

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**TĪMEKLĪ BĀZĒTA PROGRAMMATŪRAS  
PĀRVALDĪBAS SISTĒMA**

**KVALIFIKĀCIJAS DARBS**

Autors: **Kristaps Taube**

Stud. apl.: kt11023

Darba Vadītājs: LVU FMF lietišķā matem. Valdis Žuravļevs

RĪGA 2013

## **ANOTĀCIJA**

Šajā kvalifikācijas darbā ir aprakstīta Tīmeklī bāzētas programmatūras pārvaldības sistēmas izstrādes process. Sistēma sniedz iespēju ātri un viegli izveidot jaunu lietotnes versijas zaru un veikt šīs versijas pirmkoda kompilāciju, ar iespēju nogādāt lietotnes pakotni uz ārējiem serveriem izmantojot FTP protokolu un iespēju nosūtīt automatizētus e-pastus ar maināmu informāciju par kompilācijas veiksmi vai neveiksmi.

Darbs sevī iekļauj sistēmas biznesa prasības, funkcionalitātes un arhitektūras aprakstus.

## **ABSTRACT**

This qualification work describes Web-based software management System development process. System allows some trained person to quickly and easily create new application branch and build afterwards delivering created packages to remote servers using FTP protocol and sending differentiated emails depending if build succeeded or failed.

The paper contains descriptions of the system's business requirements, functionality and architecture.

## SATURA RĀDĪTĀJS

Ievads.....	8
1. Programmatūras prasību specifikācija .....	9
1.1. Ievads .....	9
1.1.1. Nolūks .....	9
1.1.2. Darbības sfēra.....	9
1.1.3. Definīcijas .....	9
1.1.4. Saistība ar citām nodaļām un dokumentiem .....	10
1.2. Vispārējs apraksts .....	10
1.2.1. Produkta perspektīva .....	10
1.2.2. Produkta funkcijas.....	10
1.2.3. Lietotāju raksturiezīmes .....	11
1.2.4. Vispārējie ierobežojumi .....	11
1.2.5. Pieņēmumi un atkarības .....	11
1.3. Funkcionālās prasības .....	11
1.3.1. Jauna lietotāja izveidošana .....	11
1.3.2. Pieslēgšanās.....	12
1.3.3. Atslēgšanās.....	12
1.3.4. Apskatīt moduļa zarus .....	13
1.3.5. Izveidot jaunu zaru .....	13
1.3.6. Izveidot jaunu sistēmas pakotni no jau esoša zara .....	14
1.3.7. Pārsūtīt izveidoto pakotni uz ārējo serveri .....	15
1.3.8. Izsūtīt e-pastu .....	15
1.3.9. Pakotņu lejuplāde .....	16
1.3.10. Konfigurācijas iestatīšana.....	17
1.3.11. Iestatīt e-pasta tekstu. ....	17
1.3.12. Pievienot e-pasta adresi .....	18
1.3.13. Izdzēst e-pasta adresi .....	18

1.3.14.	Pievienot jaunu FTP adresi.....	19
1.3.15.	Izdzēst FTP adresi .....	19
1.3.16.	Apturēt komandu izpildi.....	20
1.3.17.	Konfigurācijas failu rediģēšana .....	20
1.4.	Datu plūsmu diagrammas .....	21
1.4.1.	Līmenis 0.....	21
1.4.2.	Līmenis 1 .....	22
1.4.3.	Līmenis 2.....	23
1.5.	Arējās saskarnes.....	24
1.5.1.	Lietotāju saskarne.....	24
1.5.2.	Programmatūras saskarne.....	24
1.6.	Nefunkcionālās prasības .....	24
1.6.1.	Veiktspējas prasības .....	24
1.6.2.	Drošība .....	25
2.	Programmatūras projektējuma apraksts .....	26
2.1.	Ievads .....	26
2.1.1.	Nolūks .....	26
2.1.2.	Darbības sfēra.....	26
2.1.3.	Definīcijas .....	26
2.2.	Saistība ar citām nodaļām un dokumentiem .....	26
2.3.	Dekompozīcijas apraksts .....	26
2.4.	Sistēmas moduļu apraksts .....	28
2.5.	Datu projektējums.....	28
2.5.1.	Datu projektējuma apraksts .....	28
2.6.	Datu dekompozīcija .....	29
2.6.1.	Tabula „User”.....	29
2.6.2.	Tabula „Group” .....	30
2.6.3.	Tabula „Configuration” .....	30

2.6.4.	Tabula „Email_recipient” .....	31
2.6.5.	Tabula „Ftp_recipient” .....	32
2.6.6.	Tabula „History” .....	32
3.	Testēšanas dokumentācija .....	34
3.1.	Testēšanas metodika .....	34
3.2.	Testēšanas žurnāls.....	34
3.2.1.	Lietotāja pieteikšanās un atteikšanās.....	34
3.2.2.	Sistēmas konfigurācijas labošana.....	34
3.2.3.	CVS zaru veidošana .....	35
3.2.4.	Lietotnes pakotnes veidošana.....	35
3.3.	Testēšanas rezultāti .....	36
4.	Projekta organizācija .....	37
5.	Kvalitātes nodrošināšana.....	38
6.	Konfigurācijas pārvaldība .....	39
7.	Darbietilpības novērtējums .....	40
7.1.	COCOMO modeļa darbietilpības aprēķināšana .....	40
	Secinājumi .....	41
	Izmantotā literatūra un avoti .....	42
	Pielikumi.....	43
	Pielikums 1. ....	43
	Pielikums 2. ....	45
	Programmatūras pirmkoda paraugs. ....	45

## APZĪMĒJUMI

**MVC**(Model-View-Controller) – Modeļa-skata-kontroliera šablons.

**CVS**(Concurrent Versions System) – vien no versiju kontroles rīkiem.

**Javascript** – klienta pusē darbināma programmēšanas valoda, kas ļauj pārlūka programmā dinamiski rediģēt DOM koka datus.

**FTP**(File Trasfere Protocol) – failu pārvietošanas protokols.

**Ūdenskrituma modelis** – projektu plānošanas modelis, kurš ir secīgs, lejupejošs, ar skaidrām un strikti atdalītām fāzēm.

**HTML** (HyperText Markup Language) – iezīmēšanas valoda, kas ir izstrādāta tīmekļa lappušu un citas pārlūkprogrammā attēlojamās informācijas glabāšanai.

## IEVADS

Programmatūras uzturēšanas procesa laikā programmas pirmkodā tiek veiktas dažādas izmaiņas, kā arī ieviestas no klientiem saņemtās jaunās lietotnes iespējas. Lai pārvaldītu programmas pirmkodā veiktās izmaiņas, tiek lietota versiju kontroles sistēma. Lai pārliecinātos, ka sistēmā nav ieviesusies regresija, vai, ka jaunās iespējas savas funkcijas pilda pareizi, veiktās izmaiņas ir jānogādā testētājiem, kur sistēmai tiks veikti integrācijas, regresijas un veiktspējas testi. Šīs darbības izstrādātājam var aizņemt daudz laika, tāpēc ir nepieciešams tās automatizēt.

Šis kvalifikācijas darbs apraksta tīmeklī bāzētas programmatūras pārvaldības sistēmas izstrādes procesus. Sistēmai ir jābūt spējīgai ērti un ātri izveidot jaunus versiju zarus norādītā CVS repositoriņā, palaist sistēmas kompilācijas lietotni un nogādāt nokompilētu sistēmas pakotni uz norādītiem ārējiem serveriem izmantojot FTP protokolu.

Sistēmas izveidei tika izmantota Python 2.7, HTML, Javascript valoda izmantojot jQuery bibliotēku, Twitter Bootstrap bibliotēka, CSS. Sistēma tika izstrādāta izmantojot Flask 0.9 ietvaru un tā paplašinājumus (flask-login, Flask-SQLAlchemy, Flask-WTF). Papildus tika izmantots Apache serveris ar mod\_wsgi moduli un SQLite datubāze.

Darba mērķa sasniegšanai tika izvēlēti šādi uzdevumi:

- Iepazīties ar programmatūras pārvaldībai nepieciešamajām funkcijām;
- Iepazīties ar programmēšanas valodu Python un Flask ietvaru;
- Izveidot programmaprodukta prasību specifikāciju un atbilstoši tai veikt programmaprodukta projektēšanu;
- Izstrādāt programmatūras produktu;
- Veikt programmatūras vienībtestēšanu un sagatavot pavadošo dokumentāciju.

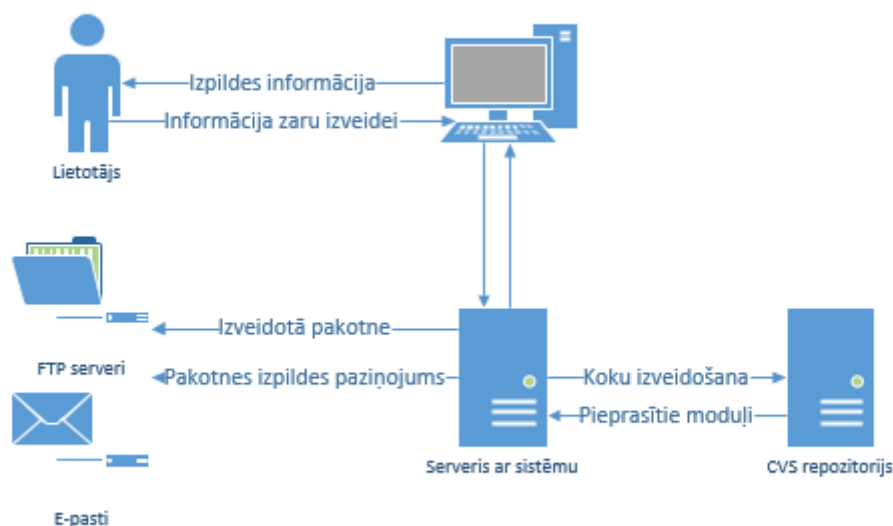
# 1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

## 1.1. Ievads

### 1.1.1. Nolūks

Programmatūras prasību specifikācijas (PPS) nolūks ir precīzi un viennozīmīgi aprakstīt tīmeklī bāzētas programmatūras pārvaldības sistēmas prasības. Dokuments paredzēts programmatūras izstrādātājiem, kuri veiks programmatūras projektējuma apraksta (PPA) izstrādāšanu.

### 1.1.2. Darbības sfēra



#### *Att 1. Lietotnes jauna versijas zara izveide un kompilācija*

Programmatūras prasību specifikācija apraksta prasības tīmeklī bāzētas programmatūras pārvaldības sistēmai.

Sistēma ir paredzēta sistēmas versiju un izmaiņu procesu automatizācijai. Lietotājs izmantojot tīmekļa vietni var izveidot jaunu sistēmas zaru CVS repozitorijā, kas reprezentē jaunu versiju, izveidot sistēmas pakotni un nogādāt to uz norādītajiem FTP serveriem.

### 1.1.3. Definīcijas

**Apmeklētājs** – tīmekļa vietnes apmeklētāji, kuri nav pieslēgušies sistēmai.

**Reģistrēts lietotājs** – lietotājs kurš ir reģistrēts sistēmā un ir pieslēdzies ar savu lietotājvārdu un paroli.

**Administrators** - reģistrēts lietotājs ar īpašam tiesībām.

**Sesija** – laika intervāls no sistēmas pirmā apmeklējuma līdz pārlūkprogrammas aizvēršanas.

#### **1.1.4. Saistība ar citām nodaļām un dokumentiem**

Dokumenta noformēšanai un struktūras izveidei tika izmantots standarts LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”. Programmēšanas prasību specifikācija ir „Tīmeklī bāzēta programmatūras pārvaldības sistēma” sastāvdaļa.

### **1.2. Vispārējs apraksts**

#### **1.2.1. Produkta perspektīva**

Izstrādājamais produkts ir pirmkoda sistēma. Tā ir neatkarīga sistēma un var tikt izmantota kā daļa no lielākas sistēmas.

#### **1.2.2. Produkta funkcijas**

Sistēma nodrošina sekojošu funkcionalitāti:

- Jauna lietotāja izveidošana,
- Pieslēgšanās,
- Iespēja lietotājam iepazīties ar CVS repozitorijā esošajiem moduļa zariem,
- Iespēja CVS repozitorijā izveidot jaunu zaru,
- Iespēja no CVS repozitorija izņemt moduli un to kompilēt izveidojot lietotnes pakotni,
- Iespēja nosūtīt nokompilētās lietotnes pakotni uz ārējiem serveriem izmantojot FTP protokolu,
- Automātiska (jaunākās versijas) sistēmas kompilācijai nepieciešamo pakotņu lejuplāde izmantojot FTP protokolu,
- Informācijas e-pastu nosūtīšana ar lietotnes kompilācijas informāciju,
- Komandu priekšskatīšana pirms to izpildes uz servera,
- Iespēja priekšskatīt un labot konfigurācijas failus pirms komandu izpildes.
- Saglabāt sistēmas vēsturē informāciju par kompilāciju.

### 1.2.3. Lietotāju raksturiezīmes

Sistēmas lietotāji ir iedalīti 3 grupās – administratori, reģistrētie lietotāji un apmeklētāji. Administratoriem un reģistrētiem lietotājiem ir jābūt pamatzināšanām par darbu ar datoru, kā arī lai pilnvērtīgi izmantotu šo sistēmu – pamatzināšanām par versiju kontroli.

### 1.2.4. Vispārējie ierobežojumi

Sistēmas darbināšanai jābūt bez instalēšanas, izmantojot tīmekļa pārlūkprogrammu, tāpēc sistēmai ir jāatbalsta visas populārākās pārlūkprogrammu versijas. Lietotājam jāvar pieslēgties tīmekļa vietnei no visām populārākajām platformām (Windows, Linux, Mac).

### 1.2.5. Pieņēmumi un atkarības

Gadījumā, ja kāds no pakotņu lejuplādei vai augšuplādei nepieciešamajiem serveriem kļūst nepieejams, sistēmai ir jāizmanto kāds no spoguļserveriem.

## 1.3. Funkcionālās prasības

### 1.3.1. Jauna lietotāja izveidošana

<b>Ievads</b>		
Šīs funkcijas mērķis ir izveidot jaunu sistēmas lietotāju		
<b>Pieejas tiesības</b>	Administrators	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
Lietotājvārds	Jā	Teksts no 2 līdz 20 simboliem
Parole	Jā	Teksts no 6 līdz 60 simboliem
Lietotāja grupa	Jā	Grupa no saraksta
<b>Apstrāde</b>		
1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.		
<b>Izvaddati</b>		

Nav
<b>Kļūdu ziņojumi</b>
1. Šis lauks ir obligāts

### 1.3.2. Pieslēgšanās

<b>Ievads</b>		
Šīs funkcijas mērķis ir identificēt lietotāju		
<b>Pieejas tiesības</b>	Apmeklētājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
Lietotājvārds	Jā	Teksts no 2 līdz 20 simboliem
Parole	Jā	Teksts no 6 līdz 60 simboliem
<b>Apstrāde</b>		
<ol style="list-style-type: none"> <li>1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.</li> <li>2. Ja dati ievadīti korekti, tad tiks meklēts lietotājs. Ja lietotājs netiks atrasts tad virs formas parādīsies kļūdu ziņojums nr. 2.</li> <li>3. Ja lietotājs ir atrasts, tad tiks salīdzināta lietotāja parole ar norādīto paroli. Ja paroles nesakrīt tiek parādīts ziņojums nr. 2.</li> </ol>		
<b>Izvaddati</b>		
Tiek piedāvāta iespēja atslēgties.		
<b>Kļūdu ziņojumi</b>		
<ol style="list-style-type: none"> <li>1. Šis lauks ir obligāts.</li> <li>2. Lietotājvārds vai parole ir norādīti nepareizi.</li> </ol>		

### 1.3.3. Atslēgšanās

<b>Ievads</b>	
Šīs funkcijas mērķis ir atslēgt lietotāju no sistēmas.	
<b>Pieejas tiesības</b>	Administrators, lietotājs

<b>Ievade</b>
Nav
<b>Apstrāde</b>
No sesijas tiek izdzēst lietotāja identifikators.
<b>Izvaddati</b>
Nav
<b>Kļūdu ziņojumi</b>
Nav

#### 1.3.4. Apskatīt moduļa zarus

<b>Ievads</b>		
Šīs funkcijas mērķis ir atgriezt sarakstu ar CVS repozitorijā esošajiem moduļa zariem.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
Modulis	Jā	Teksts no 2 līdz 100 simboliem
<b>Apstrāde</b>		
1. No CVS repozitorija atgriež sarakstu ar moduļa zariem. Ja tādu nav atgriež tukšu sarakstu.		
<b>Izvaddati</b>		
Saraksts ar pieejamajiem zariem		
<b>Kļūdu ziņojumi</b>		
Nav		

#### 1.3.5. Izveidot jaunu zaru

<b>Ievads</b>		
Šīs funkcijas mērķis ir CVS repozitorija norādītajam modulim izveidot jaunu versijas zaru.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>

Zars no	Jā	Zars no saraksta
Modulis	Jā	Teksts no 1 līdz 60 simboliem
Jaunā zara nosaukums	Jā	Teksts no 1 līdz 60 simboliem
Sistēmas apakšversijas numurs	Nē	Naturāls skaitlis
Datums	Nē	Datuma formāts
Laiks	Nē	Laika formāts
Zara tips	Jā	Tips no izvēlnes
Izveidot pakotni	Nē	Izvēles rūtiņa
<b>Apstrāde</b>		
1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.		
<b>Izvad dati</b>		
Nav		
<b>Kļūdu ziņojumi</b>		
1. Šis lauks ir obligāts.		

### 1.3.6. Izveidot jaunu sistēmas pakotni no jau esoša zara

<b>Ievads</b>		
Šīs funkcijas mērķis ir no CVS repozitorijā esošā zara izveidot sistēmas pakotni.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
Zars no	Jā	Zars no saraksta
Modulis	Jā	Teksts no 1 līdz 60 simboliem
Datums	Nē	Datuma formāts
Laiks	Nē	Laika formāts
Zara tips	Jā	Tips no izvēlnes

<b>Apstrāde</b>
1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.
<b>Izvaddati</b>
Nav
<b>Kļūdu ziņojumi</b>
1. Šis lauks ir obligāts.

### 1.3.7. Pārsūtīt izveidoto pakotni uz ārējo serveri

<b>Ievads</b>		
Šīs funkcijas mērķis pārsūtīt izveidoto pakotni uz ārēju serveri.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
FTP adrese	Jā	Teksts no 1 līdz 200 simboliem
<b>Apstrāde</b>		
1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.		
2. Ja ievadītā FTP adrese neatbilst ftp://<lietotājvārs>:<parole>@<ip adrese>/<ceļš>, tad tiks parādīts 1. kļūdu ziņojums.		
<b>Izvaddati</b>		
Nav		
<b>Kļūdu ziņojumi</b>		
1. Šis lauks ir obligāts.		
2. FTP adrese neatbilst ierobežojumiem! Adrese: "<adrese>"		

### 1.3.8. Izsūtīt e-pastu

<b>Ievads</b>	
Šīs funkcijas mērķis izsūtīt e-pastu uz norādītajām adresēm.	
<b>Pieejas tiesības</b>	Administrators, lietotājs



### 1.3.10. Konfigurācijas iestatīšana

Ievads		
Šīs funkcijas mērķis ir iestatīt servisa konfigurāciju.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
Ievade		
Laukuma nosaukums	Obligāts	Tips/ierobežojumi
CVSROOT	Jā	Teksts ar papildus validāciju
Build directory	Jā	Teksts, kas atbilst Unix stila direktorijas konvencijai
FTP delivery directory	Jā	Teksts, kas atbilst Unix stila direktorijas konvencijai
Autobuilder tmp dir	Jā	Teksts, kas atbilst Unix stila direktorijas konvencijai
Build configuration file directory	Jā	Teksts, kas atbilst Unix stila direktorijas konvencijai
Build script	Jā	Teksts, kas atbilst Unix stila direktorijas konvencijai
Apstrāde		
<ol style="list-style-type: none"><li>1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.</li><li>2. Ja lauka satur neatbilst tā ierobežojumam, tad blakus laukam parādīsies kļūdas ziņojums nr. 2.</li></ol>		
Izvaddati		
Nav		
Kļūdu ziņojumi		
<ol style="list-style-type: none"><li>1. Šis lauks ir obligāts!</li><li>2. Šāds ceļš failu sistēmā neatrodas!</li></ol>		

### 1.3.11. Iestatīt e-pasta tekstu.

Ievads
Šīs funkcijas mērķis ir nodrošināt iespēju iestatīt tekstu, kas tiks sūtīts e-pastā.

<b>Pieejas tiesības</b>		Administrators, lietotājs
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
E-pasta teksts	Jā	Teksts
<b>Apstrāde</b>		
1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas ziņojums nr. 1.		
<b>Izvaddati</b>		
Nav		
<b>Kļūdu ziņojumi</b>		
1. Šis lauks ir obligāts.		

### 1.3.12. Pievienot e-pasta adresi

<b>Ievads</b>		
Šīs funkcijas mērķis ir nodrošināt iespēju pievienot e-pasta adresi.		
<b>Pieejas tiesības</b>		Administrators, lietotājs
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
E-pasta teksts	Jā	E-pasta formāts
<b>Apstrāde</b>		
1. Ja ievadītā e-pasta adrese neatbilst formātam, tiek parādīts 1. kļūdu ziņojums		
<b>Izvaddati</b>		
Nav		
<b>Kļūdu ziņojumi</b>		
1. Ievadītais e-pasts neatbilst formātam.		

### 1.3.13. Izdzēst e-pasta adresi

<b>Ievads</b>		
Šīs funkcijas mērķis ir nodrošināt iespēju izdzēst e-pasta adresi no saraksta.		

<b>Pieejas tiesības</b>	Administrators, lietotājs
<b>Ievade</b>	
Ievaddatu nav.	
<b>Apstrāde</b>	
Nav.	
<b>Izvaddati</b>	
Nav.	
<b>Kļūdu ziņojumi</b>	
Nav.	

### 1.3.14. Pievienot jaunu FTP adresi

<b>Ievads</b>		
Šīs funkcijas mērķis ir pievienot FTP adresi.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
FTP adrese	Jā	FTP adreses formāts
<b>Apstrāde</b>		
1. Ja ievadītā FTP adrese neatbilst <a href="#">ftp://&lt;lietotājvārs&gt;:&lt;parole&gt;@&lt;ip adrese&gt;/&lt;ceļš&gt;</a> , tad tiks parādīts 1. kļūdu ziņojums.		
<b>Izvaddati</b>		
Nav		
<b>Kļūdu ziņojumi</b>		
1. FTP adrese neatbilst ierobežojumiem! Adrese: "<adrese>"		

### 1.3.15. Izdzēst FTP adresi

<b>Ievads</b>	
Šīs funkcijas mērķis ir nodrošināt iespēju izdzēst FTP adresi no saraksta.	
<b>Pieejas tiesības</b>	Administrators, lietotājs
<b>Ievade</b>	

Ievaddatu nav.
<b>Apstrāde</b>
Nav.
<b>Izvaddati</b>
Nav.
<b>Kļūdu ziņojumi</b>
Nav.

### 1.3.16. Apturēt komandu izpildi

<b>Ievads</b>	
Apturēt komandu izpildi.	
<b>Pieejas tiesības</b>	Administrators, lietotājs
<b>Ievade</b>	
Nav	
<b>Apstrāde</b>	
Nav	
<b>Izvaddati</b>	
Nav	
<b>Kļūdu ziņojumi</b>	
Nav	

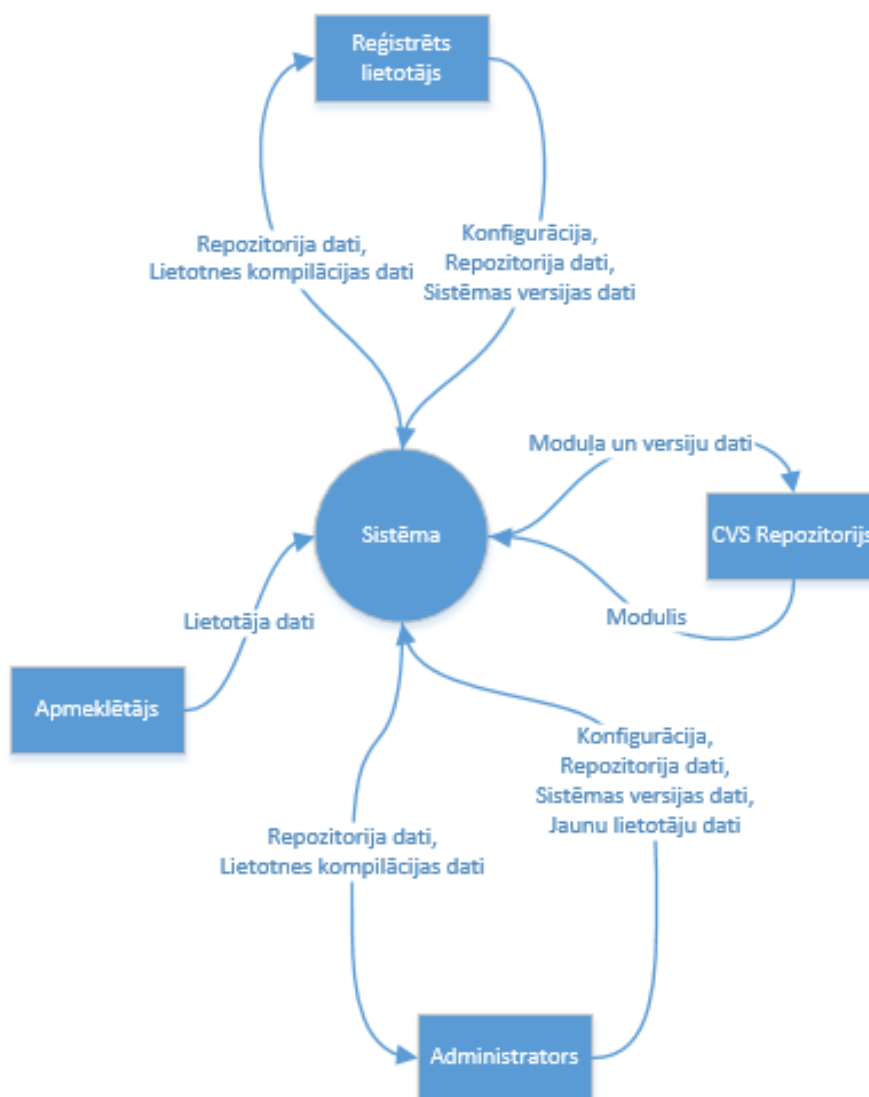
### 1.3.17. Konfigurācijas failu rediģēšana

<b>Ievads</b>		
Šī funkcija ļauj lietotājam priekšskatīt un rediģēt konfigurācijas failus pirms to izmantošanas.		
<b>Pieejas tiesības</b>	Administrators, lietotājs	
<b>Ievade</b>		
<b>Laukuma nosaukums</b>	<b>Obligāts</b>	<b>Tips/ierobežojumi</b>
Teksts	Jā	Teksts
<b>Apstrāde</b>		
1. Ja kāds no obligātajiem laukiem nav norādīts, blakus laukam parādīsies kļūdas		

ziņojums nr. 1.
<b>Izvaddati</b>
Nav
<b>Kļūdu ziņojumi</b>
1. Šis lauks ir obligāts.

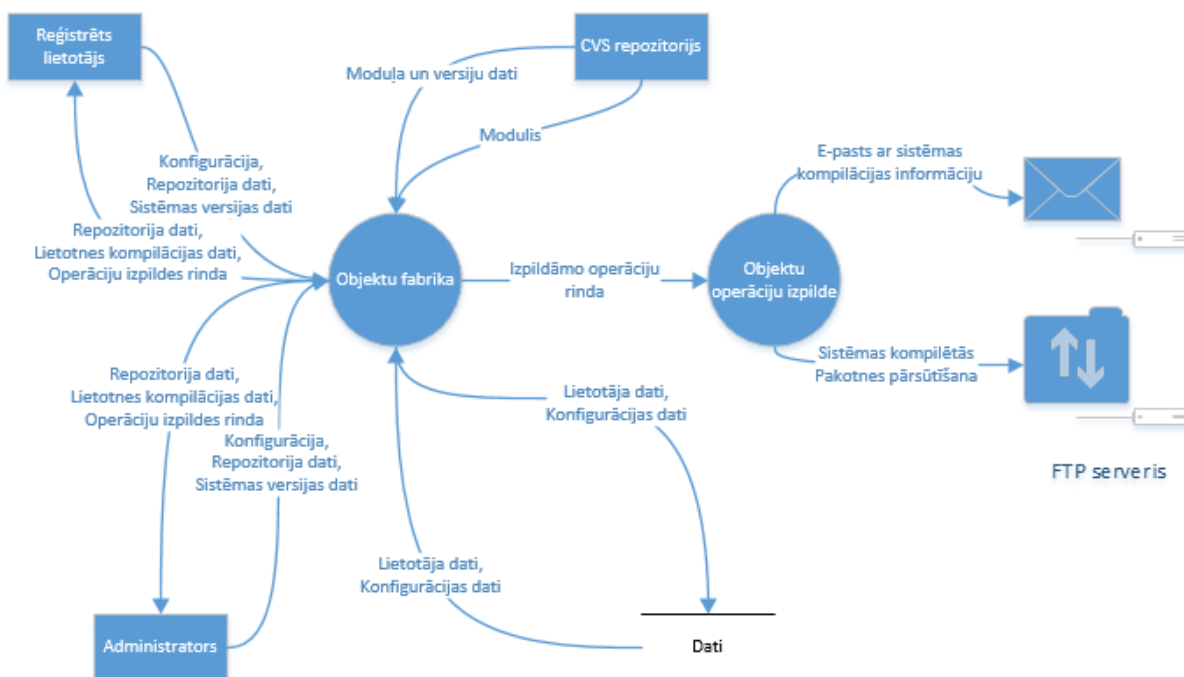
## 1.4. Datu plūsmu diagrammas

### 1.4.1. Līmenis 0

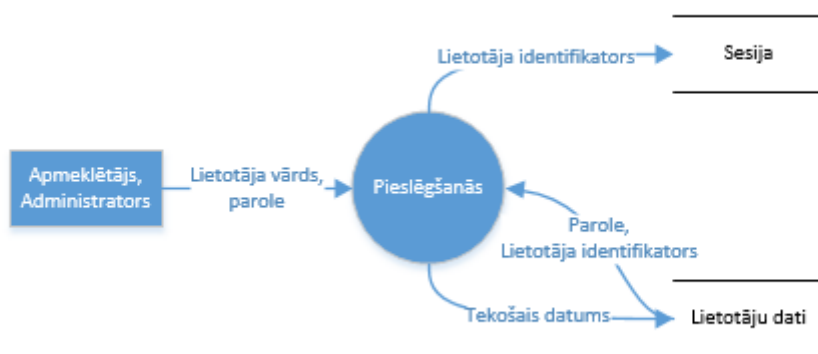


*Att. 1.1 Konteksta līmenis – sistēmas kopskats*

## 1.4.2. Līmenis 1



Att. 1.2 Līmenis 1 – sistēmas sadalīšana apakšsistēmās

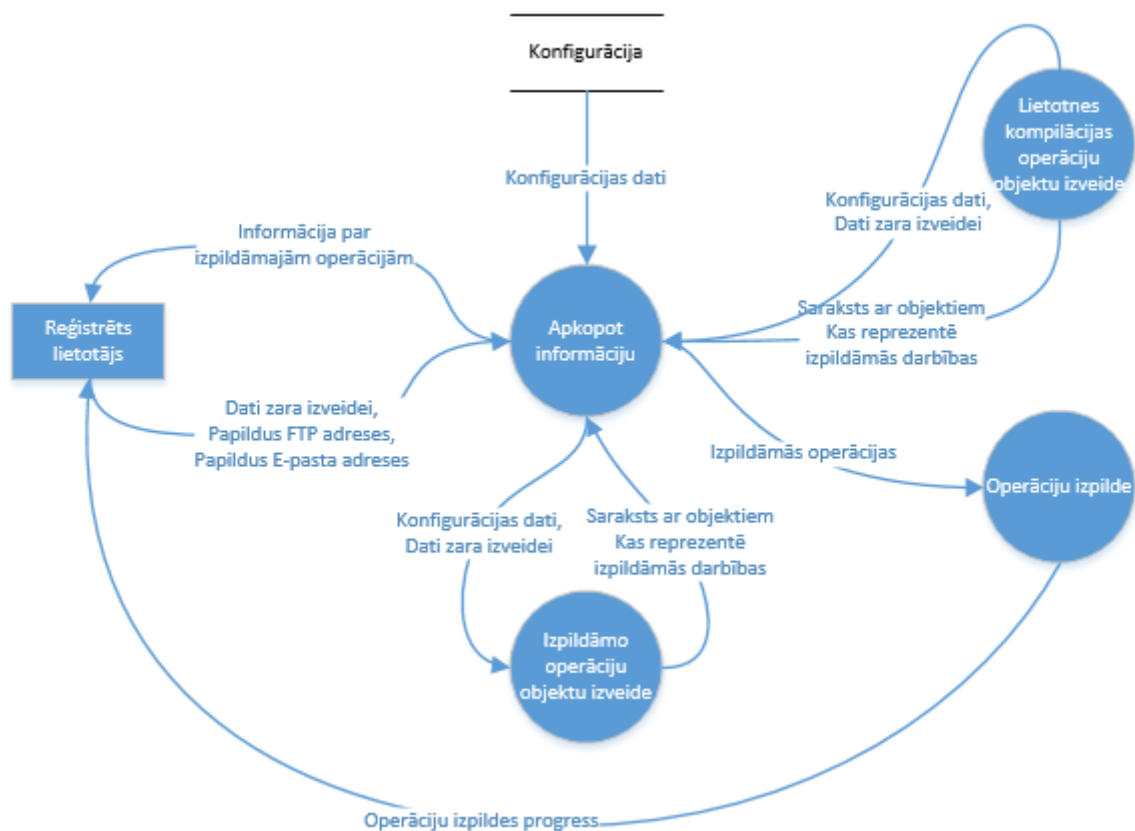


Att. 1.3 Lietotāju pieslēgšanās

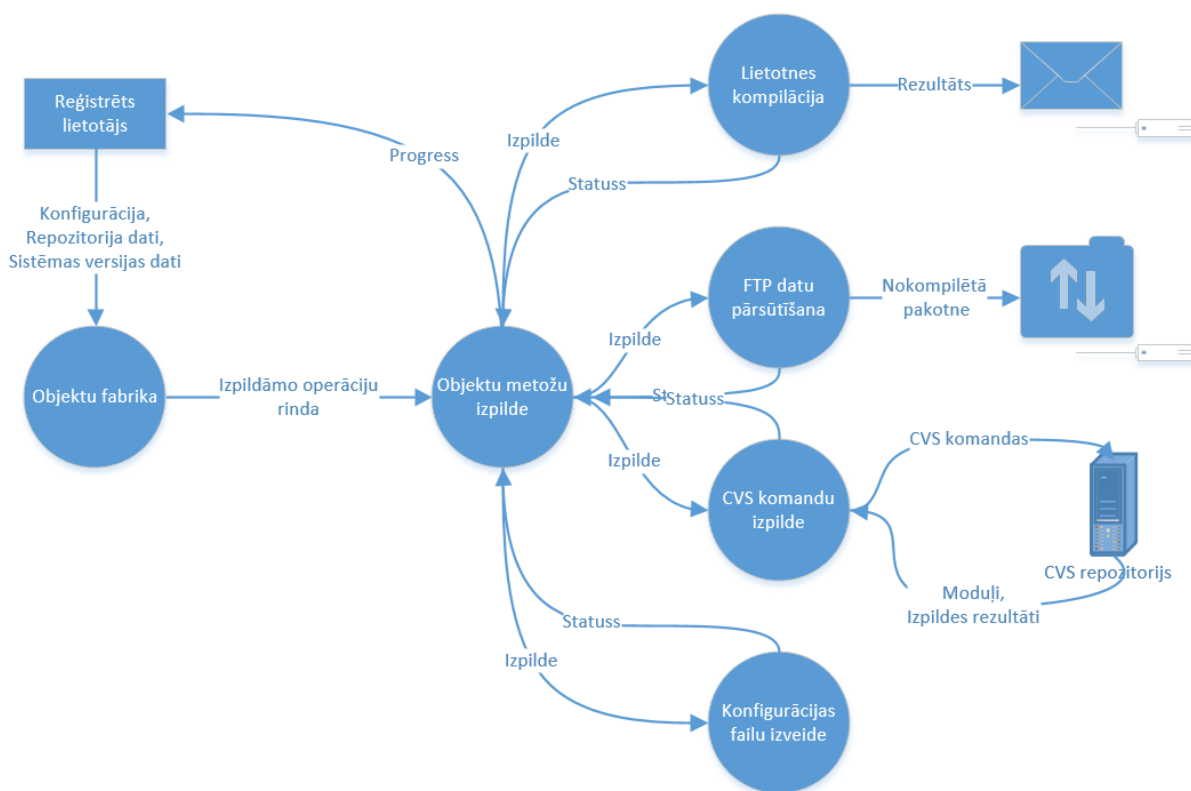


*Att. 1.4 Lietotāja reģistrācija*

### 1.4.3. Līmenis 2



*Att. 1.5 Objektu izveides modulis*



*Att. 1.6 Objektu izpildes modulis*

## 1.5. Arējās saskarnes

### 1.5.1. Lietotāju saskarne

Tīmeklī bāzētajai programmatūras pārvaldības sistēmai ir jābūt lietojamai tīmekļa vidē. Apmeklētāju saskarnei jāatbilst 1. pielikumā pievienotajam pielikumam. Jānodrošina lai sistēma ir pieejama angļu valodā.

### 1.5.2. Programmatūras saskarne

- Pārlūkprogramma: Internet Explorer 8+, Firefox 3.6+, Chrome 10+, Safari 5+, Opera 11+.
- Tīmekļa programmai jābūt aktivizētai Javascript atbalstam.

## 1.6. Nefunkcionālās prasības

### 1.6.1. Veiktspējas prasības

Sistēmai jāspēj nodrošināt vismaz 10 lietotāju vienlaicīgu sistēmas lietošanu. Katras lapas ielādes ātrumam jābūt ne ilgākam par 10 sekundēm.

## **1.6.2. Drošība**

Katra sistēmas lietotāja parole ir jāšifrē, lai nodrošinātos pret datu zagšanu. Katra lietotāja paroles sarežģītība ir mākslīgi jāpalielina. Jānodrošinās pret SQL un HTML injekcijām un automātiskas informācijas ievadi. Katram lietotāja tipam ir jānodrošina pieeja tikai tām tīmekļa vietnes daļām uz kurām lietotājam ir pieejas tiesības.

## **2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS**

### **2.1. Ievads**

#### **2.1.1. Nolūks**

Programmatūras projektējuma apraksta (PPA) nolūks ir parādīt, kā programmatūras prasību specifikācijā aprakstītās programmatūras prasības tiks realizētas sistēmā.

PPA veidots atbilstoši MVC projektēšanas šablonam, Python 2.7 programmēšanas valodai, izmantojot Flask 0.9 ietvaru un SQLite 2.8 datubāzes pārvaldīšanas sistēmai. Šis dokuments ir paredzēts sistēmas izstrādātājiem.

#### **2.1.2. Darbības sfēra**

Sistēma ir paredzēta lietotņu konfigurācijas pārvaldības automatizēšanai. Lietotājam tiek dota iespēja izvēlēties moduļus un to versijas no norādītās CVS repozitorija un no dotās informācijas sistēma izveidos jaunus versiju zarus un kompilēs lietotni, nogādājot iegūto pakotni uz norādītajiem FTP serveriem.

#### **2.1.3. Definīcijas**

### **2.2. Saistība ar citām nodaļām un dokumentiem**

Sistēmas projektējumā jārealizē „Tīmeklī bāzēta programmatūras pārvaldības sistēma” prasību specifikācijā aprakstītās prasības. Dokumenta noformēšanai un struktūras izveidei tika izmantots standarts LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

### **2.3. Dekompozīcijas apraksts**

„Tīmeklī bāzēta programmatūras pārvaldības sistēma” tiks izstrādāta izmantojot Python programmēšanas valodu un Flask ietvaru. Tiks izmantoti arī Flask ietvara paplašinājumi:

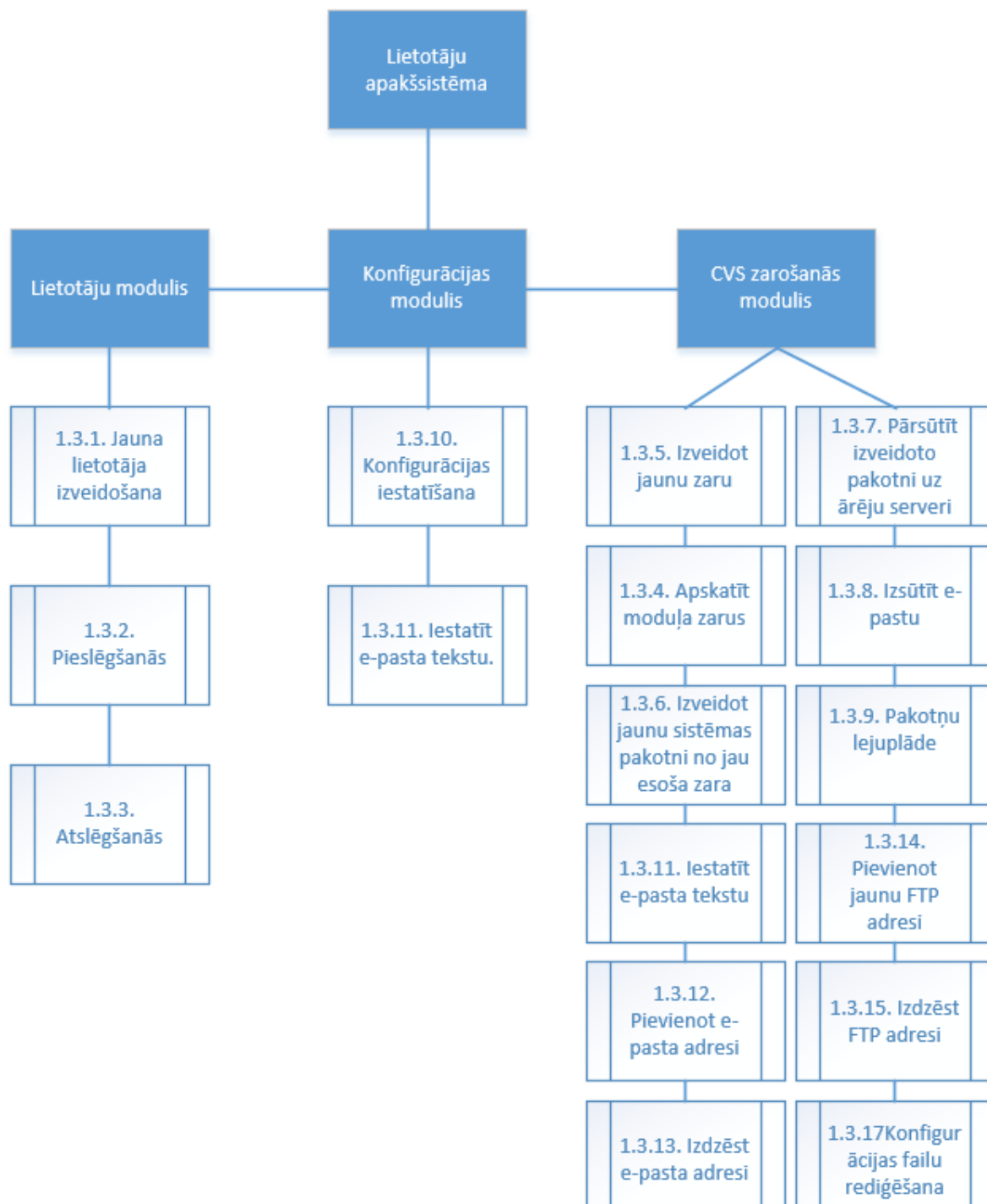
- flask-login – nodrošina lietotāja sesiju administrēšanu,
- Flask-SQLAlchemy – pievieno atbalstu SQLAlchemy bibliotēkai, kas ir Python SQL rīkkopa un ORM, kas izstrādātājiem dot pilnu spēku un pielāgojamību ko nodrošina SQL.

- Flask-WTF – pievieno augsta līmeņa funkcionalitāti darbībām ar tīmekļa formām.

Sistēmā tiek izmantotas arī papildus bibliotēkas:

- Simplejson – ātrs un vienkāršs Python JSON objektu kodēšanu un atkodēšanu.
- SQLAlchemy - Python SQL rīkkopa un ORM, kas izstrādātājiem dot pilnu spēku un pielāgojamību ko nodrošina SQL.

## 2.4. Sistēmas moduļu apraksts

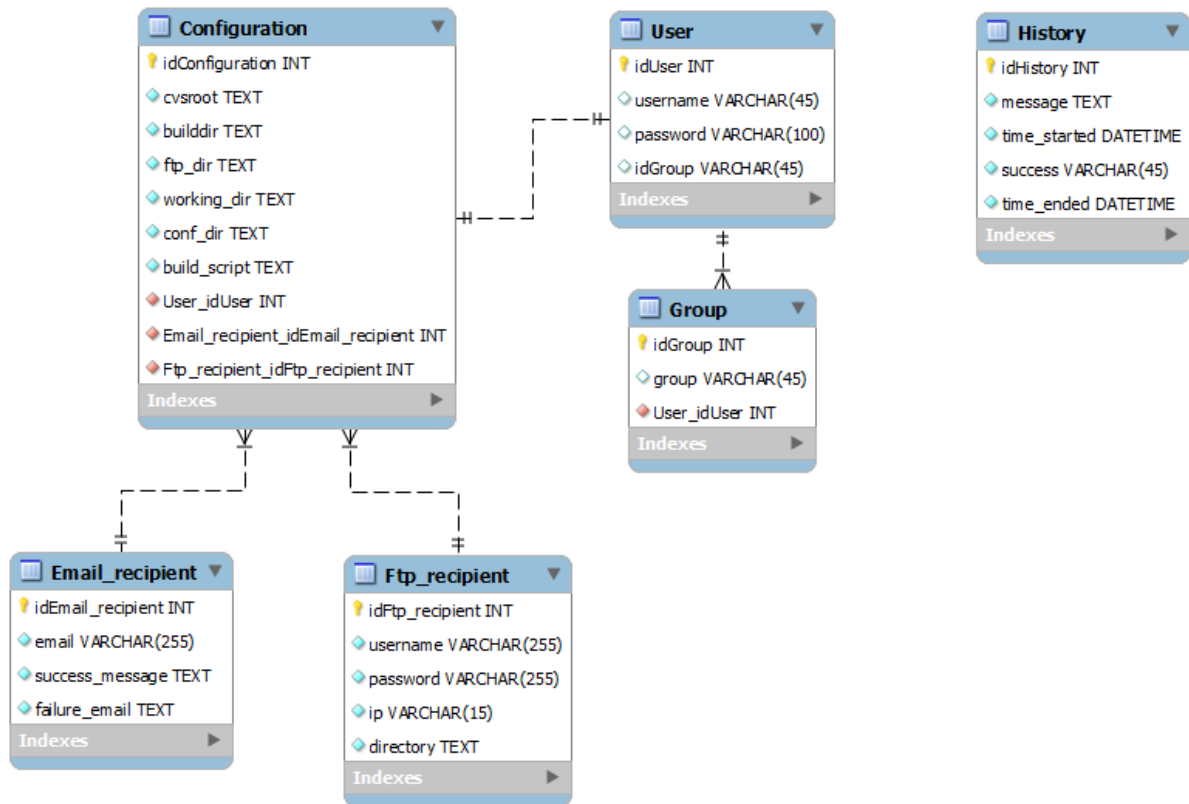


Att. 2.1 Sistēmas moduļi un to saistība ar funkcionālām prasībām

## 2.5. Datu projektējums

### 2.5.1. Datu projektējuma apraksts

Datu atkarības attēlotas ER modelī (Att. 2.2.).



Att. 2.2 ER modelis

## 2.6. Datu dekompozīcija

### 2.6.1. Tabula „User”

Šīs tabulas uzdevums ir glabāt datus par reģistrētiem lietotājiem.

Lauka nosaukums	Datu tips	Obligāts	Atslēga	Noklusētā vērtība	Apraksts
idUser	INT	Jā	PK		Automātiski ģenerēts lietotāja identifikators
username	VARCHAR(45)	Jā			Lietotājevārds
password	VARCHAR(100)	Jā			Parole(parole + salt hašs, šifrēta izmantojot MD5 algoritmu)
group_id	INT	Jā			Grupas identifikators

## 2.6.2. Tabula „Group”

Tabula „Group” glabā administratora izveidotās grupas ar lietotājiem.

Lauka nosaukums	Datu tips	Obligāts	Atslēga	Noklusētā vērtība	Apraksts
idGroup	INT	Jā	PK		Automātiski ģenerēts grupas identifikators
group	VARCHAR(45)	Jā			Grupās nosaukums
user_id	INT	Jā	FK		Lietotāja identifikators. Arējā atslēga uz tabulu „User” (2.6.1.)

## 2.6.3. Tabula „Configuration”

Tabulas „Configuration” uzdevums ir glabāt lietotāja iestatītos konfigurācijas datus.

Lauka nosaukums	Datu tips	Obligāts	Atslēga	Noklusētā vērtība	Apraksts
idConfiguration	INT	Jā	PK		Automātiski ģenerēts konfigurācijas identifikators
cvsroot	TEXT	Jā			CVS sakne
builddir	TEXT	Jā			Ceļš uz direktoriju, kur tiek kompilēta lietotne.
ftp_dir	TEXT	Jā			Ceļš uz direktoriju kur atrodas ar FTP sūtāmā pakotne.
working_dir	TEXT	Jā			Ceļš uz lietotnes galveno direktoriju.
conf_dir	TEXT	Jā			Ceļš uz direktoriju, kur atrodas lietotnes

					kompilācijai nepieciešamie konfigurācijas faili.
build_script	TEXT	Jā			Ceļš uz skriptu, ko palaižot tiks kompilēta lietotne.
user_id	INT	Jā	FK		Lietotāja identifikators. Arējā atslēga uz tabulu „User” (2.6.1.)
email_id	INT	Jā	FK		E-pasta identifikators. Ārējā atslēga uz tabulu „Email_recipient”(2.6.4)
ftp_id	INT	Jā	FK		FTP identifikators. Ārējā atslēga uz tabulu „Ftp_recipient”(2.6.5)

#### 2.6.4. Tabula „Email\_recipient”

Tabulas „Email\_recipients” uzdevums ir glabāt e-pasta adreses ar divu veidu ziņojumiem:

1. Izdošanās ziņojums(success\_message), kas tiek izmantots, ja lietotnes kompilācija noritējusi bez problēmām;
2. Neveiksmes ziņojums(failure\_email), kas tiek izmantots, ja lietotnes kompilācija neizdevās.

Lauka nosaukums	Datu tips	Obligāts	Atslēga	Noklusētā vērtība	Apraksts
idEmail_recipient	INT	Jā	PK		Automātiski ģenerēts e-pasta identifikators
email	VARCHAR(255)	Jā			E-pasta adrese
success_message	TEXT	Jā			Izdošanās paziņojums
failure_email	TEXT	Jā			Neizdošanās paziņojums.

### 2.6.5. Tabula „Ftp\_recipient”

Tabulas „Ftp\_recipients” uzdevums ir glabāt ftp datus par serveriem uz kuriem jāšūta pakotnes.

Lauka nosaukums	Datu tips	Obligāts	Atslēga	Noklusētā vērtība	Apraksts
idFtp_recipient	INT	Jā	PK		Automātiski ģenerēts FTP identifikators
username	VARCHAR(255)	Jā			FTP servera lietotāja lietotājvārds.
password	VARCHAR(255)	Jā			FTP servera lietotāja parole.
directory	TEXT	Jā			Ceļš FTP serverī, kur nogādāt pakotni.

### 2.6.6. Tabula „History”

Lauka nosaukums	Datu tips	Obligāts	Atslēga	Noklusētā vērtība	Apraksts
idHistory	INT	Jā	PK		Automātiski ģenerēts vēstures identifikators
message	TEXT	Jā			Ziņojums par izpildītajām operācijām
time_started	DATETIME	Jā		0	Kompilācijas sākuma laiks
success	BINARY	Jā			Kompilācijas izdošanās.
time_ended	DATETIME	Jā		0	Kompilācijas

					beigu laiks
--	--	--	--	--	-------------

### 3. TESTĒŠANAS DOKUMENTĀCIJA

#### 3.1. Testēšanas metodika

Testēšanai tiek izmantota „melnās kastes” testēšana, t.i. tiek testētas funkcijas ar ieejas datiem un pārbaudīts izejas rezultāts. Tiek salīdzināta programmas uzvedība pret specifikāciju.

Tiks pārbaudītas šādas darbības:

- Vai neautorizējies lietotājs spēj piekļūt sistēmas sadaļām,
- Ievaddatu ievade,
- Informācijas apskate, labošana un izvade

#### 3.2. Testēšanas žurnāls

##### 3.2.1. Lietotāja pieteikšanās un atteikšanās

Nr.	Testa apraksts	Sagaidāms rezultāts	Izdevās
1.	Tiek ievadīti korekti lietotāja dati.	Lietotājs tiek pieslēgts sistēmai, pievienots sesijai.	Jā
2.	Tiek ievadīti nekorekti lietotāja dati.	Lietotājam tiek paziņots, ka pieteikties neizdevās un lietotājs netiek pielaists sistēmā.	Jā
3.	Atteikšanās no sistēmas.	Lietotāja sesija tiek izdzēsta un lietotājs ir izlogojies.	Jā
4.	Apmeklētājs mēģina piekļūt sistēmas lietotāju funkcijām neesot ielogojies.	Apmeklētājs tiek novirzīts uz pieteikšanās skatu.	Jā

##### 3.2.2. Sistēmas konfigurācijas labošana

Nr.	Testa apraksts	Sagaidāms rezultāts	Izdevās
1.	Tiek ievadīti korekti konfigurācijas dati.	Jaunie konfigurācijas dati tiek iestatīti.	Jā
2.	Tiek ievadīti nekorekti konfigurācijas dati.	Lietotājam tiek paziņots par kļūdām. Tiek atgriezta kļūdainā forma ar norādēm par kļūdām.	Jā

3.	Tiek iestatīta tukša e-pasta ziņa.	Lietotājam tiek atgriezts paziņojums par kļūdu.	Jā
----	------------------------------------	---	----

### 3.2.3. CVS zaru veidošana

Nr.	Testa apraksts	Sagaidāms rezultāts	Izdevās
1.	Tiek ievadīti korekti zara izveides dati.	Lietotājs tiek novirzīts uz komandu apskates skatu.	Jā
2.	Tiek ievadīti nekorekti zara izveides dati.	Lietotājam tiek parādīti kļūdu ziņojumi.	Jā
3.	Tiek ievadīti korekti zara izveides dati ar iespējotu lietotnes kompilācijas iespēju nenorādot piegādes FTP serverus.	Lietotājam tiek parādīti kļūdas paziņojums par tukšu FTP servera adreses lauku.	Jā
4.	Tiek pieprasīts iegūt datumu no lietotnes kompilācijas numura nenorādot „Branch From” lauku.	Lietotājam tiek parādīts kļūdas paziņojums par tukšu „Branch from” lauku.	Jā
5.	Tiek pievienota nekorekta ftp adrese.	Lietotāja norādītā ftp adrese netiek pievienota.	Jā
6.	Tiek pievienota nekorekts e-pasts	Lietotāja norādītais e-pasts netiek pievienots.	Jā

### 3.2.4. Lietotnes pakotnes veidošana

Nr.	Testa apraksts	Sagaidāms rezultāts	Izdevās
1.	Tiek ievadīti korekti pakotnes kompilācijas dati.	Lietotājs tiek novirzīts uz komandu apskates skatu.	Jā
2.	Tiek ievadīti nekorekti pakotnes kompilācijas dati.	Lietotājam tiek parādīti kļūdu ziņojumi.	Jā
3.	Tiek pievienota nekorekta ftp adrese.	Lietotāja norādītā ftp adrese netiek pievienota.	Jā
4.	Tiek pievienota nekorekts e-	Lietotāja norādītais e-pasts netiek	Jā

	past	pievienots.	
--	------	-------------	--

### **3.3. Testēšanas rezultāti**

Ja iegūtais testēšanas rezultāts atšķīrās no sagaidāmā rezultāta vai tika izvadīta izpildes kļūda, tad sistēmas kods tika mainīts un kļūda tika labota. Testējot tika konstatēts, ka visi rezultāti sakrīt ar sagaidāmajiem rezultātiem.

## 4. PROJEKTA ORGANIZĀCIJA

Projekta izstrādē tiek izmantots ūdenskrituma modelis:

1. Tika apkopotas un analizētas prasības, un veidots programmatūras prasību specifikācijas dokuments.
2. Balstoties uz prasību analīzes rezultātiem, konsultācijām ar vadītāju un iepriekšējo pieredzi, tika izvēlēti izstrādes rīki – programmēšanas valoda Python, Flask ietvars, datu bāzes pārvaldnieks SQLite, lietotāju saskarnes bibliotēka Twitter Bootstrap.
3. Balstoties uz programmatūras prasību specifikāciju tika veidots programmatūras projektējuma apraksts un datu bāzes projektējums.
4. Sistēmas izstrāde un vienībtestēšana.
5. Kopējā sistēmas testēšana izmantojot „Melnās kastes” principu.

Izstrādes laikā stingri tika ievēroti MVC(Modelis-Skats-Kontrolleris) projektēšanas šablona principi un Python programmēšanas valodas stila ieteikumi(PIP8).

Darba gaitā tika izmantota Git versijkontroles rīks, lai veidotu izveidotās sistēmas rezerves kopijas un varētu tās ērti pārskatīt, vai atjaunot vecāku versiju problēmu gadījumā. Git repozitorijs tika glabāts Bitbucket vietnē kā privāts projekts.

## 5. KVALITĀTES NODROŠINĀŠANA

Lai nodrošināt izstrādātās sistēmas kvalitāti, attiecīgi tika veiktas šādas darbības:

- Sistēmas izstrāde notika pēc ūdenskrituma modeļa, un pirms sistēma tiek izstrādāta, tika sastādīta prasību specifikācija un izstrādāts projektējums.
- Sistēma balstās uz MVC (Modelis-Skats-Kontrolleris) projektēšanas šablonu, kā arī tika ievēroti Python[1] un Flask programmēšanas vadlīnijas [2].
- Strādājot tika izmantota Git versiju kontroles sistēma. Git repozitorijs tika glabāts Bitbucket servisā kā privāts projekts.
- Katras izstrādes fāzes nobeigumā tika veikta pārbaude, vai sistēma atbilst definētajām prasībām. Izstrādes cikla pēdējā fāzē tika pārbaudīta sistēmas kopējā atbilstība nedefinētajām prasībām.
- Dokumentācija tika veidota pēc atbilstošiem nozares standartiem.
- Tika veikta moduļu vienībtestēšana, kā arī testēšanas process tika dokumentēts[3].

## **6. KONFIGURĀCIJAS PĀRVALDĪBA**

Projekta laikā tika izmantota Git versiju kontroles sistēma. Koda izmaiņu saglabāšana tika veikta katru izstrādes dienu ar komentāriem par veiktajām izmaiņām. Šī sistēma tika izmantota, jo tā palīdz sekot līdzi sistēmas izmaiņu vēsturei, kas atvieglo atklūdošanas un citas ar konfigurācijas pārvaldi saistītos uzdevumus.

Pati Git repozitorija tika glabāta vietnē Bitbucket kā privāts projekts, kas nodrošināja to, ka neparedzētu apstākļu gadījumā nepazūd uz lokālās faila sistēmas izstrādātais kods.

## 7. DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības novērtēšanai izmantoju COCOMO modeļa darbietilpības kalkulatoru.

- Darba rezultātā izveidoto koda rindiņu skaits ir aptuveni 2500 LOC,
- Atkal izmantotais kods ir aptuveni 5000 rindiņas.

### 7.1. COCOMO modeļa darbietilpības aprēķināšana

Darbietilpības novērtēšanai ar COMODO modeli izmantoju tiešsaistes darbietilpības kalkulatoru[8]. Kalkulatorā tika uzstādīti šādi parametri:

- Jauns kods – 2500,
- Atkal izmantotais kods – 5000,
- Komandas saliedētība – ļoti augsta, jo pie projekta strādājis tikai viens izstrādātājs,
- Nepieciešamais programmatūras drošums – zems, jo sistēma paredzēta lietošanai kompānijas iekšējā tīklā ar ierobežotu piekļuvi,
- Datubāzes izmērs – zems

Atlikušie parametri tiek atstāti ar noklusētajām vērtībām. Kalkulators pie šiem izejas datiem aprēķina darbietilpību kā 5.6 personmēnešu garu. Šeit jāpiebilst, ka COCOMO modelis vairāk ir domāts vidēju (sākot no 32000 koda rindiņām) un lielu (sākot no 128000 koda rindiņām) sistēmu darbietilpības prognozēšanai, taču izmantojot mazākām sistēmām var veidoties neprecizitātes un aprēķinātais personmēnešu lielums var būt nedaudz lielāks nekā patiesībā. Reāli projekta realizācija notika 3 personmēnešu ilgā laikā.

## SECINĀJUMI

Kvalifikācijas darba ietvaros izstrādātā „Tīmeklī bāzēta programmatūras pārvaldības sistēma” nodrošina iespēju trenētai personai izveidot jaunus lietotnes zarus CVS repozitorijā un kompilēt šo lietotni, pēc kompilācijas nosūtīt pakotni uz ārējiem serveriem izmantojot FTP protokolu. Autors pārliecinājās, ka ir svarīgi sekot līdzi programmēšanas stilu vadlīnijām, lai būtiski samazinātu patērētos resursus un uzlabotu koda lasāmību. Ar pašlaik ierobežoto praktisko pieredzi autors nonācis pie slēdziena, ka darbietilpības novērtējumu labāk veikt konsultējoties ar pieredzējušākiem nozares speciālistiem un novērtējot savas prasmes.

## IZMANTOTĀ LITERTŪRA UN AVOTI

1. *Python programmēšanas stila vadlīnijas*. [tiešsaiste] Pieejams internetā: <http://www.python.org/dev/peps/pep-0008/>.
2. *Flask ietvara programmēšanas stila vadlīnijas*. [tiešsaiste] Pieejams internetā: <http://www.python.org/dev/peps/pep-0008/>
3. *Testēšanas metodes*. [tiešsaiste] Pieejams internetā: [http://home.lu.lv/~garnican/Testesan/test\\_n40.htm](http://home.lu.lv/~garnican/Testesan/test_n40.htm).
4. *Latvijas Valsts Standarts LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”*.
5. *Latvijas Valsts Standarts LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”*.
6. *Python dokumentācija*. [tiešsaiste] Pieejams internetā: <http://docs.python.org/2.7/>
7. *Flask dokumentācija*. [tiešsaiste] Pieejams internetā: <http://flask.pocoo.org/docs/>
8. *COMODO modeļa darbietilpības kalkulators*. [tiešsaiste] Pieejams internetā: <http://csse.usc.edu/tools/COCOMOII.php>

# PIELIKUMI

## Pielikums 1.

Fi Manager    Build << branch name >> started    Logged in as **user**    Logout

MANAGE

- Brancher

CONFIGURATION

- Settings

Help

Branch From

HEAD

\*Module

isn

\*New Branch Name

Build Number From

Date  Time

\*Branch type

Create Build

### FTP servers

ftp://user:12345@10.122.132.41/FooBar

ftp://kristaps.Qwerty123@10.122.132.41/FooBar

### Email recipients

Admin emails | [Edit](#)

- foo@bar.com
- foo.bar@foobar.com

Additional emails

*Att. 1. Zaru izveides skats*

Fi Manager    Build << branch name >> started    Logged in as **user**    Logout

MANAGE

- Brancher

CONFIGURATION

- Settings

Help

Branch From

HEAD

\*Module

isn

Date  Time

\*Branch type

### FTP servers

ftp://user:12345@10.122.132.41/FooBar

ftp://kristaps.Qwerty123@10.122.132.41/FooBar

### Email recipients

Admin emails | [Edit](#)

- foo@bar.com
- foo.bar@foobar.com

Additional emails

*Att. 2. Pakotnes izveides skats*

Fi Manager      Build << branch name >> started      Logged in as **user**      Logout

MANAGE

**Brancher**

CONFIGURATION

Settings

Help

## Commands to be executed:

**build-version-fi-5.0** `build-5.0dev-909 # Mon Jun 03 07:45:07 EET 2013 # isn --branch fi-r5_0 #s  
FreeBSD 4.7-RELEASE Using builtin specs.gcc version 2.95.4 20020320  
[FreeBSD] % fi-5.0`

CWD: /home/ктаube/tmp/builder/isn/build-version

**Creating Directories** `/home/ктаube/tmp/builder/product_build/fi_r5.0dev,  
/home/ктаube/tmp/builder/ftp/fi_r5.0dev`

**conf-fi-5.0** `<BUILDDIR>  
/home/ктаube/tmp/builder/product_build  
</BUILDDIR>`

CWD: /home/ктаube/tmp/builder/bin

**Build Command** `/home/ктаube/tmp/builder/bin/builder_script -F /home/ктаube/tmp/builder/bin/conf-fi-5.0`

CWD: /home/ктаube/tmp/builder

*Att. 3. Komandu priekšskatīšana*

Fi Manager      Build << branch name >> started      Logged in as **user**      Logout

MANAGE

**Brancher**

CONFIGURATION

Settings

Help

build-version-fi-5.0	✓
Creating Directories	✓
conf-fi-5.0	✓
Build Command	⚠
FTP upload	🔦
FTP upload	🔦

*Att. 4. Komandu izpilde ar kļūdu*

## Pielikums 2.

### Programmatūras pirmkoda paraugs.

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
autobuild.manager.commands
```

```
~~~~~
```

Objects representing commands.

All objects share same interface Command (ABS)

:copyright: (c) 2013 by Kristaps Taube.

```
"""
```

```
from subprocess import Popen, PIPE
```

```
import shutil
```

```
import sys
```

```
import os
```

```
import re
```

```
from auto_exceptions import CvsError, ConfigurationError, AppBuildddError
```

```
class Command(object):
```

```
    """ Object representing abstract base class
```

```
        All object implementing this interface should implement  
        their own run(), show(), and log() methods.
```

```
    """
```

```
    def __init__(self, cwd):
```

```
        # Working dirrectory where command will be executed
```

```
        self.cwd = cwd
```

```
        # True if execution succeeded; False if failed; None if not executed.
```

```
        self.success = None
```

```

self.err = None

def show(self):
    raise NotImplementedError

def run(self):
    raise NotImplementedError

def log(self):
    raise NotImplementedError

class ConfFile(Command):

    def __init__(self, template):
        """ Class for constructing configuration files.
            Takes path to tempalte file as argument in construction.
        """
        super(ConfFile, self).__init__(cwd=None)
        # template file -- is instance of ConfigurationTemplate
        self.template = template

    def show(self):
        """
        Return content of configuration file rendered from Jinja template.
        """
        return CommandsRepr(label=self.template.filename,
                             content=self.template.content,
                             cwd=self.template.cwd,
                             success=self.success,
                             err=self.err,
                             input=True)

```

```

def run(self):
    """
    Create configuration file from template.

    If there was no errors, self.success is set to True;
    Else self.set is set to False.
    """
    try:
        # Create file from template.
        self.template.make_file()
    except IOError, e:
        self.success = False
        raise ConfigurationError(message=e.message)
    else:
        self.success = True

```

```

class CvsCmd(Command):

```

```

    def __init__(self, command, cwd):
        super(CvsCmd, self).__init__(cwd=cwd)

        # CVS command
        self.command = command

        # Subprocess data
        self.return_code = None
        self.stdout = None
        self.stderr = None

```

```

    def show(self):

```

```

""" Return tuple with execution path and command to be executed """
c = ""
for i in ['tag', 'co', 'log', 'commit', 'add']:
    if i in self.command:
        c = i
        break
return CommandsRepr(label='CVS %s' % c,
                    cmd=' '.join(self.command),
                    cwd=self.cwd,
                    success=self.success,
                    err=self.err)

def run(self):
    """
    Run command in subprocess.
    In case return code from subprocess is not 0, CvsError exception is
    raised.
    """
    entry_dir = os.getcwd()
    try:
        os.chdir(self.cwd)

        # TODO: make file dynamic;
        process = Popen(self.command, stdout=PIPE, stderr=PIPE)
        self.stdout, self.stderr = process.communicate()
        # Wait for subprocess to execute and get return value.
        self.return_code = process.wait()

    if self.err and int(self.err) != 0:
        self.success = False
        raise CvsError(message="Exited with: %d, cwd: %s" %
                        (self.err, os.getcwd()),

```

```

        stdout=self.stdout,
        stderr=self.stderr,
        command=self.command)

except OSError():
    self.success = False
    raise CvsError(message="No such directory: %s" % self.cwd,
                    command=self.command)

except ValueError():
    self.success = False
    raise CvsError(message="Command passed to Popen is invalid: %s"
                    % self.command)

else:
    self.success = True
finally:
    os.chdir(entry_dir)

class DevEnv(Command):

    def __init__(self, env_directories):
        super(DevEnv, self).__init__(cwd=None)
        self.env_directories = env_directories

    def show(self):
        """ Returns path for each directory to be created. """
        return CommandsRepr(label="Creating Directories",
                             cmd=', '.join(self.env_directories),
                             success=self.success,
                             err=self.err)

```

```
def run(self):
    """ Create directories for each path passed in function argument. """
    try:
        for directory in self.env_directories:
            os.mkdir(directory)
            os.chmod(directory, int('777', 8))
    except OSError, e:
        self.success = False
        e.message = "Failed to create environment directories."
    else:
        self.success = True
```

Kvalifikācijas darbs „Tīmeklī bāzēta programmatūras pārvaldības sistēma” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Kristaps Taube* \_\_\_\_\_ .06.2013.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *LVU FMF lietišķā matem., Valdis Žuravļevs* \_\_\_\_\_ .06.2013.

Recenzents: *M. dat., Inguna Pede*

Darbs iesniegts 03.06.2013.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Imants Gorbāns* \_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_\_.06.2013. prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_