

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

INTERNETA VEIKALA DZINIS

KVALIFIKĀCIJAS DARBS

Autors: **Vladislavs Pone**

Studenta apliecības Nr.: vp11068

Darba vadītājs: asoc.prof. Jānis Zuters

RĪGA 2017

ANOTĀCIJA

Kvalifikācijas darbā “Interneta veikala dzinis” aprakstīta interneta veikalu dzinēja izstrāde.

Sistēmas mērķis ir izveidot atvērta pirmkoda dzinēju, kas dot iespēju paātrināt interneta veikalu izstrādi nodrošinot visbiezāk pieprasītas sistēmas prasības. Sistēma ir izstrādāta Ruby programmēšanas valodā, ar Ruby on Rails ietvara palīdzību.

Sistēmas izstrāde notika pēc spējās (Agile) programmatūras izstrādes pieejas.

Atslēgvārdi: Ruby, Ruby on Rails, tīmekļa sistēma, atvērta pirmkoda programmatūra.

ABSTRACT

The qualification work “Online shop engine” describes e-commerce sites engine development. System goal is to develop open-source engine that would make development of online shops faster by giving the most often requested system requirements. It is developed on ruby programming language, using Ruby on Rails framework.

The system was developed using Agile software development.

Keywords: Ruby, Ruby on Rails, web application, open-source application.

SATURS

APZĪMĒJUMU SARAKSTS.....	5
IEVADS.....	6
<u>PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....</u>	<u>7</u>
Nodaļas nolūks.....	7
Ievads.....	7
Vispārējs apraksts.....	7
<u>PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....</u>	<u>10</u>
Nodaļas nolūks.....	10
Saistība ar citiem dokumentiem.....	10
Moduļu dekompozīcijas apraksts.....	10
Ievads.....	10
Moduļu dekompozīcija.....	10
Moduļu klases.....	10
Kontrolieru klases.....	12
Skati.....	13
Datu dekompozīcija.....	15
Datu bāzes fiziskais modelis.....	21
Testēšana.....	22
Darbietilpības novērtējums.....	28
<u>IZMANTOTĀ LITERATŪRA UN AVOTI.....</u>	<u>29</u>

APZĪMĒJUMU SARAKSTS

- Ruby - dinamiska, objektorientēta, augsta līmeņa atvērta pirmkoda programmēšanas valoda.
- Ruby on Rails - uz ruby programmēšanas valodas rakstīts ietvars kas ir orientēts uz interneta lietotņu izstrādi.
- Klients - lietotājs kurš interneta veikala dzinēja sistēmu pēc tās pielāgošanas konkrētā interneta veikala sistēmas prasībām lieto lai pasūtītu un iegādātos vēlamos produktus.
- Sistēmas pārvaldītājs - lietotājs kurš Interneta veikala dzinēja sistēmu pēc tās pielāgošanas konkrētā interneta veikala sistēmas prasībām lieto lai apstrādātu veiktos pasūtījumus. Šim lietotājam ir jābūt iespēja gan rediģēt pēc iespējas vairāk sistēmas iestādījumu, gan arī iespējai pielāgot interneta veikala dzini konkrēta interneta veikala prasībām.

IEVADS

Pēdējo gadu laikā Latvijā kļuvis populāri veikaliem pārdot preces ne tikai klātienē, bet arī dot iespēju klientiem pasūtīt vēlamo preci izmantojot interneta tehnoloģijas. Tirgū ir izveidojušies tā saucamie interneta veikali, jeb veikali, kuriem nav fiziska veikala, vai arī nelielas platības veikals, kurā nav lielu preces izvēles iespēju, bet kuri ļauj apskatīt savas preces interneta vietnē un pasūtīt tās, vai arī rezervēt un fiziski saņemt tās noliktavā vai uzņēmuma birojā. Šīm precēm, atšķirībā no lielajiem veikalu piedāvātām precēm, visbiežāk netiek pieskaitīta izstāžu zāles nomas un konsultantu algas cena, kas dod iespēju pārdot klientam preci lētāk. Toties, viena no šādu biznesa modeļu grūtībām ir veikala vietnes izstrādāšana un uzturēšana, kas var negatīvi ietekmēt preču cenu, vai arī visu uzņēmuma biznesu.

Lai uzsāktu savu biznesu uzņēmējam pirmkārt ir jāizvēlas vienu no divām vietnes izstrādes iespējām - vai nu noligt programmētāju komandu, vai arī jāizvēlēties kādu no jau esošiem risinājumiem. Pirmais variants acīmredzot būs dārgāks un izstrādēs laiks varētu aizņemt vairākus mēnešus. Otrais variants izrādījās dārgāks uzturēšanā, jo 2015 gada septembrī, kad šis projekts tika izveidots, pasaulē pastāvēja daži gatavi risinājumi, piemēram prestashop, bet viņi visi bija projektēti darbam ar izstrādātāju pārdotiem moduļiem. Lai pārtaisītu tādu sistēmu par kaut ko, kas strādās ir nepieciešams gan programmētājs, gan vairāki mēneši, kas rezultātā ir visai neizdevīgi.

Rezultātā izveidojās ideja apvienot abas izvēles vienā - izstrādāt dzinēju, kurš ļaus programmētajam mazāk nekā mēneša laikā dabūt strādājošo vietni ar visu bieži šādos projektos pieprasīto biznesa loģiku. No paša sākuma projekts tika veidots kā atvērta pirmkoda projekts, kurš tiek izplatīts izmantojot MIT licenci.

Interneta veikala dzinis izstrādāts uz Ruby programmēšanas valodas ar ietvara Ruby on Rails palīdzību, ievērojot ietvara sabiedrības pieņemtos izstrādāšanas principus - DRY (Don't Repeat Yourself) kas paredz ka programmas kods tiek pēc iespējas mazāk dublēts un TDD (Test Driven Development) kas paredz, ka izstrādes laikā vispirms tiek izstrādāts tests, kas parbauda koda funkcionalitāti, un tikai tad tiek veidota pati funkcionalitāte, kurai turpmāk ir jāatbilst visiem testiem.

PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1. Nodaļas nolūks

Nodaļā tiek aprakstītas prasības izstrādājamai sistēmai. Prasības sistēmai būs pamats pārējiem izstrādes procesa posmiem – projektēšanai, plānošanai, programmēšanai, testēšanai.

2. Ievads

2.1. Saistība ar citiem dokumentiem

Programmatūras prasību specifikācija ir veidota, balstoties uz LVS 68:1996 („Programmatūras prasību specifikācijas ceļvedis”) standartu.

2.2. Darbības sfēra

Interneta veikala dzinis ir sistēma, kuras mērķis ir sniegt programmētājiem iespēju paātrināt interneta veikalu sistēmu izstrādes procesu, nodrošinot pamat funkcionalitāti un visbiežāk pieprasītas sistēmu prasības. Dziņa mērķos netiek iekļauti interneta veikalu vizuālā attēlojuma risinājumi, kuri būtu raksturīgi katram atsevišķam uz dziņa balstītam interneta veikalam.

3. Vispārējs apraksts

3.1. Produkta perspektīva

Sistēma ir balstīta uz Ruby on Rails ietvara un tā standartiem. Papildus rīki, kas tiek izmantoti sistēmā tiek definēti failā *Gemfile* saskaņā ar Ruby on Rails standartu.

3.2. Lietotāja raksturierzīmes

Sistēmai ir divi lietotāju tipi:

- Sistēmas pārvaldītāis (bieži programmētais), jeb cilvēks (vai cilvēki), kurš ir spējīgs pielāgot dzini sava projekta sistēmas prasībām - pievienot vajadzīgo funkcionalitāti un vizuālo noformēšanu, ka arī nokonfigurēt serveri.
Šim lietotāju tipam ir iespēja kontrolēt pasūtījumus un sistēmas iestādījumus
- Klients, jeb cilvēks (vai cilvēki), kas lieto jau pielāgoto sistēmu. Interneta veikala dzinējam ir jānodrošina pamat funkcionalitāte tieši šai lietotāju grupai lai programmētājam nebūtu jātērē savu laiku tās īstenošanai.

3.3. Sistēmas prasības

3.3.1. Izstrādes vide

Interneta veikala dzinēja izstrāde tiek veikta uz Ubuntu Server 16.04 LTS operētājsistēmas darbinātas mašīnas, uz kuras obligāti ir uzstādīts:

- **Ruby** programmēšanas valoda vismaz 2.0 versija, izmantojot vienu no divām versiju pārvaldības sistēmām - rbenv vai Ruby Version Manager (RVM)

- **Bundler** bibliotēka Ruby valodai
- **Ruby on Rails** vismaz 5.0 versija
- **Git** versiju pārvaldības sistēma
- Jebkura ar Ruby on Rails savienojama datu bāze (piem. **SQLite 3**)

3.4. Lietotājstāsti

Interneta veikalu dziņa mērķis ir uztaisīt pamata funkcionalitāti (back end) topošiem interneta veikaliem un pēc iespējas izvairīties no vizuālas reprezentācijas (front end) noteikšanas. Tāpēc visi lietotājstāsti tiek filtrēti, lai izslēgtu tos, kuri nosaka kā kaut kam ir jāizskatās un iekļautu tos, kuri nosaka kam un kā kaut-kam ir jābūt.

3.4.1. Sistēmas klientiem ir jābūt iespējai apskatīt produktus. Produktu raksturo:

- unikāls nosaukums.
- ir noteikta cena un daudzums.
- var būt noteikta atlaide, apraksts un kāds attēls vai vairāki attēli.

3.4.2. Produkti ir jābūt iedalītiem kategorijās. Kategorijas raksturo:

- ir noteikts nosaukums.
- var būt noteikts apraksts, ka arī viena vai vairākas apakškategorijas.

3.4.3. Klientiem sistēmā ir jābūt iespējai sistēmā pierēģistrēties. Pierēģistrēto klientu raksturo:

- unikāls lietotāja epasts un parole, kas ļaus šo klientu identificēt.
- klientam ir jābūt iespējai pieslēgties sistēmai (izveidot lietotāja sesiju).
- klientam ir jābūt iespējai atslēgties no sistēmas (iznīcināt lietotāja sesiju).
- klientam ir iespējai pieslēgties sistēmai tikai pareizi norādot savu lietotāja paroli.

3.4.4. Sistēmai ir jādod klientiem iespēja veikt pasūtījumus. Veiktie pasūtījumi tiek glabāti nenoteiktu laiku un pasūtījumus raksturo:

- Pasūtījumus var veikt tikai pierēģistrēts klients.
- tiem ir noteikts status un cena, kas tiek noteikta pasūtījuma veikšanas brīdī.

Pasūtījuma cenā ir jāiekļauj visu produktu cenas ar attiecīgiem daudzumiem, kā arī ir jāiekļauj visas atlaides, ja kādiem produktiem tādas pastāv, un visus kuponus, ja attiecīgajam klientam kādiem produktiem tie pastāv un klients izvēlās tos izmantot.

- Tikko izveidotiem pasūtījumiem ir jābūt statusam “New”. Sistēmas pārvaldītāis var uzstādīt statusu “Processing”, kamēr pasūtījums tiek apstrādāts vai gaida kamēr klients apmaksās rēķinu, uzstādīt statusu “Paid”, kamēr pasūtījums tika apstrādāts, apmaksāts un gaida piegādi, uzstādīt statusu “Delivery”, kamēr

pasūtījums tiek piegādāts klientam, uzstādīt statusu “Success”, kad pasūtījums tika veiksmīgi apstrādāts. Sistēmas pārvaldītājs var uzstādīt statusu “Fail”, gadījumos kad pasūtījums tika neveiksmīgi apstrādāts.

3.4.5. Sistēmai ir jādot iespēja klientiem glābāt komentārus par produktiem.

Komentārus raksturo tas, ka katram komentāram ir autors, komentāra teksts un tam ir jākomentē kāds produkts.

3.4.6. Sistēmas pārvaldītājam ir jābūt iespējai iedot pierēģistrētam klientam kuponu.

Kuponus raksturo:

- ir noteikts nosaukums, jeb kupona kods
- kuponam ir jādefinē atlaide, vai nu kādas pasūtījuma cenas daļas apmērā, vai nu kāda procenta no pasūtījuma cenas apmērā.
- kuponi, pat tie kuri tika izlietoti, glabājās sistēma nenoteiktu laiku. Izlietotie kuponi klientam netiek radīti.

3.4.7. Sistēmas pārvaldītājam ir jābūt iespējai rediģēt gan interneta veikala nosaukumu, gan tā pārvaldošā uzņēmuma nosaukumu, bankas informāciju, adresi un reģistrācijas un nodokļu maksātāja numurus. Tāpat sistēmas pārvaldītājam ir jābūt iespēja rediģēt produktu informācija.

3.4.8. Sistēmā ir jābūt iespējai pierēģistrētiem klientiem glābāt savu piegādes informāciju.

PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

1. Nodaļas nolūks

Šajā nodaļā tiek aprakstīts zema līmeņa datu un algoritmu projektējumi izstrādājamai sistēmai.

Tā kā izstrāde notika iteratīvi, pēc spējās programmatūras izstrādes pieejas, pirms dažām iterācijām sistēma tika pārprojektēta, lai ieviestu jauno funkcionalitāti. Šajā nodaļā aprakstīts tikai pēdējais projektējums, kas ir aktuāls uz nodaļas uzrakstīšanas brīdi.

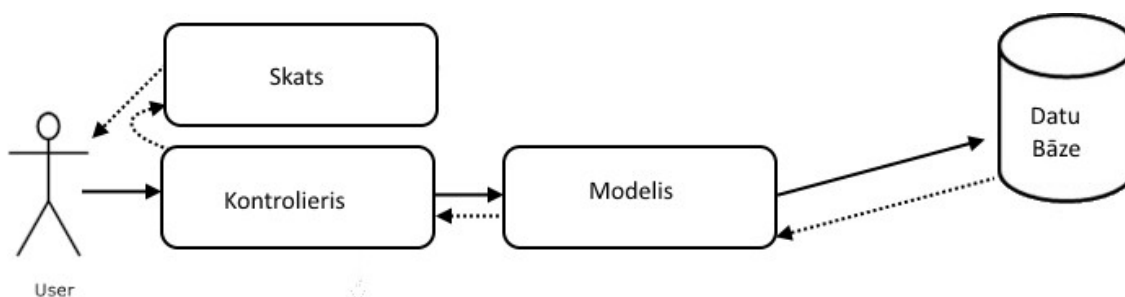
2. Saistība ar citiem dokumentiem

Programmatūras projektējuma apraksts ir veidots, balstoties uz LVS 72:1996 („Ieteicamā prakse programmatūras projektējuma aprakstīšanai”) standartu.

3. Moduļu dekompozīcijas apraksts

3.1. Ievads

Uz Ruby on Rails ietvara vietnes strādā pēc MVC shēmas, kas nozīmē, ka visus pieprasījumus var aprakstīt ar sekojošo shēmu:

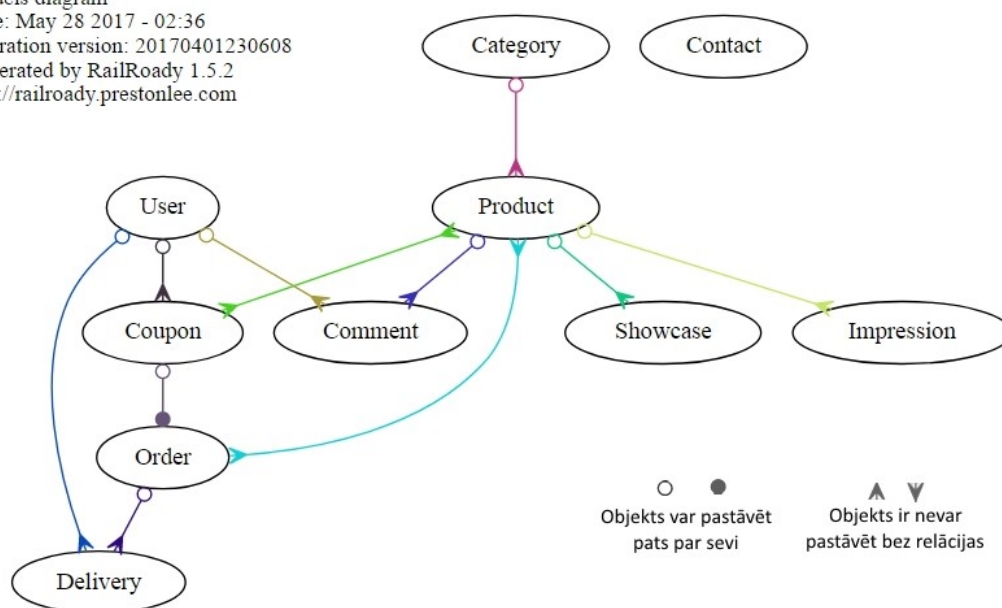


Programmas struktūra ir iedalāma trīs grupās:

- Modeļi – satur sistēmas biznesa loģiku, nodrošina datu pārbaudi un apstrādi.
- Kontrolieri – nodrošina komunikāciju ar lietotāju, atgriežot nepieciešamo skatu ar lietotājam piemēroto informāciju.
- Skati – atbild par datu reprezentāciju lietotājam.

3.2. Moduļu dekompozīcija

3.2.1. Moduļu klases



3.2.1.1. Klase *Category*

- Nolūks
Nodrošināt sistēmā iespēju kārtot produktus kategorijās.
- Funkcija
Nodrošina katras kategorijas informācijas validitāte un meklēšanas sistēmu optimizācijas (Search Engine Optimization) funkciju.
- Atkarības
Product

3.2.1.2. Klase *Comment*

- Nolūks
Nodrošināt sistēmā iespēju komentēt produktus.
- Funkcija
Nodrošina katra komentāra informācijas validitāti.
- Atkarības
Product, User

3.2.1.3. Klase *Contact*

- Nolūks
Nodrošināt sistēmā iespēju glābāt sistēmas informāciju.
- Funkcija
Nodrošina sistēmas informācijas validitāti.

3.2.1.4. Klase *Coupon*

- Nolūks
Nodrošināt sistēmā iespēju izmantot kuponus.
- Funkcija
Nodrošina katra kupona informācijas validitāti.
- Atkarības
Product, User, Order

3.2.1.5. Klase *Delivery*

- Nolūks
Nodrošināt sistēmā iespēju glābāt piegādes informāciju.
- Funkcija
Nodrošina katra piegādes informācijas validitāti.

3.2.1.6. Klase *Order*

- Nolūks
Nodrošināt sistēmā iespēju pasūtīt produktus.
- Funkcija
Nodrošina katra pasūtījuma informācijas validitāte, tai skaitā pasūtījuma statusu.
- Atkarības
Product, User, Coupon

3.2.1.7. Klase *Product*

- Nolūks
Nodrošināt sistēmā iespēju glābāt produktu informāciju.
- Funkcija
Nodrošina katra produkta informācijas validitāti, meklēšanas dzinēju optimizācijas (Search Engine Optimization) funkciju, statistikas informāciju, ka arī metodes darbam ar produktu un produkta attēliem.
- Atkarības
Category, User, Order, Coupon, Comment, Showcase

3.2.1.8. Klase *Showcase*

- Nolūks
Nodrošināt attēlu pārvaldību produktiem.
- Funkcija
Nodrošina produktam attiecīgo attēlu izmantošanu.
- Atkarības

Product

3.2.1.9. Klase *User*

- Nolūks

Nodrošināt lietotāju informācijas pārvaldību.

- Funkcija

Nodrošina katra lietotāja informācijas validitāti, tai skaitā to lietotāju, kuri ir sistēmas pārvadītāji.

3.2.2. Kontrolieru klases

3.2.2.1. Klase *AdminsController*

- Nolūks

Nodrošināt iespēju sistēmas pārvaldītajam mainīt sistēmas datus.

- Funkcija

Nodrošina sistēmas pārvaldītajam iespēju apskatīt un rediģēt lietotāju, pasūtījumu, produktu, kategoriju, kuponu, piegādes un sistēmas iestādījumu informācijas.

3.2.2.2. Klase *ApplicationController*

- Nolūks

Pēc Ruby on Rails ietvara standarta no ApplicationController klases metodes manto visas pārejas kontrolieru klases.

- Funkcija

Nodrošina pārejas klases ar biznesa loģiku, kas tiks pielietota ja nē visos kontrolieros, tad lielākajā to daļā. Nodrošina lietotāju autentifikāciju, darbības ar pasūtījuma sagatavošanu, kā arī statistikas informācijas ierakstīšanu.

3.2.2.3. Klase *CategoriesController*

- Nolūks

Nodrošināt pieeju kategorijas datiem.

- Funkcija

Nodrošina kategoriju attēlošanu.

3.2.2.4. Klase *CouponsController*

- Nolūks

Nodrošināt pieeju kuponu datiem.

- Funkcija

Nodrošina lietotājiem iespēju apskatīt sev piederošos kuponus. Nodrošina

sistēmas pārvaldītājam iespēju izveidot, rediģēt un dzēst lietotāju kuponus.

3.2.2.5. Klase *OrdersController*

- Nolūks
Nodrošināt iespēju lietotājam veikt pasūtījumus.
- Funkcija
Nodrošina lietotāju ar pasūtījuma izveides skatu, ka arī veic ievadītas informācijas apstrādi.

3.2.2.6. Klase *ProductsController*

- Nolūks
Nodrošināt iespēju lietotājam informāciju par produktu.
- Funkcija
Nodrošina lietotājam iespēju apskatīt produktus un atzīmēt tos nākamajām pasūtījumam.

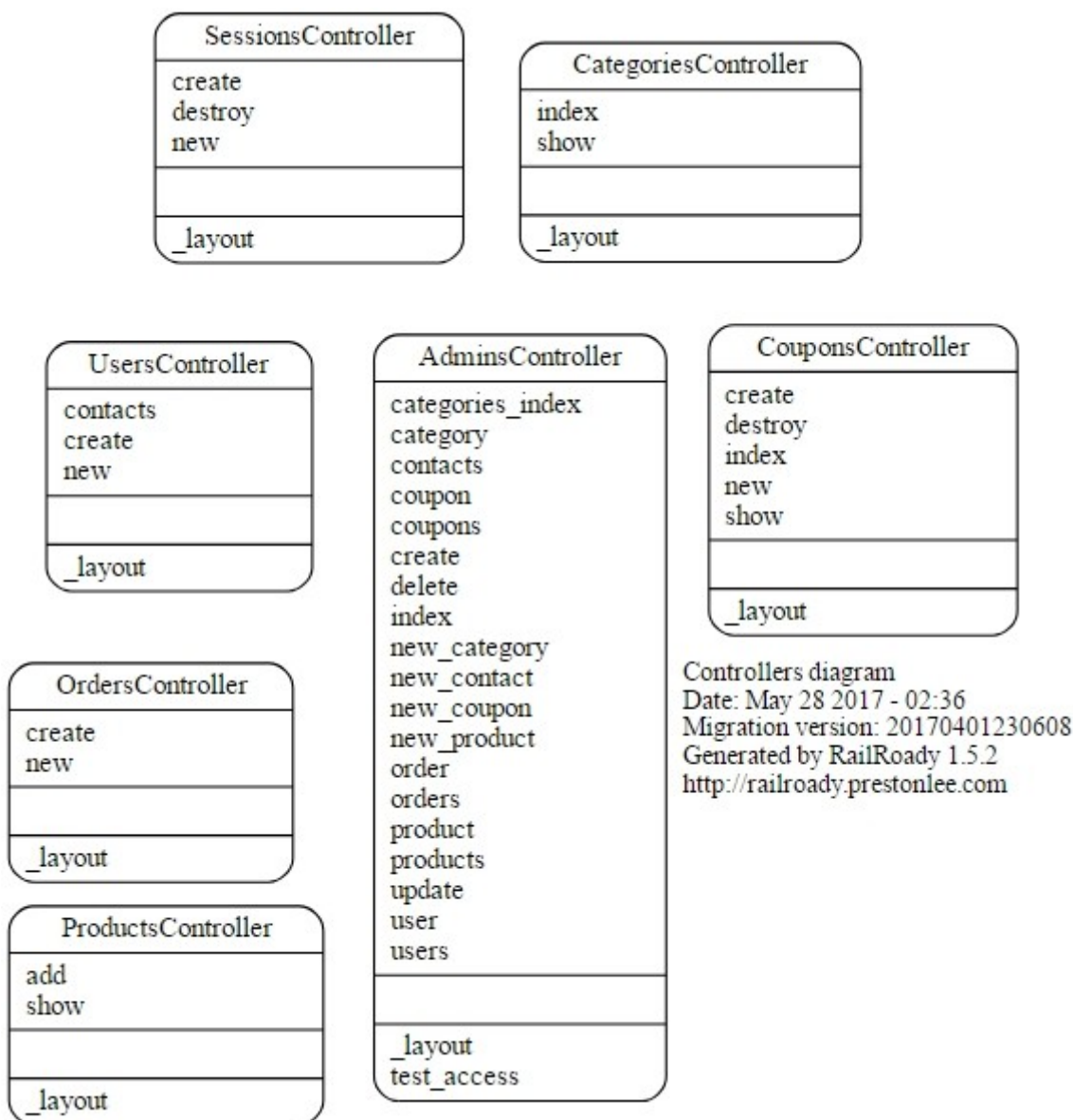
3.2.2.7. Klase *SessionsController*

- Nolūks
Nodrošināt lietotāju autorizāciju.
- Funkcija
Nodrošina lietotāju ar autorizācijas skatu un apstrādā lietotāja ievadīto informāciju.

3.2.2.8. Klase *UsersController*

- Nolūks
Nodrošināt iespēju lietotājiem pierēģistrēties sistēmā.
- Funkcija
Nodrošina lietotāju reģistrāciju.

3.2.3. Skati



3.2.3.1. Klases *AdminsController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu lietotāju, pasūtījumu, produktu, kategoriju, kuponu, piegādes un sistēmas iestādījumu rediģēšanas formām un sarakstiem.

3.2.3.2. Klases *CategoriesController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu kategoriju un produktu sarakstu

attēlošanai.

3.2.3.3. Klases *CouponsController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu lietotājiem piederošo kuponu un to rediģēšanas sadaļai.

3.2.3.4. Klases *OrdersController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu pasūtījuma izveides skatu.

3.2.3.5. Klases *ProductsController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu produkta sadaļai.

3.2.3.6. Klases *SessionsController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu autorizācijas sadaļai.

3.2.3.7. Klases *UsersController* skati

- Nolūks
Nodrošināt vizuālās saskarnes ģenerāciju.
- Funkcija
Ģenerē iezīmēšanas valodas (HTML) kodu reģistrācijas un sistēmas informācijas sadaļām.

3.3. Datu dekompozīcija

3.3.1. Ievads

Interneta veikala dzinēja izstrādāšanas laikā tika izmantota SQLite3 datu bāze, bet Ruby on rails ietvars dod iespēju viegli nomainīt to pret kādu citu, atbilstoši katra uz interneta veikalu dzinēja projekta sistēmas prasībām (iestādījumi atrodas failā /config/databases.yml). Visas interneta veikala dzinēja sistēmai vajadzīgas

tabulas tiek saglabātas gan kā migrācijas (tabulu uzstādīšanas instrukcija), gan kā datu bāzes shēma.

3.3.2. Tabula *categories*

Šajā tabula tiek glabāta informāciju par veikalā esošajām produktu kategorijām un to koku.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
name	varchar(255)	Nē			Kategorijas nosaukums.
ancestry	varchar(255)	Jā			Kategorijas atrašanas vieta kategoriju kokā (tiek izmantota ancestry bibliotēkā).
description	text	Jā			Kategorijas apraksts.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.
slug_en	varchar(255)	Nē			Papildus atslēga meklēšanas dzinēju optimizācijas (Search Engine Optimization) sistēmām (domāta angļu valodai).

3.3.3. Tabula *comments*

Šajā tabula tiek glabāta informāciju par produktiem piesaistītiem klientu komentāriem.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
user_id	int(11)	Nē		FK	Lietotāja (komentāra autora) identifikators.
product_id	int(11)	Nē		FK	Produkta (komentāra mērķa) identifikators.
text	text	Nē			Komentāra teksts.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.

3.3.4. Tabula *contacts*

Šajā tabula tiek glabāta veikala informācija. Katras ieraksts reprezentē to savā valodā.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
language	varchar(255)	Nē			Sniegtas informācijas valoda.
name	varchar(255)	Nē			Veikala nosaukums.
adress	varchar(255)	Jā			Veikala fiziska adrese.
phone	varchar(255)	Jā			Veikala fiziskais telefona numurs.
email	varchar(255)	Jā			Veikala epasta adrese.
company_name	varchar(255)	Nē			Veikala pārvaldoša uzņēmuma nosaukums.
company_reg_nr	varchar(255)	Nē			Veikala pārvaldoša uzņēmuma reģistrācijas numurs.
company_vat_nr	varchar(255)	Nē			Veikala pārvaldoša uzņēmuma.
bank	text	Jā			Veikala pārvaldoša uzņēmuma bankas konts.
time_table	text	Jā			Veikala darba laiks.
description	text	Jā			Veikala apraksts.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.

3.3.5. Tabula *coupons*

Šajā tabula tiek glabāta informāciju par klientiem esošiem kuponiem, tai skaitā arī jau izlietotiem.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
code	varchar(255)	Nē			Kupona kods.
percent	varchar(255)	Jā			Atlaides lielums, kad atlaide ir izteikta procentos.
amount	text	Jā			Atlaides lielums, kad atlaide ir izteikta noteiktā vienībā.
user_id	int(11)	Jā		FK	Lietotāja (kupona īpašnieka) identifikators.
used_flag	smallint(1)	Jā			Norāde, vai kupons jau ir izlietots.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.

3.3.6. Tabula *coupons_products*

Šajā tabula tiek glabāta informāciju par kuponu un produktu savstarpējo savietojumu.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
coupon_id	int(11)	Jā		FK	Kupona (kurš ir izmantojams ar attiecīgo produktu) identifikators.
product_id	int(11)	Jā		FK	Produkta (kuram kupona atlaide tiek piemērota) identifikators.

3.3.7. Tabula *deliveries*

Šajā tabula tiek glabāta piegādes informācija.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
deliverable_type	varchar(255)	Nē			Piesaistīta modeļa nosaukums.
deliverable_id	int(11)	Nē			Piesaistīta modeļa identifikators.
country	varchar(255)	Nē			Valsts nosaukums.
city	varchar(255)	Nē			Pilsētas nosaukums.
adress	varchar(255)	Nē			Adrese norādītājā pilsētā.
zip	varchar(255)	Nē			Adreses pasta indekss.
info	text	Jā			Papildus informācija.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.

3.3.8. Tabula *impressions*

Šajā tabula tiek glabāta statistikas informācija. To izmanto kopā ar *impressionist* bibliotēku

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
user_id	int(11)	Jā		FK	Lietotāja identifikators.
controller_name	varchar(255)	Jā			Iesaistīta kontroliera nosaukums.
action_name	varchar(255)	Jā			Iesaistītas kontroliera darbības nosaukums (piem. skata nosaukums).
view_name	varchar(255)	Jā			Iesaistīta skata nosaukums (Šajā bibliotēkas versijā netiek izmantots).
request_hash	varchar(255)	Jā			Pieprasījuma informācija.
ip_address	varchar(255)	Jā			Klienta IP adreses.
session_hash	varchar(255)	Jā			Klienta sesijas informācija.
message	text	Jā			Papildus informācija, kuru varetu pievienot informācijas ierakstīšanas brīdī.
referrer	text	Jā			Pieprasījuma novirzes avots, gadījumos kad tāds pastāv.
created_at	datetime	Jā			Izveidošanas laiks.
updated_at	datetime	Jā			Rediģēšanas laiks.
params	text	Jā			Pieprasījuma parametri. Kontrolieris un tā darbība tiek filtrēti lai izvairītos no dublikātiem tabulā.

3.3.9. Tabula *orders*

Šajā tabula tiek glabāta informācija par veiktajiem pasūtījumiem, tai skaitā arī par jau veiksmīgi un neveiksmīgi izpildītiem pasūtījumiem.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
user_id	int(11)	Nē		FK	Lietotāja (pasūtītāja) identifikators.
price	decimal(8,2)	Nē			Pasūtījuma preču cena tā noformēšanas brīdī, iekļaujot visas atlaides.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
status	varchar(255)	Nē			Pasūtījuma status.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.
coupon_id	int(11)	Jā		FK	Kupona (kas tika izmantots lai mainītu cenu) identifikators.

3.3.10. Tabula *orders_products*

Šajā tabula tiek glabāta informāciju par pasūtījumu un produktu savstarpējo savietojumu.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
order_id	int(11)	Nē		FK	Pasūtījuma (kurā produkts ir pasūtīts) identifikators.
product_id	int(11)	Nē		FK	Produkta (kas tika iekļauts pasūtījumā) identifikators.

3.3.11. Tabula *products*

Šajā tabula tiek glabāta informāciju par veikalā pārdotiem produktiem.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
name	varchar(255)	Nē			Produkta nosaukums.
price	decimal(8,2)	Nē			Produkta oriģinālā cena.
discount	decimal(8,2)	Jā			Produkta cena ar atlaidi.
sale_flag	smallint(1)	Nē	False		Norāde uz to, vai produktam darbojas atlaides cena.
description	text	Jā			Produkta apraksts.
category_id	int(11)	Nē		FK	Kategorijas (kurā produkts ir iedalīts) identifikators.
quantity	int	Nē	0		Palikuša produkta daudzums.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
counter_cache	int	Jā			Skatījumu skaits.
image	varchar(255)	Jā			Noklusēta attēla adrese.
slug_en	varchar(255)	Nē			Papildus atslēga meklēšanas dzinēju optimizācijas (Search Engine Optimization) sistēmām (domāta angļu valodai)

3.3.12. Tabula *showcases*

Šajā tabula tiek glabāta informāciju par produktu attēlu galeriju.

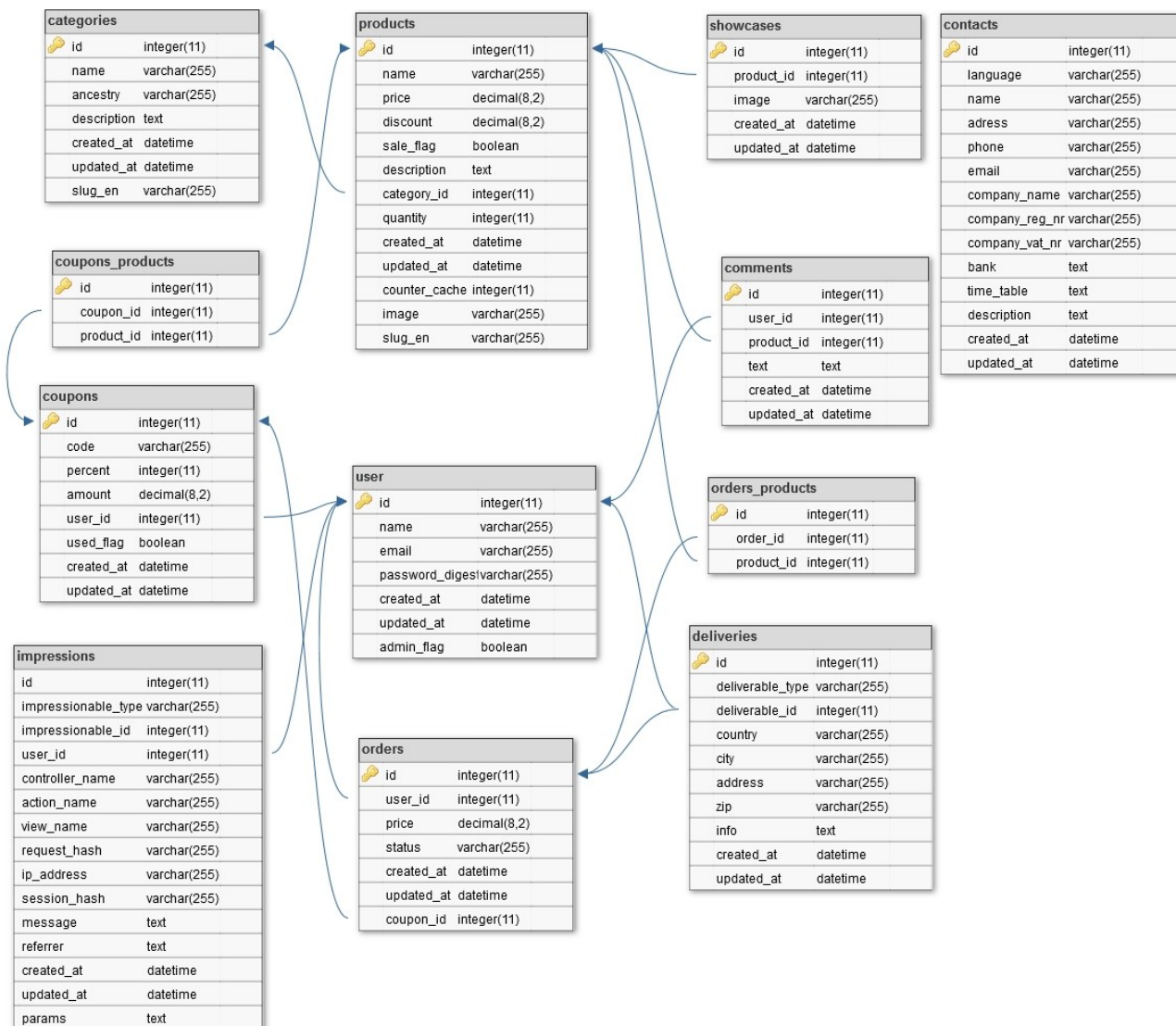
Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators.
product_id	varchar(255)	Jā		FK	Produkta (kuram attēli ir piesaistīti) identifikators.
image	varchar(255)	Jā			Attēla adrese.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.

3.3.13. Tabula *users*

Šajā tabula tiek glabāta informāciju par lietotājiem.

Lauks	Datu tips	Var būt nulle	Noklusēta vērtība	Atslēga	Komentāri
id	int(11)	Nē		PK	Unikāls identifikators
name	varchar(255)	Nē			Lietotāja vārds un/vai uzvārds.
email	varchar(255)	Jā			Lietotāja epasta adrese.
password_digest	text	Jā			Lietotāja parole, kriptētā veidā.
created_at	datetime	Nē			Izveidošanas laiks.
updated_at	datetime	Nē			Rediģēšanas laiks.
admin_flag	smallint(1)	Jā	False		Norāde uz to, vai lietotājam ir sistēmas pārvaldītāja iespējas.

3.4. Datu bāzes fiziskais modelis



4. Testēšana

4.1. Ievads

Interneta veikala dzinēja kods tika izstrādāts pēc testu virzītas izstrādes principa (Test Driven Development), kas nosaka ka pirms rakstīt sistēmas kodu programmētājam ir, balstoties uz lietotārstāstu, jāuzraksta automatizēts test (kurš no sākuma radīs ka kods šo testu neizpilda) un tikai pēc tam ir jāraksta kods, kurš veiksmīgi pildīs šo testu. Ruby on Rails ietvarā pēc noklusējuma tas tiek realizēts ar Minitest bibliotēku. Šajos testos tiek izmantota cita datu bāze ar tādu pašu struktūru, bet ar citiem datiem. Katram testam pēc iespējas tiek ģenerēta jauna informācija (piem. katrā testā, kur ir vajadzīgs produkts, tiek ģenerēts pavisam jauns produkts tieši šim testam) un šie testi katru reizi tiek palaisti nejaušā kārtībā.

```

.....
.....
Finished in 9.278187s, 10.1313 runs/s, 14.0114 assertions/s.
94 runs, 130 assertions, 0 failures, 0 errors, 0 skips
vlad@ubuntu:~/www/eMarket$ █

```

Šobrīd visi automatizētie testi izpildās sekmīgi. Tas tiek pārādīts izpildot konsolē standarta Ruby on Rails komandu *rake*, kura palaiž visus uzrakstītos automatizētos testus.

4.2. Kontrolieru testi

Kontrolieru testus sauc arī par funkcionāliem testiem. To nolūks ir pārbaudīt vai sistēma pareizi apstrādā ienākošos pieprasījumus. Tāpēc katrai kontroliera klasei ir izveidota viena testu kopa, kur katram kontroliera skatam tiek veidots viens test lai pārbaudītu vai kontrolieris veiksmīgi ģenerē skatu (pats skats pārbaudīts tiek integrācijas testos, skata esamību pārbauda ar atbildes statusa koda palīdzību, piem. 200 nozīme ka skats tika veiksmīgi ģenerēts). Gadījumos kad skata ģenerācijā ir atkarīga no lietotāja autentifikācijas tiek ģenerēti vairāki testi, kas pārbauda gan vai skats tiek veiksmīgi ģenerēts, gan vai skats ģenerēts netiek (piem. notiek pāradresācija, kods 401) attiecīgajos nosacījumos.

Kontrolierim ApplicationController testi veidoti parasti netiek, jo tā darbība tiek pārbaudīta katrā testā.

Npk.	Testējamā Kontrolieru klase	Automatizētā testa nosaukums	Testa nolūks
1	AdminsController	test should not get update if guest, test should not get update if user, test should get update if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt rediģēšanas rīcībai.
2	AdminsController	test should not get index if guest, test should not get index if user, test should get index if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt galvenai administrēšanas lapai.

Npk.	Testējamā Kontrolieru klase	Automatizētā testa nosaukums	Testa nolūks
3	AdminsController	test should not get users if guest, test should not get users if user, test should get users if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt lietotāju administrēšanas lapai.
4	AdminsController	test should not get user if guest, test should not get user if user, test should get user if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt lietotāja administrēšanas lapai.
5	AdminsController	test should not get categories if guest, test should not get categories if user, test should get categories if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt kategoriju administrēšanas lapai.
6	AdminsController	test should not get category if guest, test should not get category if user, test should get category if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt kategorijas administrēšanas lapai.
7	AdminsController	test should not get new category page if guest, test should not get new category page if user, test should get new category page if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt jaunās kategorijas administrēšanas lapai.

Npk.	Testējamā Kontrolieru klase	Automatizētā testa nosaukums	Testa nolūks
8	AdminsController	test should not get products if guest, test should not get products if user, test should get products if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt produktu administrēšanas lapai.
9	AdminsController	test should not get product if guest, test should not get product if user, test should get product if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt produkta administrēšanas lapai.
10	AdminsController	test should not get new product page if guest, test should not get new product page if user, test should get new product page if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt jaunā produkta administrēšanas lapai.
11	AdminsController	test should not get orders if guest, test should not get orders if user, test should get orders if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt pasūtījumu administrēšanas lapai.
12	AdminsController	test should not get order page if guest, test should not get order page if user, test should get order page if admin	Pārbauda lai tikai sistēmas pārvaldītājs varētu piekļūt pasūtījuma administrēšanas lapai.

Npk.	Testējamā Kontrolieru klase	Automatizētā testa nosaukums	Testa nolūks
13	AdminsController	test should not get new contacts page if guest, test should not get new contacts page if user, test should get new contacts page if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt jaunās vietnes iestādījumu valodas administrēšanas lapai.
14	AdminsController	test should not get contacts if guest, test should not get contacts if user, test should get contacts if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt vietnes iestādījumu administrēšanas lapai.
15	AdminsController	test should not delete anything if guest, test should not delete anything if user, test should delete anything if admin	Pārbauda lai tikai sistēmas pārvadītājs varētu piekļūt dzēšanas rīkam.
16	CategoriesController	test get category	Pārbauda lai kategorijas lapa būtu pieejama.
17	CategoriesController	test get category index	Pārbauda lai kategoriju lapa būtu pieejama.
18	CouponsController	test index is available to users	Pārbauda lai klienta kuponu saraksts būtu pieejams pierēģistrētam klientam.
19	CouponsController	test index is not available to guests	Pārbauda lai klienta kuponu saraksts nebūtu pieejams neautenticētam klientam.
20	CouponsController	test show is available to users	Pārbauda lai klienta kupons būtu pieejams pierēģistrētam klientam.
21	CouponsController	test show is not available to guests	Pārbauda lai klienta kupons nebūtu pieejams neautenticētam klientam.

Npk.	Testējamā Kontrolieru klase	Automatizētā testa nosaukums	Testa nolūks
22	OrdersController	test get order redirects if guest	Pārbauda lai pasūtījumu forma nebūtu pieejama neautenticētam klientam.
23	OrdersController	test get order if user	Pārbauda lai pasūtījumu forma būtu pieejama pierēģistrētam klientam.
24	ProductsController	test get product	Pārbauda lai produkta lapa būtu pieejama.
25	SessionsController	test create session	Pārbauda lai autentifikācijas lapa būtu pieejama.
26	SessionsController	test create user session	Pārbauda lai veiksmīgs autentifikācijas mēģinājums būtu sekmīgs.
27	SessionsController	test create user incorrect session fail	Pārbauda lai neveiksmīgs autentifikācijas mēģinājums nebūtu sekmīgs.
28	SessionsController	test destroy user session	Pārbauda lai autentifikācijas dzēšanas lapa būtu pieejama.
29	UsersController	test get new user	Pārbauda lai reģistrācijas forma būtu pieejama.
30	UsersController	test create user	Pārbauda lai veiksmīgs reģistrācijas mēģinājums būtu sekmīgs.
31	UsersController	test create user without password fail	Pārbauda lai neveiksmīgs reģistrācijas mēģinājums būtu nesekmīgs.

4.3. Modeļu testi

Modeļu testus sauc arī par vienības testiem. To nolūks ir pārbaudīt vai pareizi notiek attiecīga modeļa informācijas validācija. Tāpēc katram modeļa klasei tika izveidota viena testu kopa, kur katram parametram (tai skaitā arī attiecībām ar citiem moduļiem) noteikts test uz validitāti.

Npk.	Automatizētā testa nosaukums	Testa nolūks	Lietotārstāsta numurs
1	Category can not be saved without name	Pārbauda lai katrai kategorijai būtu vārds.	3.4.2
2	Category has no ancestry if it is root	Pārbauda lai katrai kategorijai varētu būt apakškategorijas.	3.4.2
3	Comment can not be saved without author	Pārbauda lai katrai kategorijai varētu būt apakškategorijas.	3.4.5
4	Comment can not be saved without product	Pārbauda lai katrs komentārs pieder kādam produktam.	3.4.5
5	Comment can not be saved without text	Pārbauda lai katram komentāram būtu teksts.	3.4.5
6	Contacts should have specific language	Pārbauda lai katra interneta veikala informācija būtu savā valodā.	3.4.7
7	Shop must have name	Pārbauda lai katra interneta veikala informācija saturētu interneta veikala nosaukumu.	3.4.7
8	Company must have name	Pārbauda lai katra interneta veikala informācija saturētu interneta veikala nosaukumu.	3.4.7
9	Company must have registration number	Pārbauda lai katra interneta veikala informācija saturētu interneta veikala pārvaldoša uzņēmuma reģistrācijas numuru.	3.4.7
10	Company must have VAT number	Pārbauda lai katra interneta veikala informācija saturētu interneta veikala nodokļu maksātāja numuru.	3.4.7
11	Coupon can not be saved without code	Pārbauda lai katram kuponam būtu nosaukums, jeb kupona kods	3.4.6
12	Coupon can not be saved without amount or percent	Pārbauda lai katram kuponam būtu definēta atlaide, vai nu kādas pasūtījuma cenas daļas apmērā, vai nu kāda procenta no pasūtījuma cenas apmērā.	3.4.6

Npk.	Automatizētā testa nosaukums	Testa nolūks	Lietotārstāsta numurs
13	Coupon is invisible to users if it is used	Pārbauda lai kuponi kuri tika izmantoti vairs neparādās klientam.	3.4.6
14	Delivery can not be saved without deliverable	Pārbauda lai katrai klienta piegādes informācijai būtu autors.	3.4.8
15	Delivery can not be saved without country	Pārbauda lai katrai klienta piegādes informācijai būtu piegādes valsts.	3.4.8
16	Delivery can not be saved without city	Pārbauda lai katrai klienta piegādes informācijai būtu piegādes pilsēta.	3.4.8
17	Delivery can not be saved without address	Pārbauda lai katrai klienta piegādes informācijai būtu piegādes adrese.	3.4.8
18	Delivery can not be saved without zip	Pārbauda lai katrai klienta piegādes informācijai būtu piegādes pasta kods.	3.4.8
19	save order without price fails	Pārbauda lai katram pasūtījumam būtu cena.	3.4.4
20	save order without status fails	Pārbauda lai katram pasūtījumam būtu status	3.4.4
21	Product can not be saved without name	Pārbauda lai katram produktam būtu nosaukums.	3.4.1
22	Product default quantity is 0	Pārbauda lai katra produktam daudzums ir skaitliska vērtība (pēc noklusējuma nulle).	3.4.1
23	Product must have price	Pārbauda lai katram produktam būtu cena.	3.4.1
24	Product must have category	Pārbauda lai katram produktam būtu nosaukums.	3.4.1
25	User email must be uniq	Pārbauda lai katram klientam būtu unikāla eposta adrese.	3.4.1

4.4. Integrācijas testi

Integrācijas testi ir domāti lai testētu darba gaitu, kas var mainīties dažādos interneta veikala dzinēja pielietojumos. Tāpēc šajā darbā viņi netiek iekļauti, bet tiek stingri ieteikts lai tie tiktu uzrakstīti piemērojot interneta veikala dzini konkrētas sistēmas prasībām.

5. Darbietilpības novērtējums

Darbietilpības novērtējumam tika izvēlēta COCOMO basic metode, kas nosaka darbietilpību

pēc uzrakstīta koda rindiņu skaita.

Lai dabūtu darbā uzrakstīta koda rindiņu skaitu es izmantoju rīku cloc

```

152 text files.
146 unique files.
96 files ignored.

http://cloc.sourceforge.net v 1.60 T=1.13 s (96.1 files/s, 3598.5 lines/s)
-----
Language          files          blank          comment          code
-----
Ruby              89            233            304             1614
HTML              4             20             0               239
SASS              9            307            1204             42
YAML              4             12             39              28
Javascript        1             1              17              1
CSS               1             1              16              0
CoffeeScript      1             0              3               0
-----
SUM:              109           574            1583            1924
-----

```

Pirmkoda rindiņu skaits - 1924	KLOC = 1,924	
Komanda sastāv no 1 cilvēka	a=2,4	b=1,05
Projekta tips - Organic	c=2,5	d=0,38
Personmēneši = $a * KLOC^b = 2,4 * 1,924^{1,05} = 4,8$ (mēneši)		
Izstrādes laiks = $c * Personmēneši^d = 4,54$ (mēneši)		

IZMANTOTĀ LITERATŪRA UN AVOTI

1. <https://github.com/VladRoker/eMarket/> - interneta veikala dzinēja Git repozitorija atrašanas vieta.
2. <http://guides.rubyonrails.org/> - satur ļoti daudz vadlīnijas par to, kā labāk izstrādāt vietnes ar Ruby on Rails ietvara palīdzību.
3. <http://api.rubyonrails.org/> - Ruby on Rails oficiāla ietvara dokumentācija.
4. <https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-with-rbenv-on-ubuntu-16-04> - satur instrukcijas kā pareizi uzstādīt Ubuntu 16.04 sistēmu darbam ar ietvaru un interneta veikala dzinēju.
5. <https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-with-rvm-on-ubuntu-16-04> - līdzīga instrukcija, ja rbenv vietā gribas lietot rvm (kas arī tika darīts izstrādes laikā).
6. <https://github.com/charlotte-ruby/impressionist> - statistikas vākšanas bibliotēkas kods un lietošanas instrukcijas.
7. <https://www.packtpub.com/books/content/working-rails-%E2%80%93-activerecord-migrations-models-scaffolding-and-database-completion> - Satur Ruby on Rails migrāciju un MySQL kolonu atbilstības tabulu, ka arī vispārēji noderīgu informāciju par darbu ar datu bāzēm Ruby on Rails ietvarā.
8. <https://www.prestashop.com/> - projekts ar līdzīgu tēmu, izstrādāts PHP programmēšanas valodā. Izmanto moduļu sistēmu
9. <https://opensource.org/licenses/MIT> - MIT licencēšanas teksts.
10. <http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/kutcher/kutcher.html> - tīmeklī pieejams COCOMO basic kalkulators.
11. <https://github.com/AlDanial/cloc> - cloc rīks pirmkoda rindiņu skaita uzskaitītšanai.
12. <http://railroady.prestonlee.com> - RailRoady rīks, kas ģenerē modeļu un kontrolieru diagrammas Ruby on Rails projektiem.

Kvalifikācijas darbs „*Interneta veikala dzinis*” 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: *Vladislavs Pone* _____ .05.2017.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *asoc.prof. Jānis Zuters* _____ .05.2017.

Recenzents: *dipl.inž. Miks Rozenbergs*

Darbs iesniegts 29.05.2017.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2017. prot. Nr. _____

Komisijas sekretārs(-e): _____