu aktīvo uzdevumu sarakstu. Pamatojoties uz objekta statusu, mainās tā indikatora krāsa (precīzi arī SimBASE sistēmā). Šis atlase palīdz lietotājam viegli redzēt problēmuzdevumus. Šiem uzdevumiem ir visaugstākā prioritāte.

Uzdevuma kartītē tiek rādīta šāda informācija: uzdevuma nosaukums, kontrolieris, izpildes sākuma datums, beigu datums un izpildes procenti. 

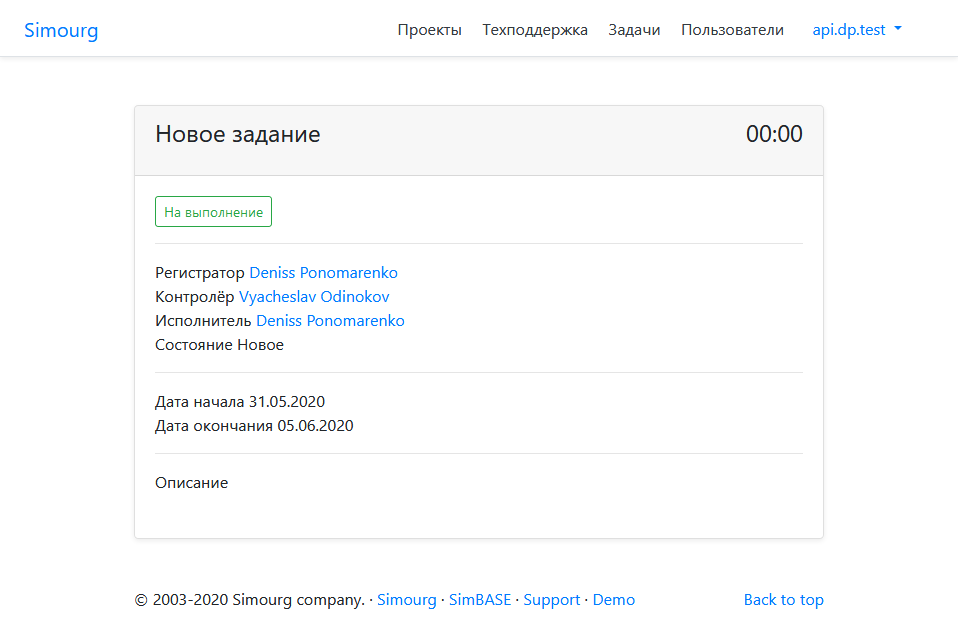
5.3.7.att. Uzdevumu modulis

Noklikšķinot uz uzdevuma nosaukuma, lietotājs pāries uz uzdevuma lapu, kur tai būs pieejama papildu informācija. Arī uzdevuma lapā viņš varēs ierakstīt komentārus un pāriet uz stāvokļiem, izmantojot pogas, kuras atrodas kartes augšdaļā.

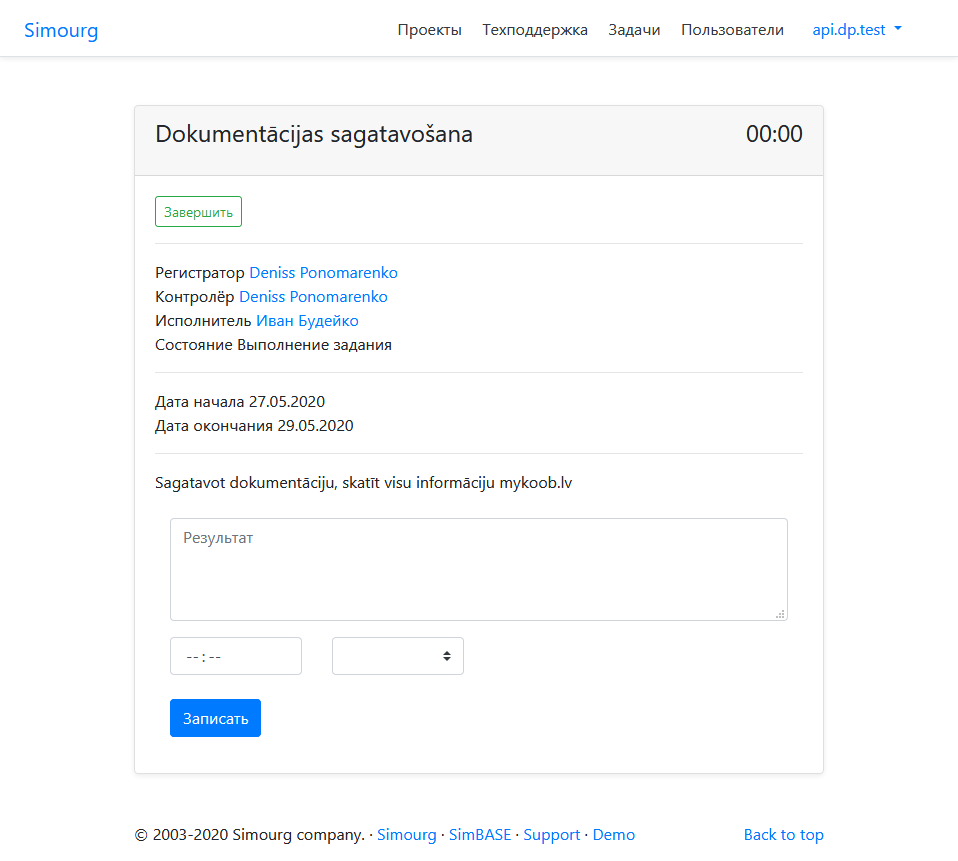
“Jauns” stāvoklī lietotājs var nosūtīt objektu uz stāvoklī “Izpilde”, noklikšķinot uz pogu ar nosaukumu “Izpildīt”.

“Izpildes” stāvoklī, noklikšķinot uz pogu ar nosaukumu “Pabeigt”, objekts pāriet uz stāvoklī “Kontrolēšana”, kur izpildītājs jau ir kontrolieris vai kontroliera vietnieks (vietnieks tiek konfigurēts atsevišķi SimBASE sistēmā). Ja uzdevums ir pabeigts, lietotājs (kontrolieris) var nosūtīt uzdevumu uz apstrādi, ja uzdevums nav pareizi izpildīts vai ir nepieciešamas izmaiņas, vai arī pabeigt uzdevumu, noklikšķinot uz atbilstošu pogu.

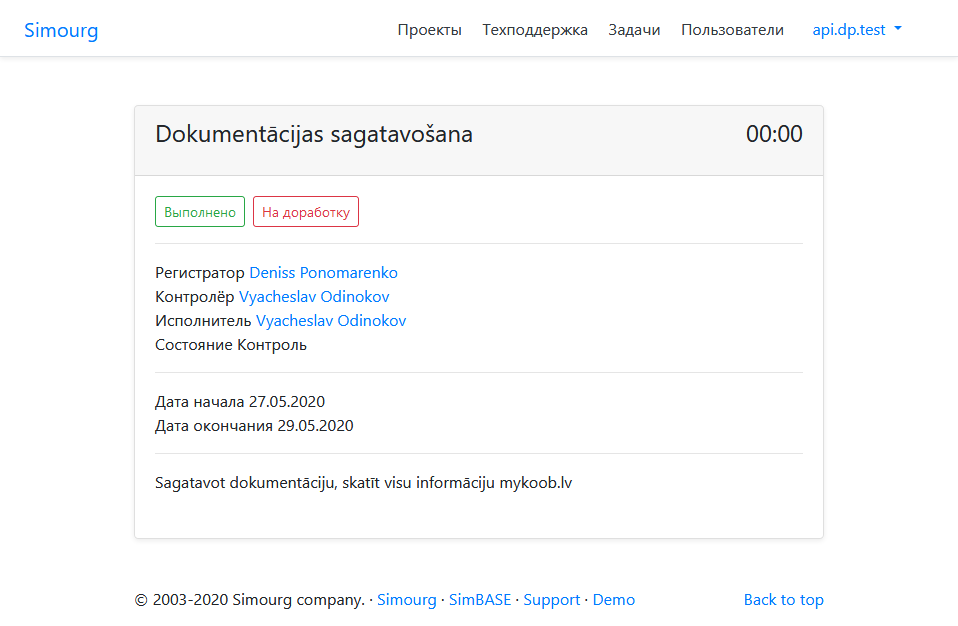
Uzdevuma ietvaros lietotāji var pievienot komentārus tikai stāvoklī “Izpilde”.



5.3.8.att. Uzdevums“Jauna” stāvoklī

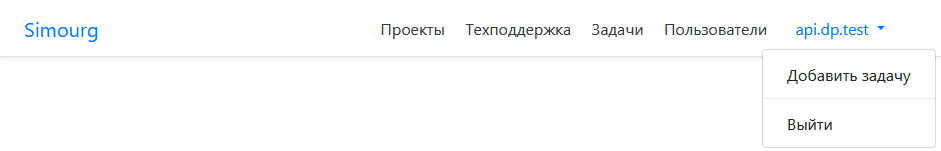


5.3.9.att. Uzdevums “Izpildes” stāvoklī



5.3.10.att. Uzdevums “Kontroles” stāvoklī

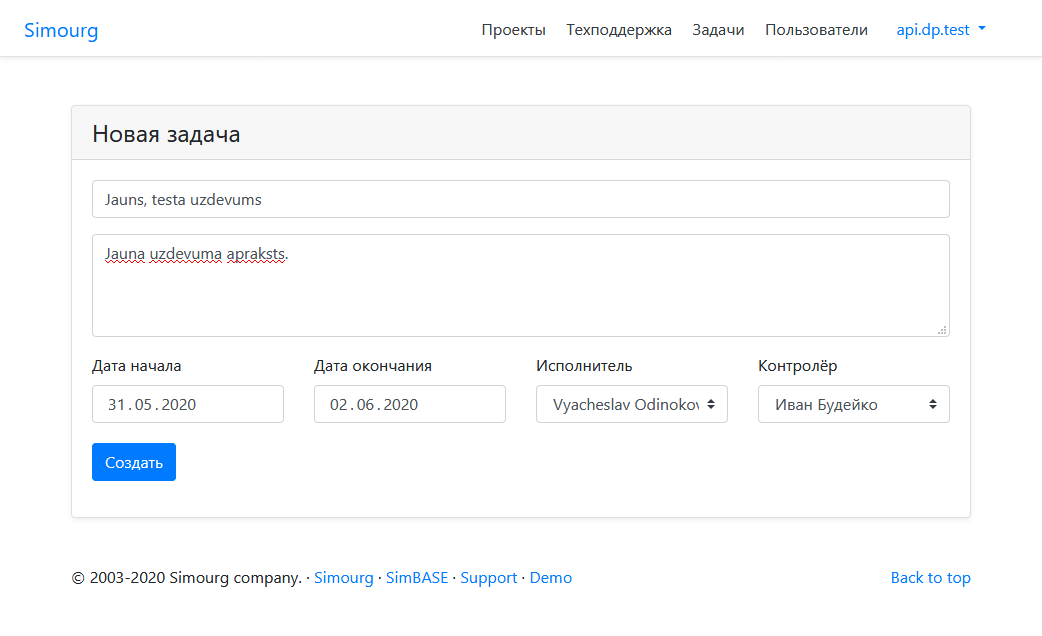
Galvenajā izvēlnē ir norādīts lietotājvārds, uz kuru noklikšķināt, atverot papildu funkcionalitāti: jauna uzdevuma pievienošana un atteikšanās no sistēmas.



5.3.11.att. Papildu sadaļas izvēlnē

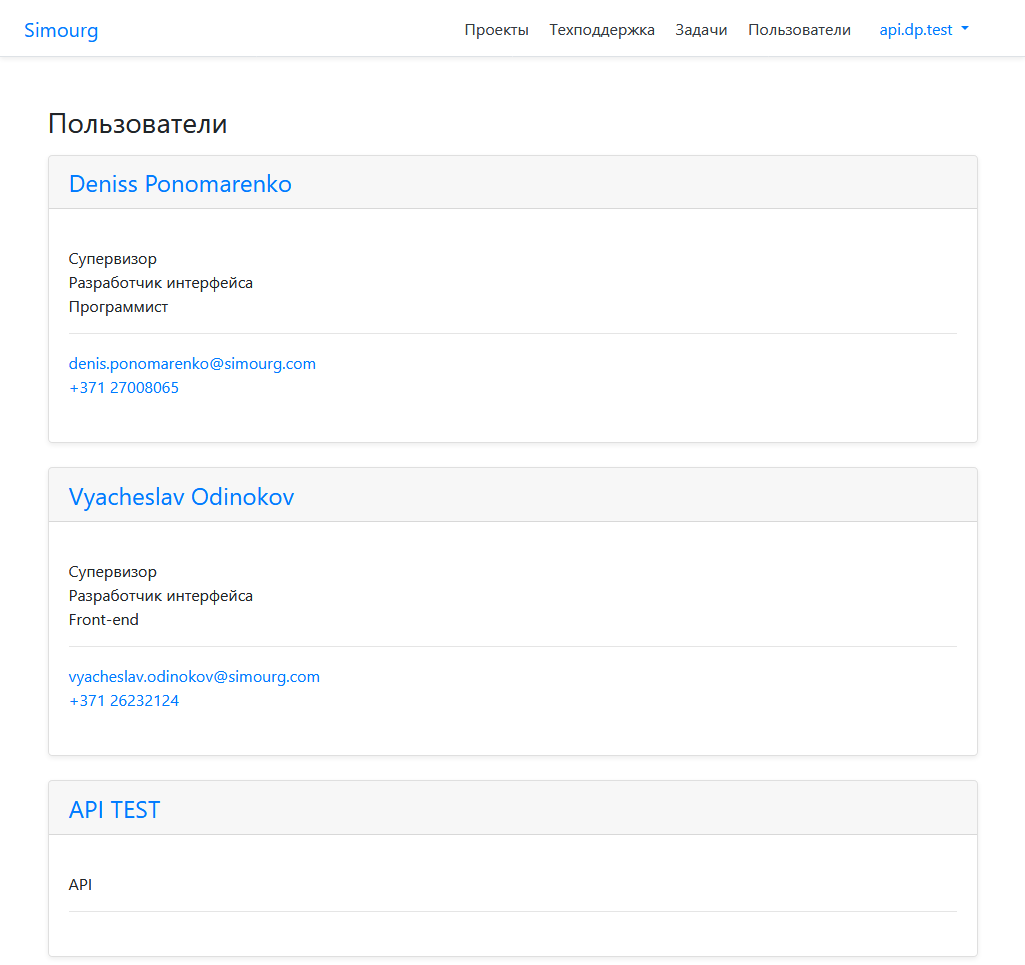
Noklikšķinot uz pogu “Pievienot uzdevumu”, lietotājs pāriet uz jaunu uzdevumas veidošanu, kurā ir jāaizpilda visi lauki un jānospiež pogu ar nosaukumu “Izveidot”.

5.3.12.att. Jauna uzdevuma pievienošana modulis



Pēc tam lietotājam tiek paziņots par jauna uzdevuma sekmīgu pievienošanu un uzdevumu sarakstā parādās tikko izveidotais objekts.

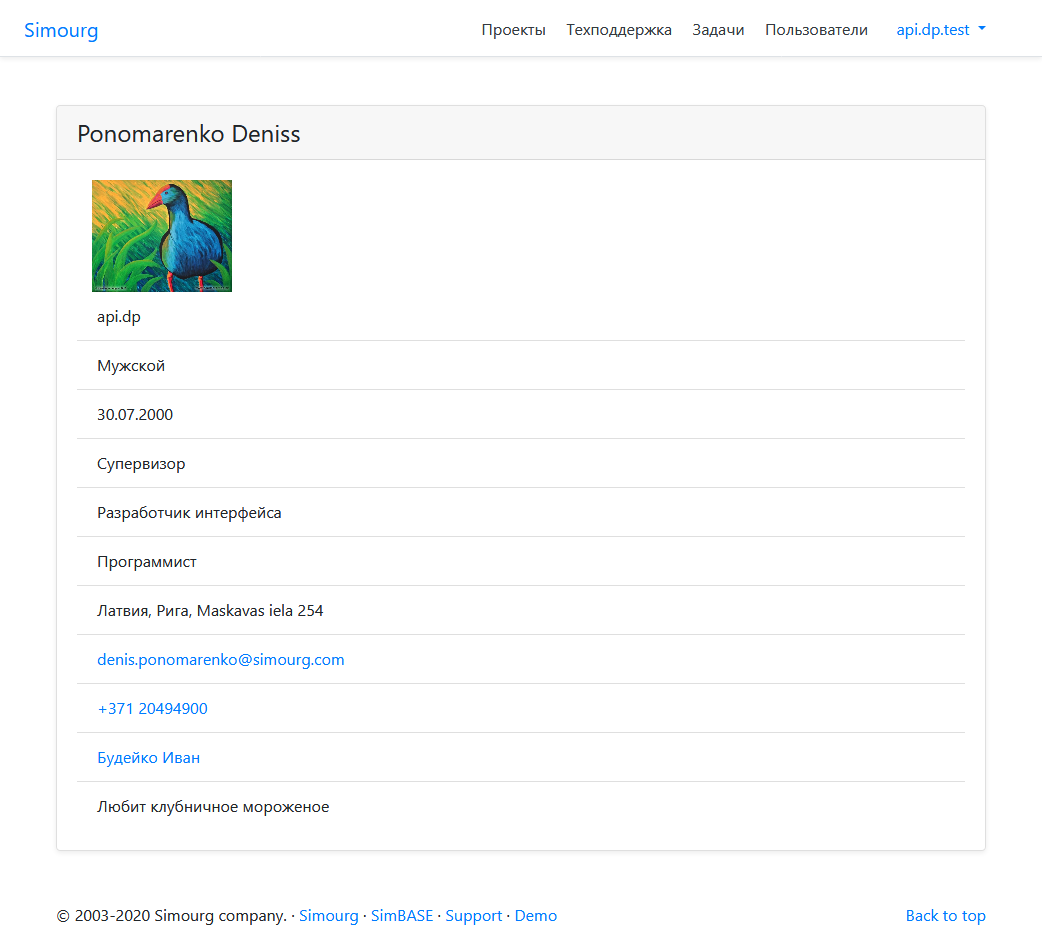
Pastāv arī lietotāju sadaļa, kurā redzams visu aktīvo sistēmas lietotāju saraksts kā vārds un uzvārds, e-pasta adrese un mobilā tālruņa numurs.



5.3.13.att. Lietotāju modulis

Noklikšķinot uz lietotāja vārda un vārda, lietotājs pāries uz lappusi, kur tiks norādīta detalizētāka informācija par lietotāju.

Ja lietotājs pašlaik izmanto mobilo ierīci, noklikšķinot uz tālruņa numura, tas var zvanīt lietotājam, tāds pats mehānisms tiek izmantots, lai nosūtītu e-pasta ziņojumus, ja lietotājs izmanto pakalpojumu Outlook, noklikšķinot uz e-pasta adreses, un tai tiek atvērta e-pasta ziņojuma iesniegšanas veidlapa.

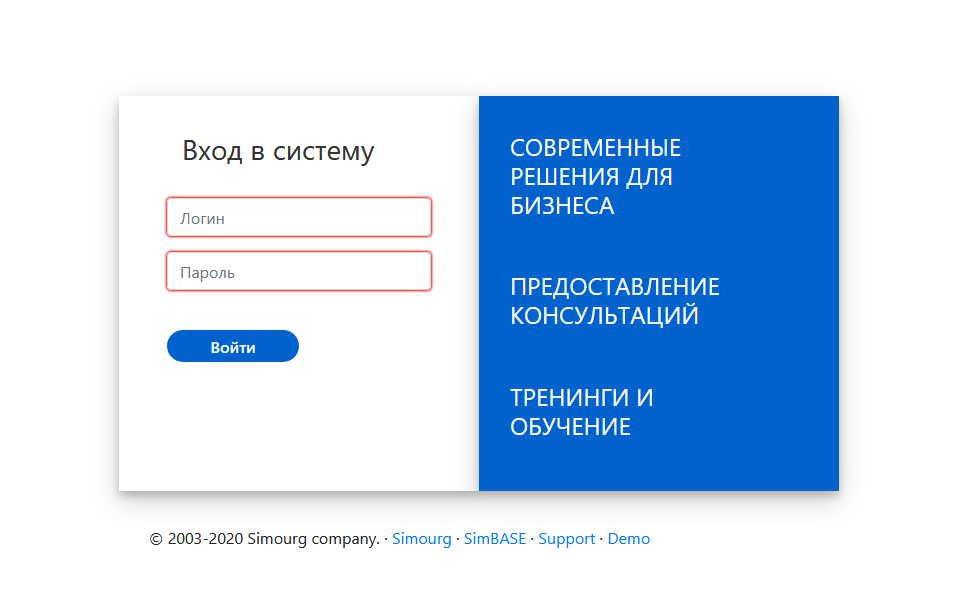


5.3.14.att. Lietotāja kontaktinformācija

* 1. **Testa piemērs**

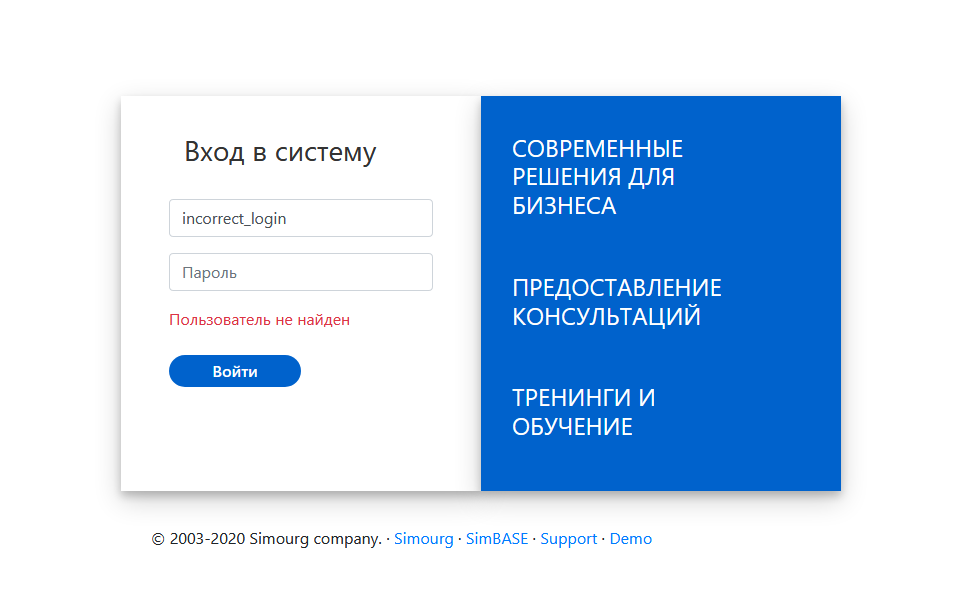
Šajā sadaļa tiks aprakstīts, kā izmantot šo sitēmu no lietottaja skata. Šeit tiks paradīts, kā pareizī pievienot komēntarus, jaunu uzdevumu un kā autorizēt sistēmā un kādas validācijās tiek nodrošinātās lietotaju darbības, lai palīdzēt lietotajam pareizi aizpildīt dažādas formas.

Pirmais solis ir autorizācija. Šajā solī lietotājam ir jāievada savs lietotājvārds un parole, un šo informāciju viņš ievada pašā SimBASE sistēmā. Šie divi lauki ir obligāti. Bez šo lauku aizpildīšanas lietotājs nevarēs autorizēt.



5.4.1.att. Neaizpildītie lauki autorizācijas lappusē

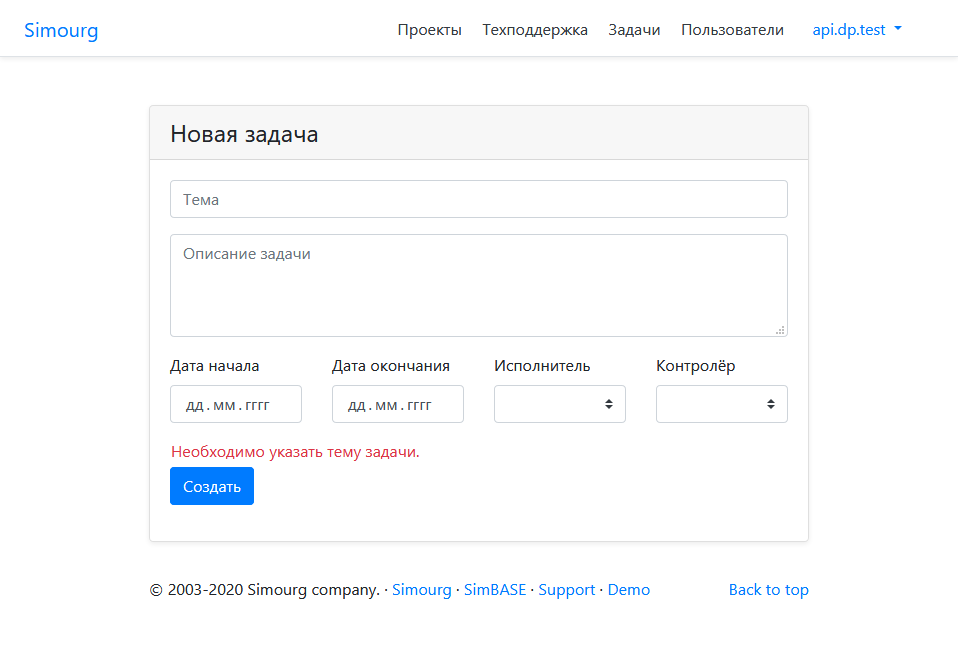
Pēc tam, kad lietotājs ievadījis visus datus un noklikšķinājis uz pierakstīšanās pogas, programma pārbauda, vai lietotājs sistēmā nepastāv. Ja šāda lietotāja nav, programma atdos ziņojumu.



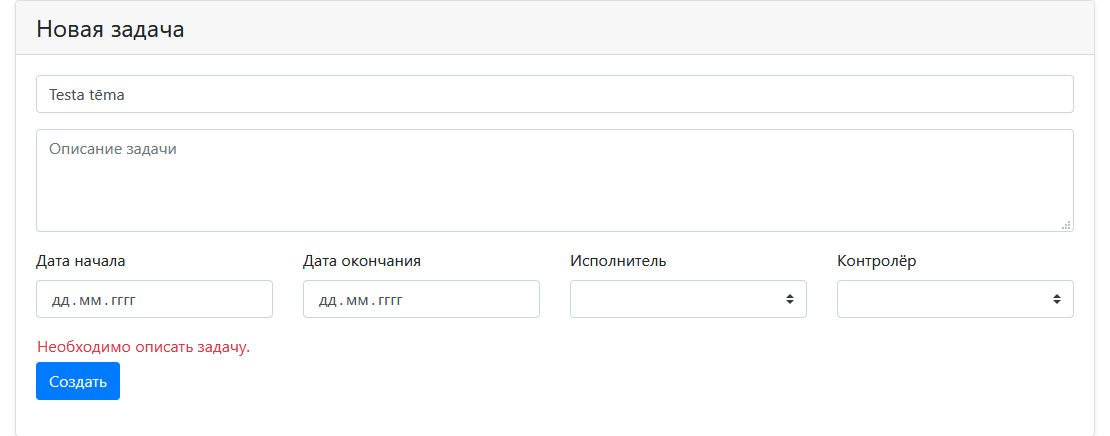
5.4.2.att. Ziņojums par neatrasto lietotāju

Ja lietotājs ievadījis pareizos datus, tas nonāk SimBASE sistēmā, izmantojot lietotāja interfeisa API. Kur viņam ir iespēja pārvietoties pa noteiktiem moduļiem. Izvēlnē tiks rādīts viņa autorizētā lietotāja vārds, lai lietotājs redzētu, ar kuru kontu viņš ir pierakstījies.

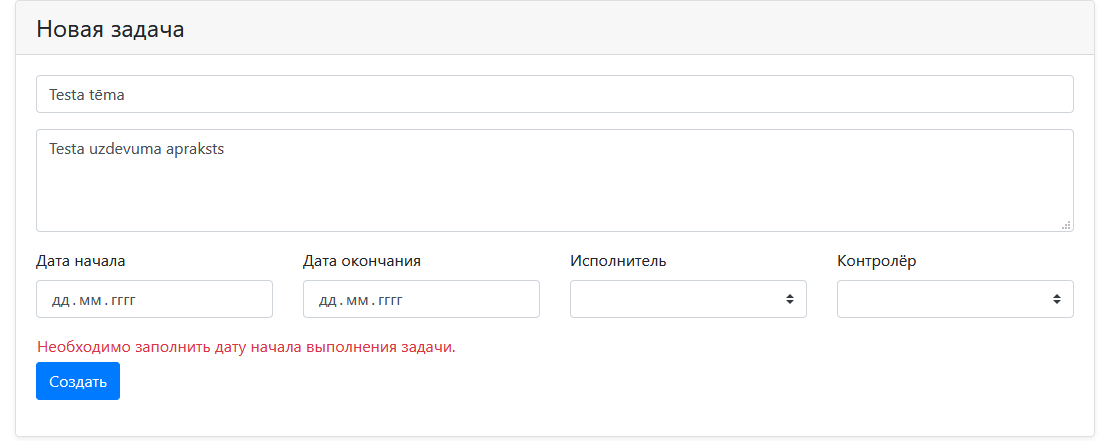
Ja lietotājs nolems pievienot jaunu uzdevumu, tam tiks rādīta datu aizpildīšanas veidlapa un, ja aizpildīsit lauka datus, lietotājs saņems kļūdas ziņojumu zem pēdējā veidlapas lauka. Piemēram, ja viņš neizrādīs uzdevuma tēmu, programma nosūtīs viņam atbilstošu kļūdas tekstu. Turklāt visiem korektiem datiem jāpaliek laukos.



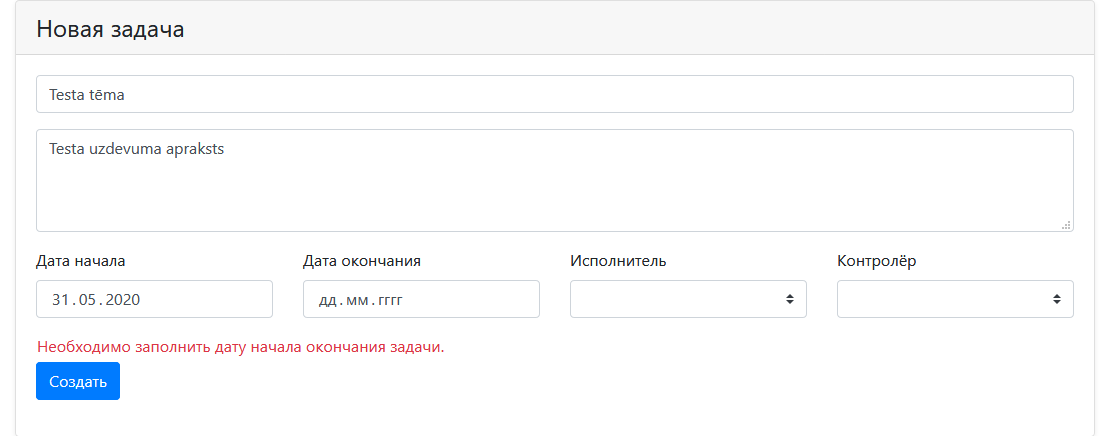
5.4.3.att. Nav norādīts uzdevuma tēma



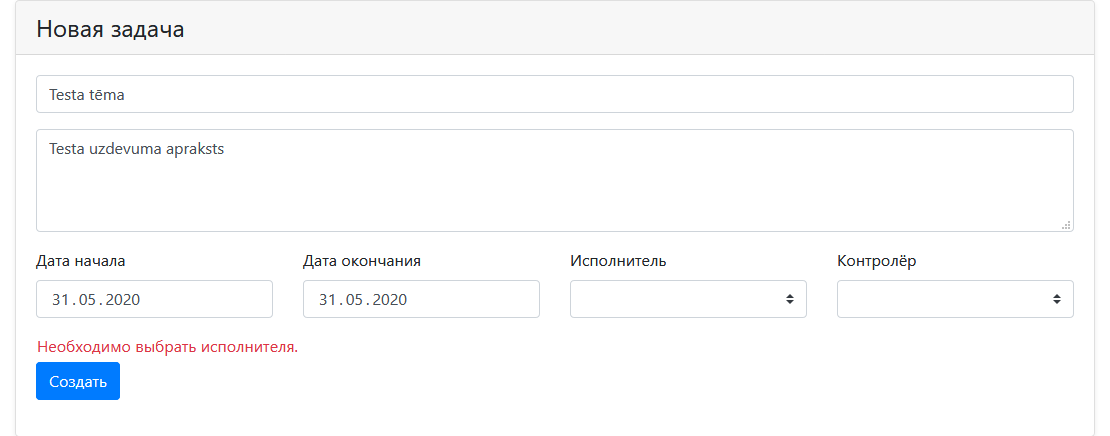
5.4.4.att. Nav norādīts uzdevuma apraksts



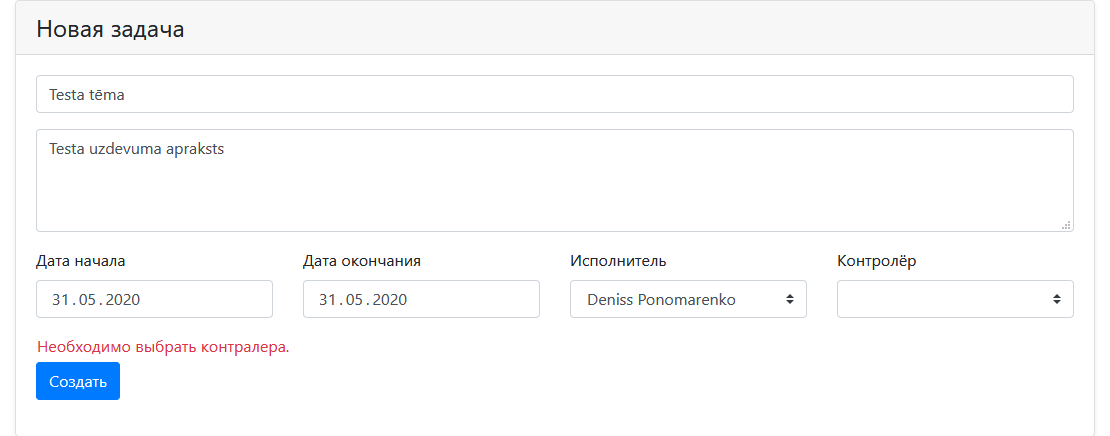
5.4.5.att. Nav norādīts uzdevuma izpildes datums



5.4.6.att. Nav norādīts uzdevuma beigu datums

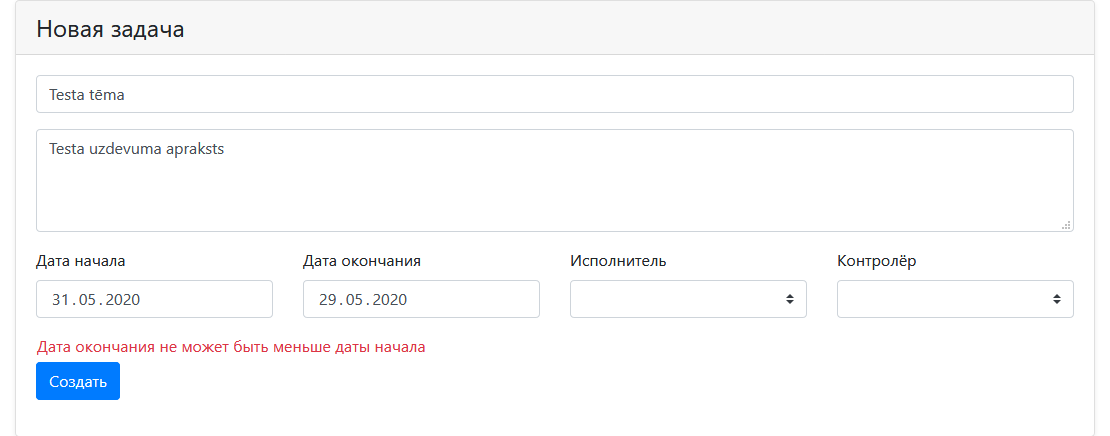


5.4.7.att. Nav norādīts izpildītājs



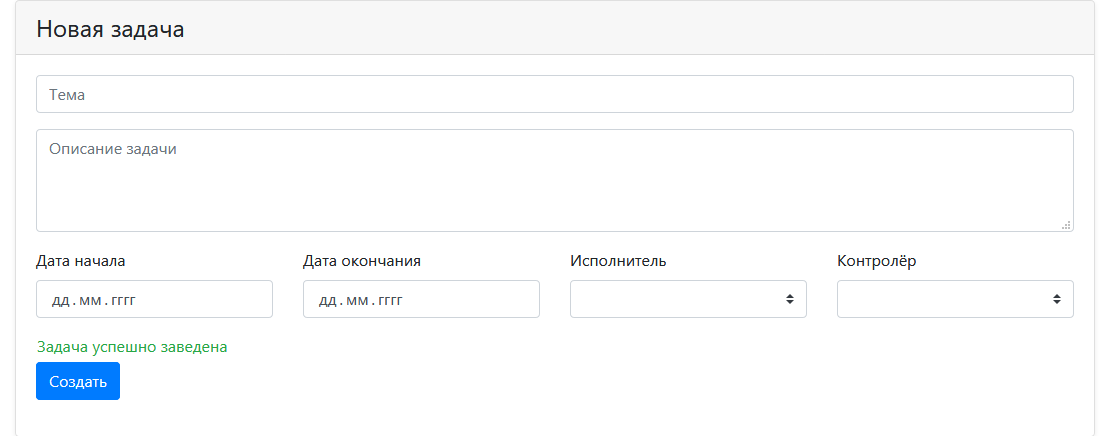
5.4.8.att. Nav norādīts kontrolieris

Sākuma datums nevar būt lielāks par vai vienāds ar beigu datumu, tāpēc, ja lietotājs norāda sākuma datumu, kas ir vēlāks par sākuma datumu, sistēma atgrieza kļūdas tekstu.



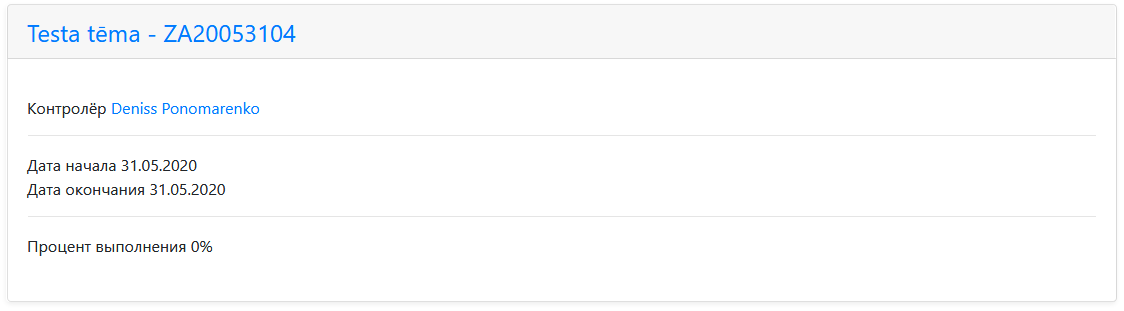
5.4.9.att. Sākuma datums ir vēlāks par beigu datumu

Kad visi dati ir veiksmīgi ievadīti, sistēma paziņos lietotājam par uzdevuma sekmīgu pievienošanu.



5.4.10.att. Ziņojums par uzdevuma sekmīgu pievienošanu

Pēc tam lietotājs var pārliecināties par to, ieejot uzdevuma sadaļā.



5.4.10.att. Uzdevumu sarakstā izveidotais uzdevums

**NOBEIGUMS**

Šī kvalifikācijas darba mērķis ir izpildīts. Izstrādātā sistēma ir stabila un ērta liela lietotāju skaita lietošanai. Šo produktu var izmantot vidēji statistikas uzņēmums, kurā strādā līdz 100 darbiniekiem. Darba laika uzskaites process ir noderīgs gan darba devējam, gan pašam darbiniekam, kurš var redzēt savu iepriekšējo darbu rezultātu.

Produkts nav ideāls, jo tas pagaidām neietver sevī integrāciju ar citām SimBASE sistēmām. Bet ir ļoti labs vienkāršais standarts SimBASE interfeiss.

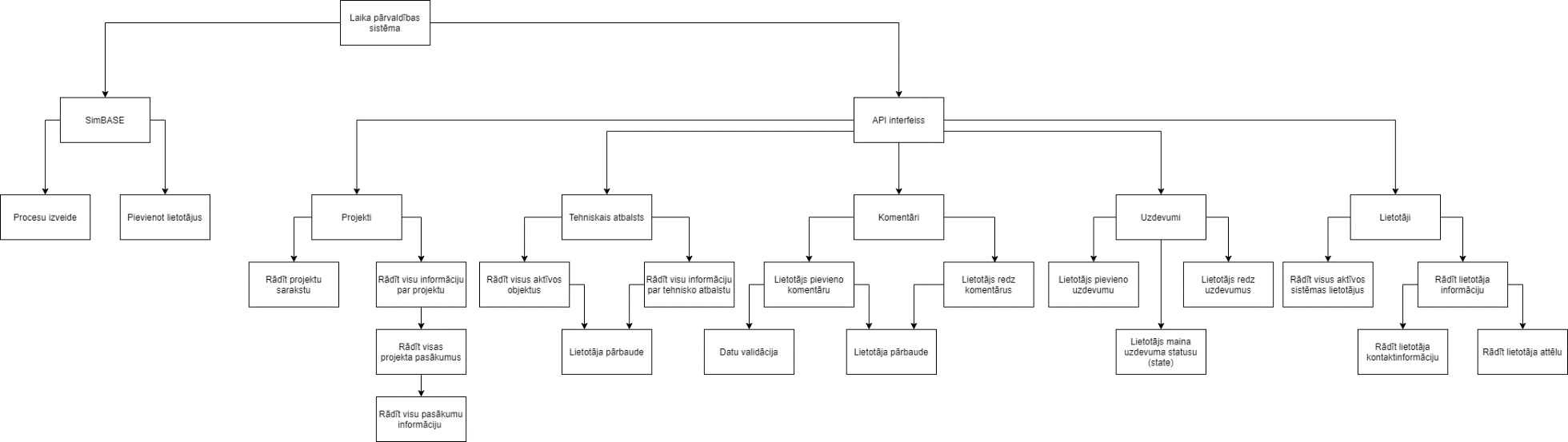
**INFORMĀCIJAS AVOTI**

1. SimBASE dokumentacija - <https://ru.td.simbase.eu/index.php> (Resurss apskatīts no 15.01.2020. līdz 30.05.2020).
2. PHP programmešānas valoda dokumentacija - <https://www.php.net/manual/ru/> (Resurss apskatīts no 12.01.2020. līdz 30.05.2020).
3. Norādījumi darbu noformēšanai - <https://www.estudijas.rvt.lv/pluginfile.php/15912/mod_resource/content/1/Norādījumi%20darba%20noformēšanai.pdf> (Resurss apskatīts 15.04.2020.).
4. Visual Studio Code – Teksta redaktora ielāde <https://code.visualstudio.com/updates/v1_42> (Resurss apskatīts 23.05.2020.).
5. HTML un CSS gatavie elementi - <https://getbootstrap.com/docs/4.4/examples/>
6. Piemērs, kā izveidot attelu no BASE64 - <https://stackoverflow.com/questions/15153776/convert-base64-string-to-an-image-file> (Resurss apskatīts 14.05.2020)

**PIELIKUMI**

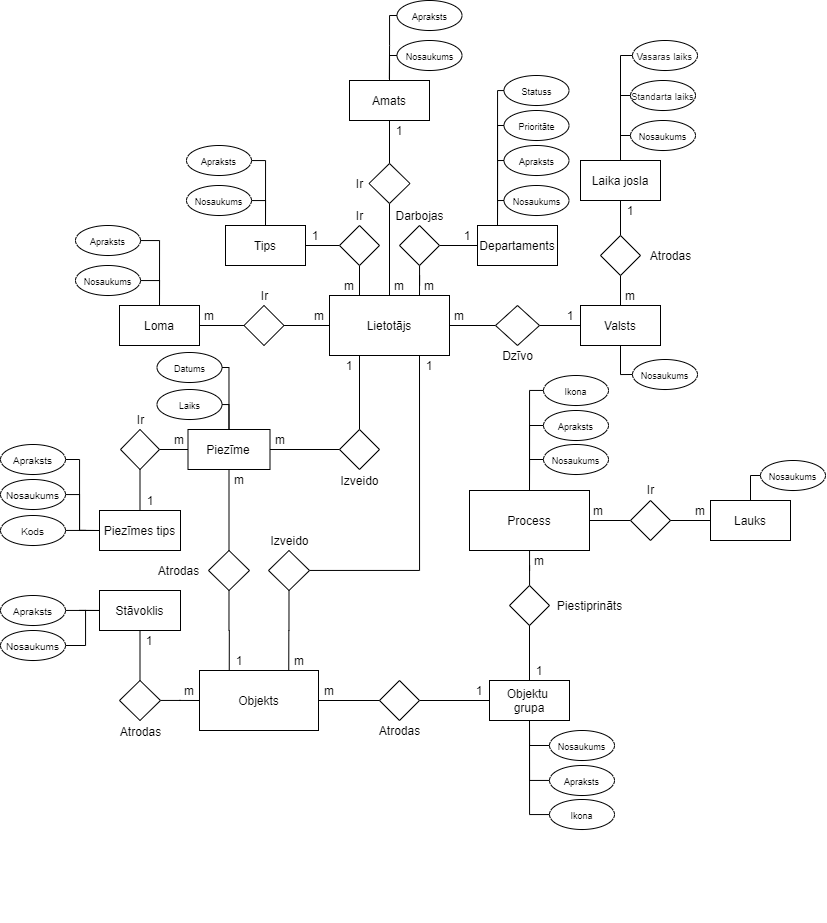
1. pielikums

**Funkcionālas Dekompozīcijas Diagramma**



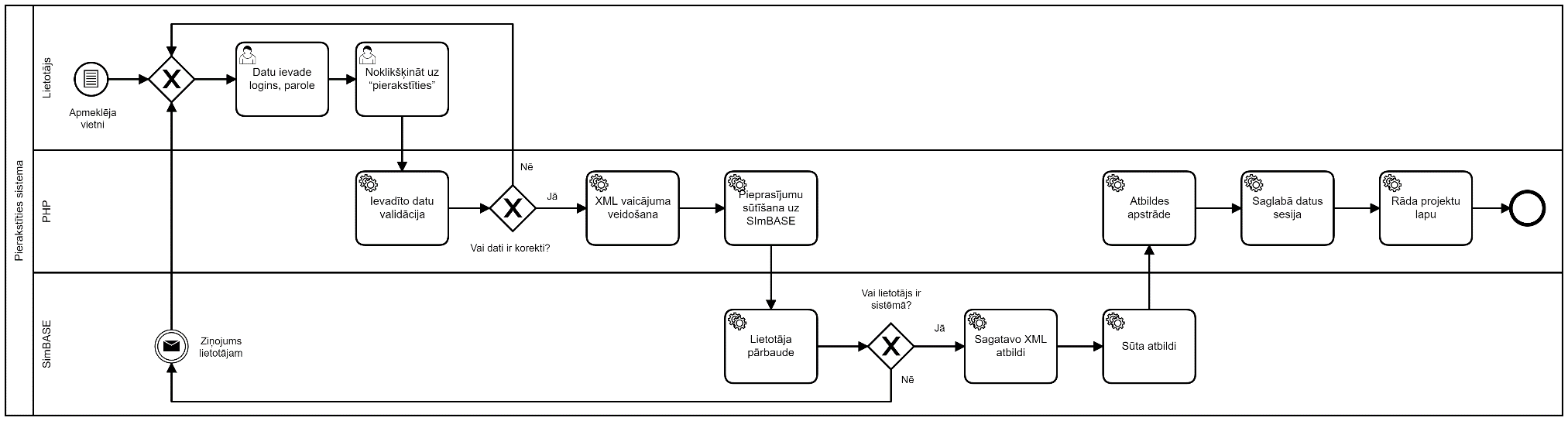
1. pielikums

**ER diagramma**

****

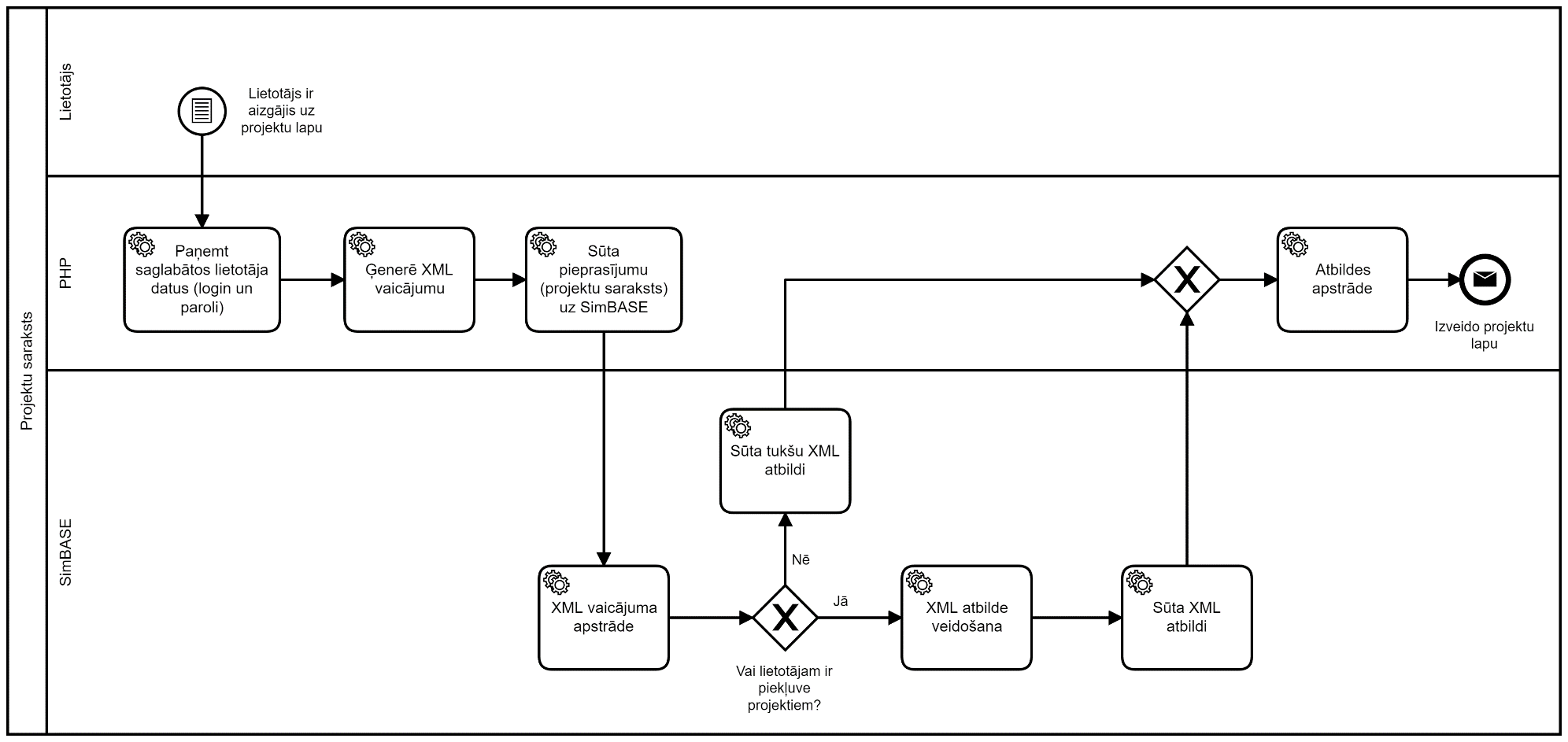
1. pielikums

**Datu plūsmu diagramma “Lietotāja autorizācija”**

****

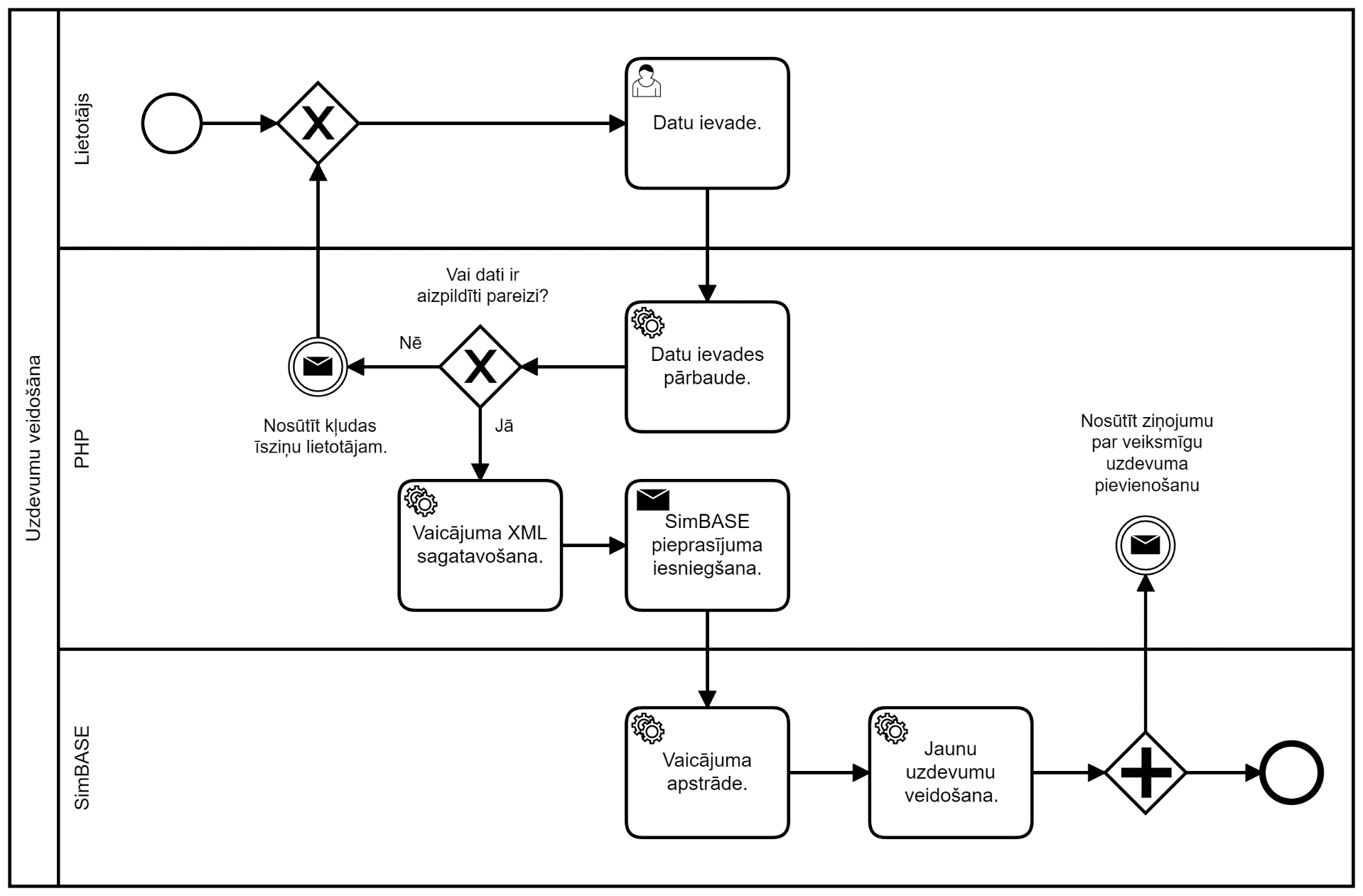
3. pielikuma turpinājums

**Datu plūsmu diagramma “Projekta datu iegūšana”**



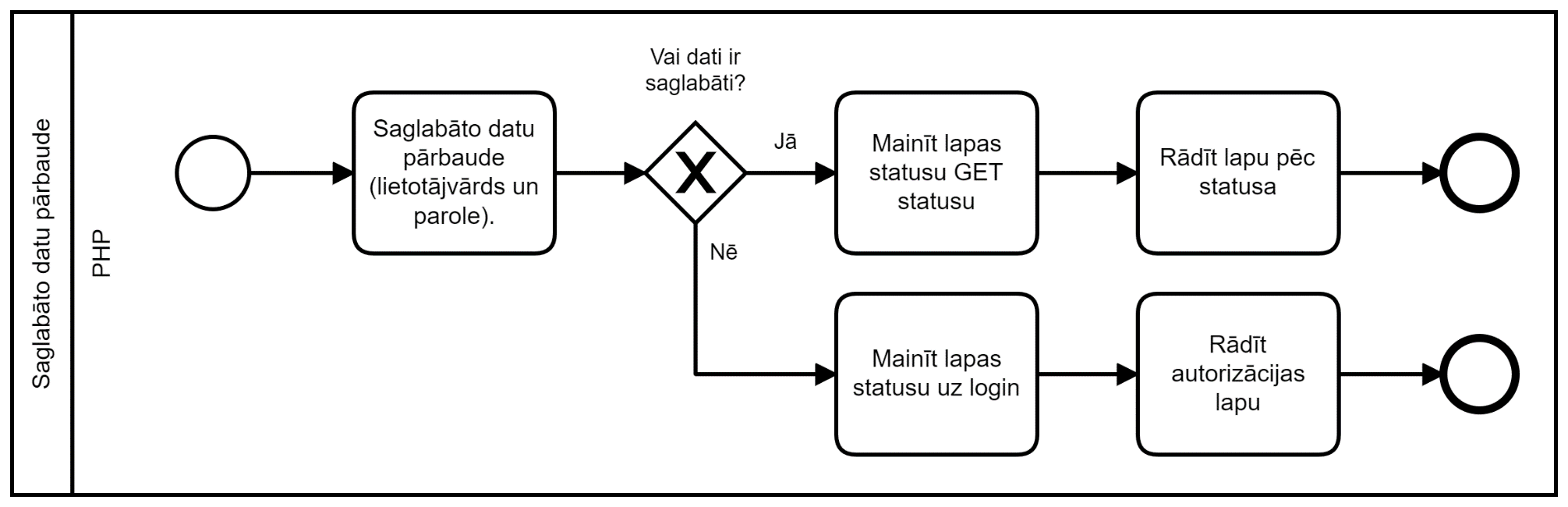
3. pielikuma turpinājums

**Datu plūsmu diagramma “Uzdevumu veidošāna”**



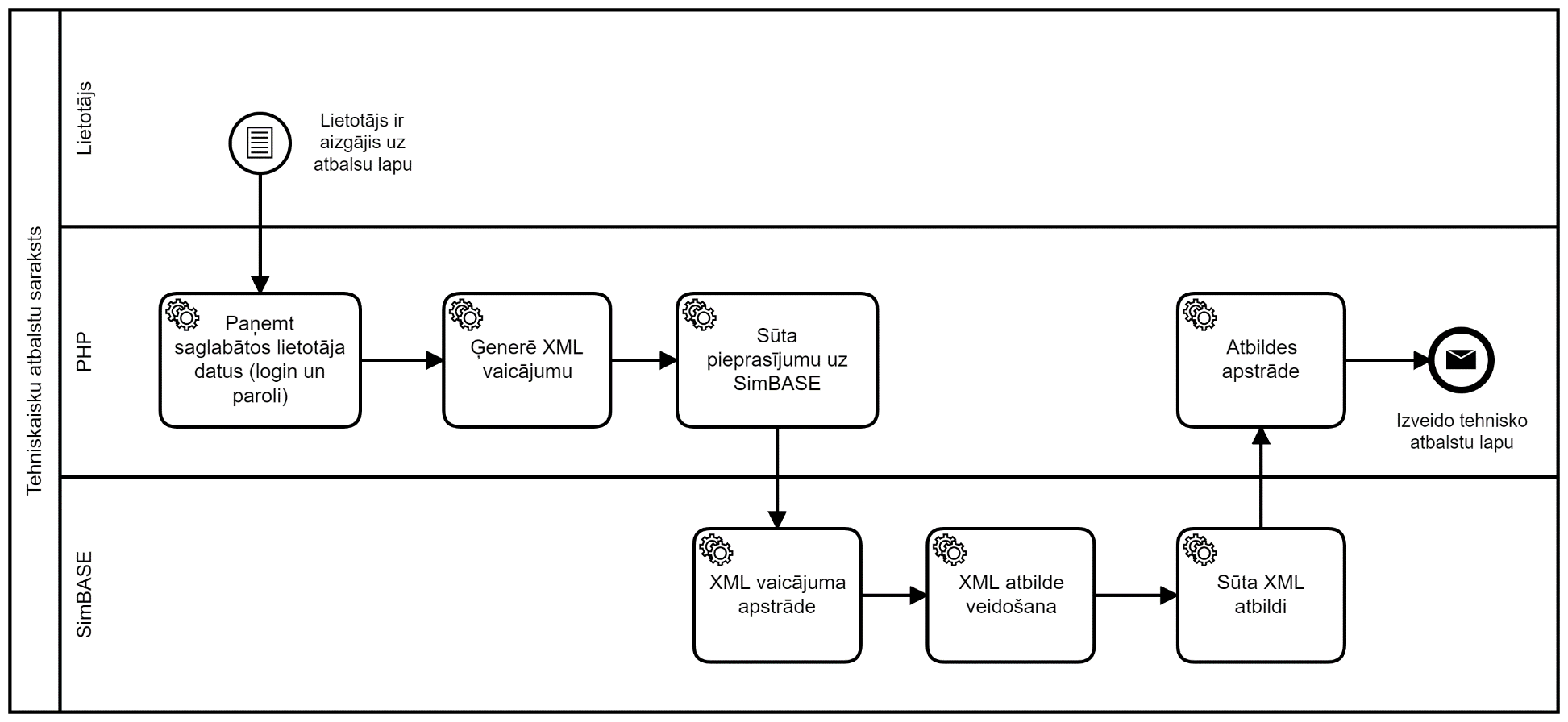
3. pielikuma turpinājums

**Datu plūsmu diagramma “Saglabātos datus pārbaude”**



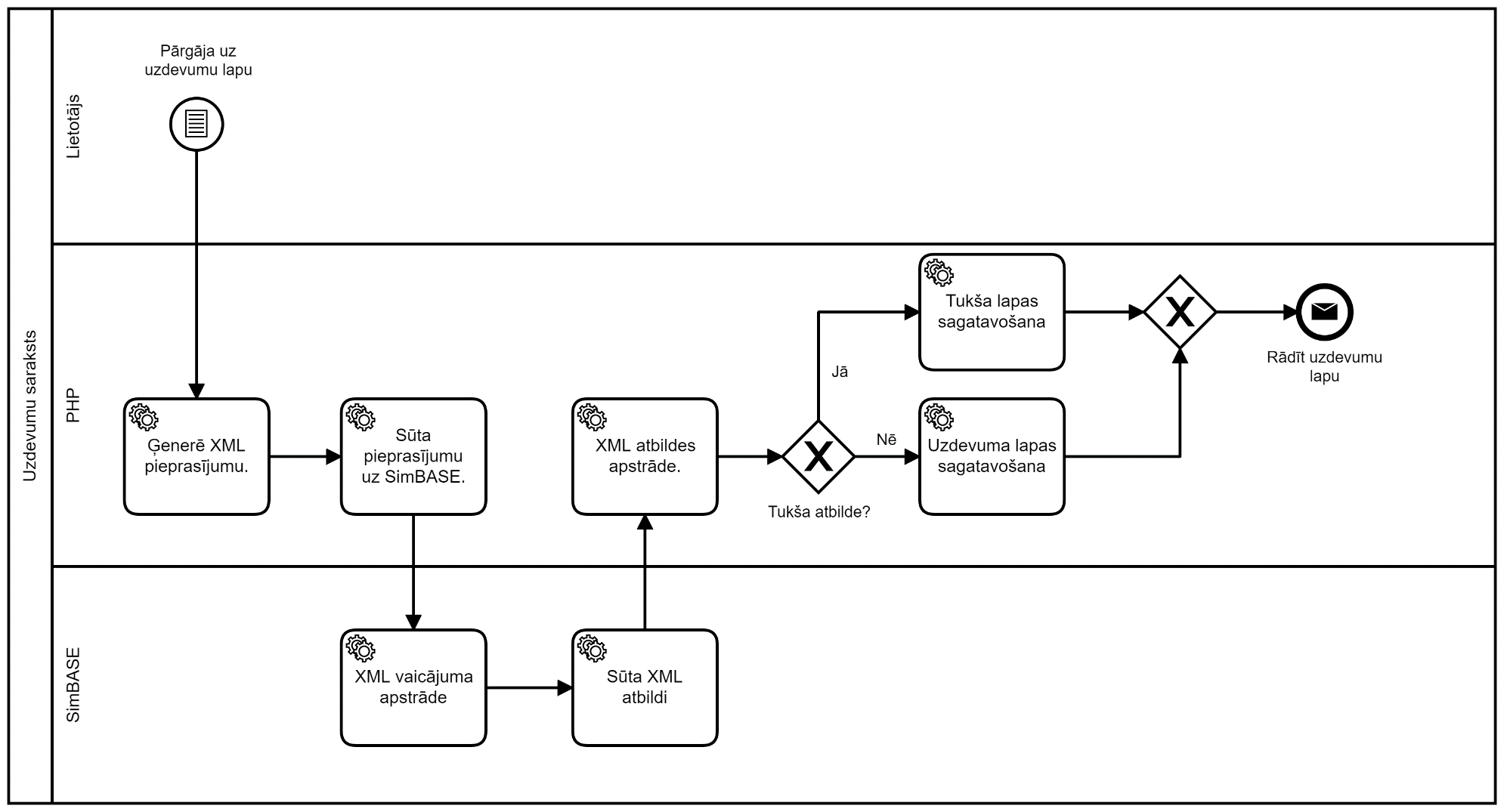
3. pielikuma turpinājums

**Datu posma diagramma “Tehniskaisku atbalstu saraksts”**



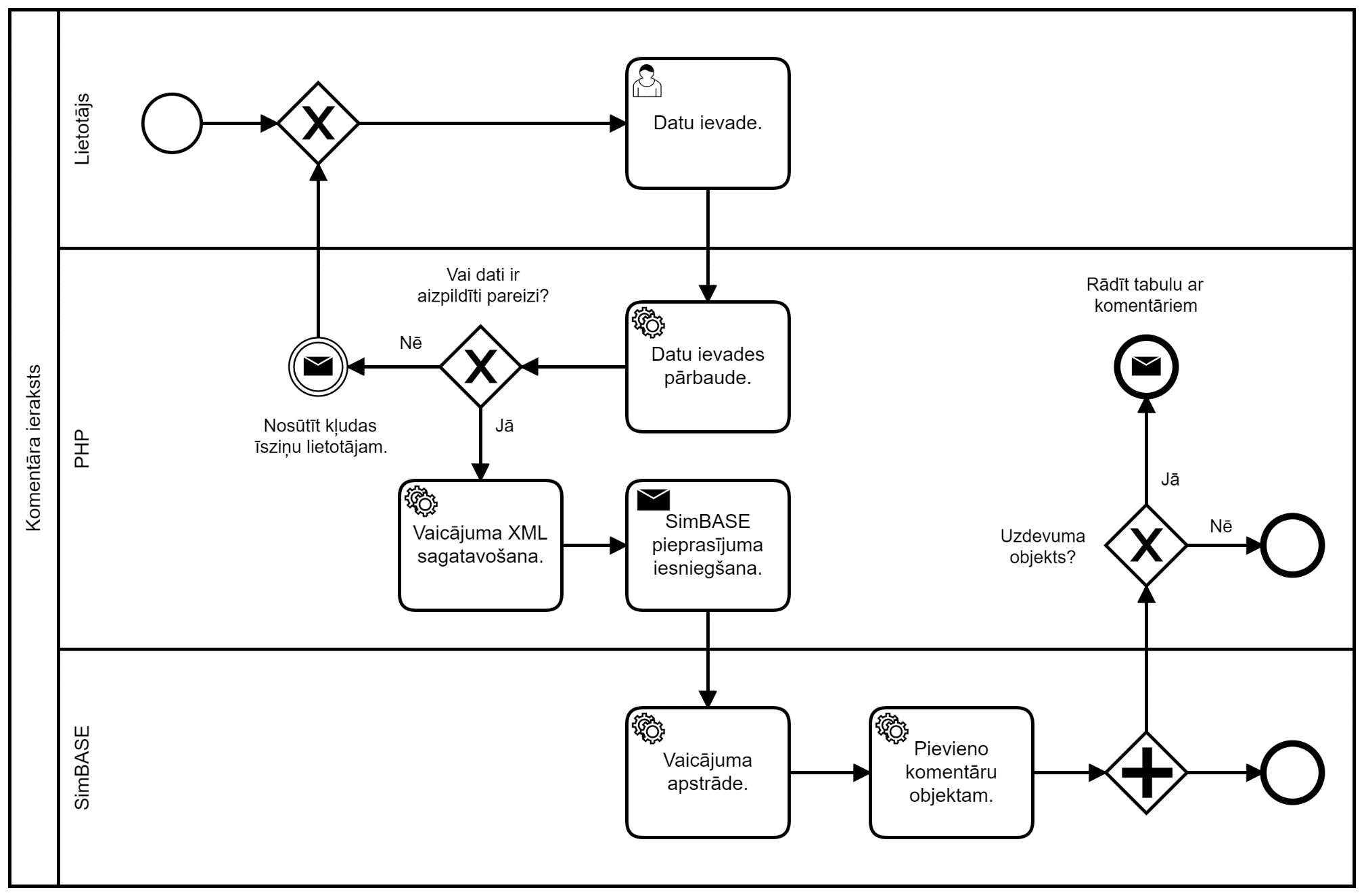
3. pielikuma turpinājums

**Datu posma diagramma “Uzdevumu saraksts”**



3. pielikuma turpinājums

**Datu posma diagramma “Komentāra ieraksts”**



1. pielikums

**Programmas kods “function/function.php”**

<?php

    define('FUNCTIONS',\_\_DIR\_\_);

    function login\_validation( $login, $password ){

        $login    = filter\_var(trim($login), FILTER\_SANITIZE\_STRING);

        $password = filter\_var(trim($password), FILTER\_SANITIZE\_STRING);

        $error\_msg = "";

        if( $password == "" ){

            $error\_msg = "Введите пароль";

        }

        if( $login == "" ){

            $error\_msg = "Введите логин";

        }

        if( $login == "" && $password == ""){

            $error\_msg = "Введите логин и пароль";

        }

        if( $error\_msg == "" ){

            session\_after\_login( $login, $password );

        }else{

            return $error\_msg;

        }

    }

    function check\_session(){

        if( $\_SESSION["username"] == "" && $\_SESSION["password"] == "" ){

            $\_SESSION["state"] = "login";

        }

    }

4. pielikuma turpinājums

    function check\_number( $state ){

        if( $state == 'task' || $state == 'support' ){

            $needed = 'o\_num';

        }

        if( $state == 'user' ){

            $needed = 'u';

        }

        $\_SESSION['state'] = 'error';

        header("Location: ".UI\_PUBLIC\_DIR."'\..\index.php");

    }

4. pielikuma turpinājums

    function get\_title(){

        switch( $\_SESSION['state'] ){

            case 'login':

                echo 'Вход в систему';

            break;

            case 'projects':

                echo 'Проекты';

            break;

            case 'project':

                echo 'Проект - ' . $\_SESSION['o\_num'];

            break;

            case 'event':

                echo 'Мероприятие - ' . $\_SESSION['o\_num'];

            break;

            case 'supports':

                echo 'Техподдержка';

            break;

            case 'new\_task':

                echo 'Новая задача';

            break;

            case 'tasks':

                echo 'Поручения';

            break;

            case 'task':

                echo 'Задача - ' . $\_SESSION['o\_num'];

            break;

            case 'users':

                echo 'Пользователи';

            break;

            case 'user':

                echo 'Пользователь';

            break;

            case 'error':

                echo 'Страница не найдена';

            break;

        }

    }

4. pielikuma turpinājums

    function session\_after\_login( $login, $password ){

        $\_SESSION["username"] = $login;

        $\_SESSION["password"] = $password;

        if( !isset( $\_SESSION['username'] )){

            $\_SESSION["state"] = "projects";

        }else{

            $\_SESSION["state"] = "login";

        }

        header("Location: index.php");

    }

    function get\_object\_color( $o\_id ){

        if( $o\_id == 5 || $o\_id == 6 ){

            $bootstrap\_color\_text = 'text-warning';

        }

        elseif( $o\_id == 1 || $o\_id == 2 || $o\_id == 3 or $o\_id == 4 ){

            $bootstrap\_color\_text = 'text-danger';

        }else{

            $bootstrap\_color\_text = 'text-primary';

        }

        return $bootstrap\_color\_text;

    }

4. pielikuma turpinājums

    function show\_page(){

        switch ( $\_SESSION["state"] ) {

            case "projects":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\projects.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "project":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\project.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "event":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\event.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "tasks":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\tasks.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "task":

                if( isset( $\_GET['task'] ) && !empty( $\_GET['task'] ) ){

                    $\_SESSION['o\_num'] = $\_GET['task'];

                }

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\task.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "supports":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\supports.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "support":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\support.php');

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

4. pielikuma turpinājums

            break;

            case "users":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\users.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "user":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\user.php');

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "new\_task":

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\new\_task.php');

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "change\_state":

                include( UI\_PUBLIC\_DIR.'\..\change\_state.php');

            break;

            case 'error':

                include( UI\_PUBLIC\_DIR.'\..\page\_details\navigation.php');

                include( UI\_PUBLIC\_DIR.'\..\error.php');

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php');

            break;

            case "exit":

                session\_destroy();

                header( "Location: /");

            break;

            default:

                include( UI\_PUBLIC\_DIR.'\..\login.php' );

                include( UI\_PUBLIC\_DIR.'\..\page\_details\footer.php' );

            break;

        }

    }

    function change\_state( $new\_state ){

        $\_SESSION["state"]  = $new\_state;

    }

    function fill\_field( $variable, $placeholder ){

        if( isset( $variable ) ){

            return 'value="'. $variable .'"';

4. pielikuma turpinājums

        }else{

            return 'placeholder="' . $placeholder . '"';

        }

    }

4. pielikuma turpinājums

    function get\_notes(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_za\_get\_notes\_names"></function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    function f\_za\_my\_tasks(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_za\_my\_tasks"></function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( $error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    function f\_za\_make\_note( $number, $text, $type, $time ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_za\_make\_note">

                            <arg name="note\_num">'. $number .'</arg>

                            <arg name="note\_text">'. $text     .'</arg>

                            <arg name="note\_type">'. $type     .'</arg>

                            <arg name="note\_time">'. $time     .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        return $response;

    }

4. pielikuma turpinājums

    function f\_za\_info( $number ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_za\_info">

                            <arg name="o\_num">'. $number .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            echo( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    // Выборка активный пользователей

    function f\_za\_select\_users(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_za\_select\_users"></function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    // Данные пользователя

    function f\_get\_user\_data( $user\_id ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_get\_user\_data">

                            <arg name="u\_id">'. $user\_id .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    // Выборка активный пользователей

    function f\_zs\_support\_list(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_zc\_support\_list">

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    function f\_zs\_project\_data( $number ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_zs\_project\_data">

                            <arg name="o\_num">'. $number .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( $error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    // Данные для техподдержки

    function f\_zs\_support\_data(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_zs\_support\_data">

                            <arg name="s\_num">'. $\_SESSION['o\_num'] .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    // Данные для техподдержки

    function f\_zs\_project\_list(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_zs\_project\_list">

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    // Данные мероприятия

    function f\_get\_event\_data( $number ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_get\_event\_data">

                            <arg name="o\_num">'. $number .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

4. pielikuma turpinājums

    function f\_za\_create( $manager, $performer, $name, $date\_start, $date\_end, $description ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_za\_create">

                            <arg name="o\_manager">'. $manager .'</arg>

                            <arg name="o\_worker">'. $performer .'</arg>

                            <arg name="o\_name">'. $name .'</arg>

                            <arg name="o\_date\_start">'. $date\_start .'</arg>

                            <arg name="o\_date\_end">'. $date\_end .'</arg>

                            <arg name="o\_description">'. $description .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        $error = $response\_xml->{'body'}->error;

        if( !empty( $error ) ){

4. pielikuma turpinājums

            die( 'API ERROR: '.$error );

        }

        return $response\_xml;

    }

    function base64\_to\_jpeg($base64\_string, $output\_file) {

        if( empty( (string)$base64\_string ) ){

            echo 'img\default\_avatar.png';

            return null;

        }

        // open the output file for writing

        $ifp = fopen( $output\_file, 'wb' );

        // split the string on commas

        // $data[ 0 ] == "data:image/png;base64"

        // $data[ 1 ] == <actual base64 string>

        // we could add validation here with ensuring count( $data ) > 1

        fwrite( $ifp, base64\_decode( (string)$base64\_string) );

        // clean up the file resource

        fclose( $ifp );

        echo $output\_file;

        return null;

    }

4. pielikuma turpinājums

         // Комментарии

         function f\_za\_get\_object\_notes(){

            $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

            $body = '<?xml version="1.0" encoding="UTF-8"?>

                    <sbapi>

                        <header>

                            <interface id="218120197" version="8" />

                            <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                            <error id="0" />

                            <auth pwd="open">'.$authdata.'</auth>

                        </header>

                        <body>

                            <function name="f\_za\_get\_object\_notes">

                                <arg name="o\_num">'. $\_SESSION['o\_num'] .'</arg>

                            </function>

                        </body>

                    </sbapi>';

            $header = array(

                "Content-type: text/xml",

                "Content-length: ".strlen( $body ),

                "Connection: close",

            );

            $ch = curl\_init();

            curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

            curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

            curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

            curl\_setopt($ch, CURLOPT\_POST, true);

            curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

            curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

            curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

            curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

            $response = curl\_exec( $ch );

            $errno    = curl\_errno( $ch );

            if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

            $response\_xml = simplexml\_load\_string($response);

            $error = $response\_xml->{'body'}->error;

            if( !empty( $error ) ){

                die( 'API ERROR: '.$error );

            }

            return $response\_xml;

        }

4. pielikuma turpinājums

    function new\_task\_validation( $manager, $worker, $name, $date\_start, $date\_end, $task ){

        if( !isset( $name ) || $name == "" ){

            return "Необходимо указать тему задачи.";

        }

        else if( !isset( $task ) || $task == "" ){

            return "Необходимо описать задачу.";

        }

        else if( !isset( $date\_start ) || $date\_start == ""  ){

            return "Необходимо заполнить дату начала выполнения задачи.";

        }

        else if( !isset( $date\_end ) || $date\_end == "" ){

            return "Необходимо заполнить дату начала окончания задачи.";

        }

        else if( !isset( $worker ) || $worker == "" ){

            return "Необходимо выбрать исполнителя.";

        }

        else if( !isset( $manager ) || $manager == "" ){

            return "Необходимо выбрать контралера.";

        }

        else if( $date\_end < $date\_start ){

            return "Дата окончания не может быть меньше даты начала";

        }else{

            return "Задача успешно заведена";

        }

    }

4. pielikuma turpinājums

    // Смена состояний

    function f\_change\_state( $object, $new\_state ){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_change\_state">

                            <arg name="o\_num">'. $object .'</arg>

                            <arg name="new\_state">'. $new\_state .'</arg>

                        </function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        return $response\_xml;

    }

4. pielikuma turpinājums

    // ID пользователя

    function f\_get\_my\_id(){

        $authdata = base64\_encode( '<authdata msg\_id="1" user="'.$\_SESSION['username'].'" password="'.$\_SESSION['password'].'" msg\_type="5000" user\_ip="127.0.0.1" />' );

        $body = '<?xml version="1.0" encoding="UTF-8"?>

                <sbapi>

                    <header>

                        <interface id="218120197" version="8" />

                        <message id="1" ignore\_id="yes" type="5000" created="2023-10-23T12:34:56Z"/>

                        <error id="0" />

                        <auth pwd="open">'.$authdata.'</auth>

                    </header>

                    <body>

                        <function name="f\_get\_my\_id"></function>

                    </body>

                </sbapi>';

        $header = array(

            "Content-type: text/xml",

            "Content-length: ".strlen( $body ),

            "Connection: close",

        );

        $ch = curl\_init();

        curl\_setopt($ch, CURLOPT\_URL, 'https://demo-api.simbase.eu');

        curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

        curl\_setopt($ch, CURLOPT\_TIMEOUT, 60);

        curl\_setopt($ch, CURLOPT\_POST, true);

        curl\_setopt($ch, CURLOPT\_POSTFIELDS, $body);

        curl\_setopt($ch, CURLOPT\_HTTPHEADER, $header);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

        curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, false);

        $response = curl\_exec( $ch );

        $errno    = curl\_errno( $ch );

        if($errno){die("Curl error! Error code: $errno; See more at https://curl.haxx.se/libcurl/c/libcurl-errors.html");}

        $response\_xml = simplexml\_load\_string($response);

        return $response\_xml;

    }

?>