

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**SAP HYBRIS PRODUKTU UZGLABĀŠANAS UN
PIEDĀVĀJUMA ĶĒDES LIETOTNE**

KVALIFIKĀCIJAS DARBS

Autors: **Katerīna Verdiša**

Studenta apliecības Nr.: kv15015

Darba vadītājs: Mg. comp. sc. Jurijs Nikolajevs

RĪGA 2018

ANOTĀCIJA

Kvalifikācijas darba ietvaros tika apskatīta un aprakstīta SAP Hybris Commerce platformā izstrādātā lietotne, kas ir saistīta ar produkta uzglabāšanas un piedāvājuma ķēdes atspoguļojumu. Izmantojot doto lietotni, lietotājs var sekot produkta piegādes procesam sākot no noliktavas atrašanās vietas līdz veikalam kā arī līdz pašam pasūtītājam.

Piegādes punkti un ķēdes tiek atspoguļotas uz pasaules kartes vektorgrafikas formātā, izmantojot klienta, noliktavas un veikala atrašanas vietas koordinātas. Dotā lietotne ir ļoti ērta, jo klientam ir iespēja redzēt un izvēlēties sev vispiemērotāko piegādēs iespēju no visām piedāvātām iespējām.

Lietotne tika izstrādāta izmantojot SAP Hybris Commerce platformu, kas ir viena no pieprasītākajām e-komercijas realizācijas platformām. Tā tika realizēta kā daļa no internetveikala, kas dod iespēju integrēt to jebkuros e-komercijas projektos. Izstrādes laikā tika izmantotas HTML, LESS un JavaScript programmēšanas valodas.

Atslēgvārdi: SAP, Hybris, SAP Hybris Commerce, Javascript, HTML, LESS, piegāde.

ABSTRACT

SAP HYBRIS PRODUCT WAREHOUSING AND SUPPLY CHAIN APPLICATION

Within the framework of the qualification paper, the author reviewed and described the application developed by the SAP Hybris Commerce platform, related to the presentation of the product storage and supply chain. By using the given application, the user can track the delivery process of the product from the warehouse to the store as well as to the customer.

Supply points and chains are depicted on the world map in vector graphic format, by using the coordinates of the customer, warehouse and store's location. This application is very convenient because the customer has the opportunity to see and choose the most suitable delivery option from the all given choices.

The application was developed on the SAP Hybris Commerce platform, which is one of the most sought after e-commerce platforms. It was implemented as part of an online store, which gives an opportunity to integrate it into any e-commerce project. The HTML, LESS and JavaScript programming languages were used during the development.

Keywords: SAP, Hybris, SAP Hybris Commerce, Javascript, HTML, LESS, delivery.

SATURS

APZĪMĒJUMU SARAKSTS.....	6
IEVADS	7
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	8
1.1. Ievads	8
1.1.1. Nolūks	8
1.1.2. Darbības sfēra.....	8
1.1.3. Saistība ar citiem dokumentiem	8
1.1.4. Pārskats.....	8
1.2. Vispārējs apraksts	9
1.2.1. Produkta perspektīva.....	9
1.2.2. Produkta funkcijas.....	9
1.2.3. Lietotāja raksturozīmes	10
1.2.4. Esošais stāvoklis.....	10
1.2.5. Vispārējie ierobežojumi	11
1.2.6. Pieņēmumi un atkarības	11
1.3. Funkcionālās prasības	11
1.3.1. Lietotnes modulis	11
1.3.1.1. Kartes inicializēšana	11
1.3.1.2. Darbības izsaukšana, pēc kartes inicializēšanas	12
1.3.1.3. Kartē esošo punktu un loku dzēšana.....	12
1.3.1.4. Loka zīmēšana	13
1.3.1.5. Klienta atrašanas vietas detektēšana.....	13
1.3.1.6. Veikala atrāšanas vietas detektēšana	14
1.3.1.7. Noliktavas atrāšanas vietas detektēšana	15
1.3.1.8. Piegādes ķēdes attēlošana.....	15
1.4. Nefunkcionālās prasības	16
1.4.1. Veiktspējas prasības	16
1.4.2. Pieejamība	16
1.4.3. Drošība	16
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	17
2.1. Ievads	17
2.1.1. Nolūks	17

2.1.2. Darbības sfēra.....	17
2.1.3. Definīcijas un saīsinājumi	17
2.1.4. Saistība ar citiem dokumentiem	17
2.2. Dekompozīcijas apraksts	17
2.3. Atkarību apraksts	18
2.4. Datu bāzes projektējums	19
2.4.1. Detalizēts datu bāzes tabulu projektējums	19
2.4.1.1. Tabula User	19
2.4.1.2. Tabula Customer	19
2.4.1.3. Tabula Shop_category	20
2.4.1.4. Tabula Shop.....	21
2.4.1.5. Tabula Warehouse	22
2.4.1.6. Tabula Product.....	22
2.4.1.7. Tabula Cart	23
2.4.1.8. Tabula User_group	24
2.4.1.9. Tabula Order.....	24
2.5. Saskarnes apraksts	26
3. TESTĒŠANAS DOKUMENTĀCIJA	29
3.1. Testēšanas metodika	29
3.2. Testēšanas žurnāls.....	29
4. PROJEKTA ORGANIZĀCIJA	31
5. KVALITĀTES NODROŠINĀŠANA	32
6. KONFIGURĀCIJU PĀRVALDĪBA	33
7. DARBIETILPĪBAS NOVĒRTĒJUMS.....	34
SECINĀJUMI	35
IZMANTOTĀ LITERATŪRA	36
PIELIKUMI.....	37

APZĪMĒJUMU SARAKSTS

SAP Hybris Commerce – e-komercijas platforma, uz kuras bāzes tiek veidotas sistēmas ar B2B un B2C funkcionalitāti.

B2B (Business to Business) – ir komerciālās attiecības starp diviem uzņēmumiem, kas savstarpējos darījumos izmanto e-komercijas informācijas sistēmas.

B2C – Business to Customer sistēmas ir paredzētas gala patērētājiem, tās ir analogs tradicionālajam veikalam, katalogu pasūtījumiem un telemārketingam, tikai Internetā.

HTML – Hyper-Text Markup Language.

JavaScript - skriptu valoda, pārsvarā izmantota tīmekļa vietņu izstrādē.

PPS – Programmatūras prasību specifikācija.

PPA – Programmatūras projektējuma apraksts.

LESS – Dinamiskā stila lapas apstrādes valoda.

Backoffice – administratīvs rīks, kas tiek pielietots SAP Hybris commerce platformā, lai atvieglotu administratīvus uzdevumus.

UI – User Interface - atbild par produkta vizuālo noformējumu, hierarhiju, interakcijām un citiem vizuāliem elementiem ar kuriem saskarsies lietotājs.

IEVADS

Mūsdienās ir strauji pieaudzis tīmekļa veikalu un to lietotāju skaits. Iepirkšanās interneta vidē dod pircējam iespēju izvēlēties dažādas preces, neņemot vērā to izcelsmes valsti vai pieejamību pircēja dzīves vietā.

Viena no svarīgākajām īpašībām, kam lietotāji mēdz pievērst uzmanību, pasūtot preci, neskaitot preces cenu un kvalitāti, ir paredzētais preces piegādes laiks un līdz ar to arī izsūtīšanas vieta. Ļoti iespējams, cilvēki uzticētu veikalam lielāku daļu savu ikdienas pirkumu, ja būtu iespēja redzēt piedāvātās preces piegādes ceļu un izvēlēties sev vispiemērotāko un ērtāko variantu.

Šī kvalifikācijas darba mērķis ir izstrādāt lietotnes moduli, kas ļautu SAP Hybris Commerce platformā izveidotā interneta veikalā vizuāli attēlot izvēlēta produkta iespējamo atrašanās vietu. Projekta ietvaros paredzēts kartē parādīt noliktavas, kurās atrodas prece, veikala atrašanās vietu, ka arī klienta pozīciju uz kartes, izmantojot lietotāja ierīcē iebūvētās globālās pozicionēšanas iespējas, vizualizēt piegādes ķēdi no noliktavas līdz veikalam un tālāk līdz klientam. Šis modulis dos iespēju pircējam efektīvi izvēlēties ērtāko un ātrāko piegādi, balstoties uz lietotāja un preces atrašanās vietu.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Programmatūras prasību specifikācijas nolūks ir aprakstīt programmas sistēmas prasības, tās funkcijas un pielietojuma iespējas.

Programmatūras prasību specifikācija ir paredzēta un izstrādāta programmatūras pasūtītājiem, kā arī lietotājiem, lai palīdzētu iepazīties ar izveidojamās sistēmas darbošanās principiem.

1.1.2. Darbības sfēra

SAP Hybris Commerce platformā izstrādāta produktu uzglabāšanas un piedāvājuma ķēdes sistēma ir lietotne, kas paredzēta produkta glabāšanas vietas un piegādes iespējamā ceļa izsekošanai attiecībā pret klienta atrašanas vietu.

1.1.3. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” prasības.

1.1.4. Pārskats

Programmatūras prasību specifikācija (turpmāk PPS) sastāv no četrām daļām:

Pirmajā daļā tiek aprakstīta ievadinformācija, kas sniedz īsu PPS apskatu, aprakstīts nolūks, darbības sfēra un saistība ar citiem dokumentiem

Otrajā daļā ir aprakstīti vispārējie faktori, kas iedarbojas uz produktu un tā prasībām:

- lietotnes attīstības perspektīvas;
- visas iespējamās lietotnes funkcijas;
- produkta iespējamo lietotāju raksturiezīmes;
- PPS iespaidojošu faktoru apraksts.

Trešajā daļā tiek aprakstītas sistēmas funkcionālās prasības un funkciju detalizēts apraksts.

Ceturtajā daļā tiek aprakstītas nefunkcionālās prasības, sistēmas veiktspēja, pieejamība un drošība.

1.2. Vispārējs apraksts

1.2.1. Produkta perspektīva

Šī lietotne ir modulis, kas integrēts SAP Hybris Commerce B2C sistēmā. Tās uzdevums ir attēlot lietotāja, veikala un noliktavas atrašanas koordinātas uz kartes un, savienojot tos secīgi, parādīt visizdevīgāko produktu piegādes veidu lietotāja izvēlei. Šīs lietotnes uzdevums ir saistīts tikai ar datu lasīšanu un vizualizēšanu.

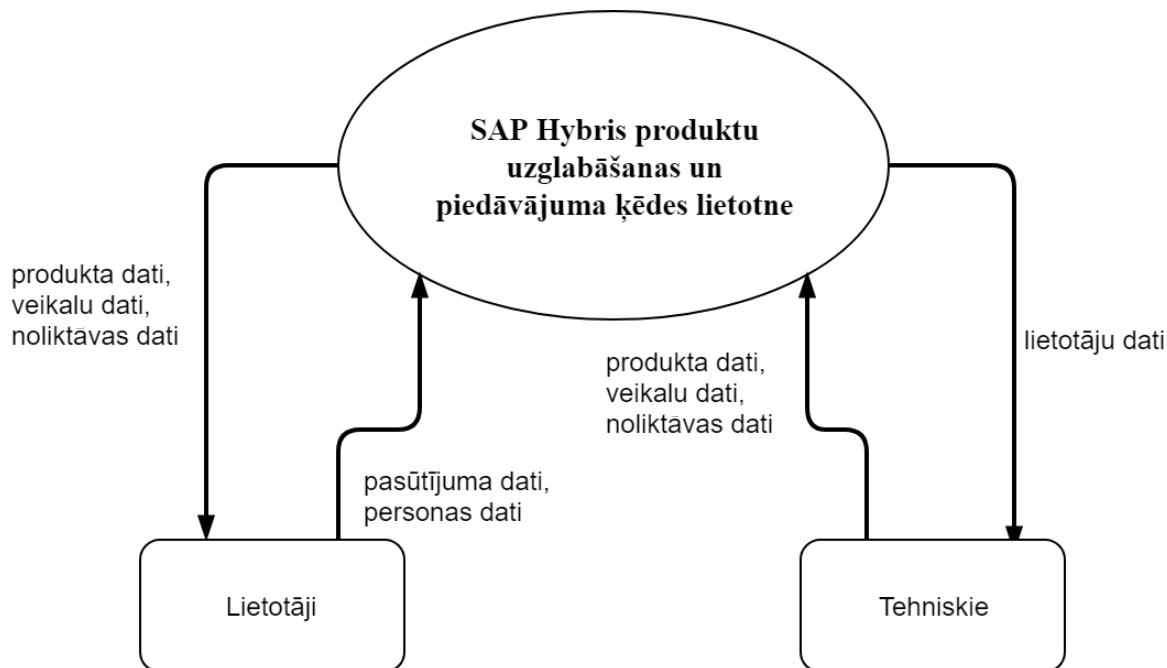
Ar laiku paredzēts paplašināt lietotnes funkcionalitāti, pievienojot tai funkciju izrēķināt iespējamo produkta piegādēs laiku, balstoties uz attālumu un vidējo piegādes ātrumu.

1.2.2. Produkta funkcijas

Produktam tiek realizētas šādas funkcijas:

1. Kartes inicializēšana
2. Darbības izsaukšana pēc kartes inicializēšanas
3. Kartē esošo punktu un loku dzēšana
4. Loka zīmēšana
5. Klienta atrašanās vietas detektēšana
6. Veikala atrašanās vietas detektēšana
7. Noliktavas atrašanās vietas detektēšana
8. Piegādes ķēdes attēlošana

1.2.3. Lietotāja raksturiežīmes



1.1. att. Uzglabāšanas un piedāvājuma ķēdes lietotnes 0. līmeņa datu plūsmu diagramma

Sistēmas lietotājus var iedalīt divas grupās. Katrai no tām ir savas raksturiežīmes (skatīt 1.1. attēlu). Piemēram, abu grupu lietotājiem jābūt datora lietošanas pamatprasmēm, kā arī prasmei strādāt ar pārlūkprogrammām. Nav noteiktas īpašas prasības saistībā ar izmantojamajām pārlūkprogrammām, bet ieteicams tās atjaunināt uz pēdējo (jaunāko) pieejamo versiju, lai sasniegtu labāko rezultātu.

Grupā "Tehniskie" ietilpst sistēmas izstrādātāji un administratori, kam jābūt izglītotiem visas sistēmas izmantošanā. Viņu uzdevums ir pārvaldīt sistēmu, ievadīt tajā un rediģēt nepieciešamos datus. Sistēmas izstrādātājiem jābūt ar pieredzi programmēšanā, lai tie nākotnē spētu lietotni uzturēt.

Grupai "Lietotāji" nav nepieciešamas papildu zināšanas bez iepriekš noteiktajām vispārīgajām zināšanām.

1.2.4. Esošais stāvoklis

Lietotnei pastāv analogi, tomēr tā atšķirsies ar savu unikālo vizualizāciju.

1.2.5. Vispārējie ierobežojumi

Lai lietotājs turpmāk vārētu pilnīgi izmantot lietotni, ir nepieciešams interneta pieslēgums un atļauja pārlūkprogrammā izmantot ģeolokāciju. Savukārt tehniskajiem lietotājiem datu ievadei ir nepieciešama uz servera instalēta un pareizi konfigurēta SAP Hybris Commerce platforma.

1.2.6. Pieņēmumi un atkarības

Atkarība tiek izpausta ar nepieciešamību tehniskajai lietotāju grupai izmantot SAP Hybris Commerce platformu datu ievadei.

1.3. Funkcionālās prasības

1.3.1. Lietotnes modulis

1.3.1.1. Kartes inicializēšana

Identifikators: initD3map
Mērķis
Funkcija dod iespēju izveidot un attēlot uz ekrāna pasaules karti.
Ievaddati
Lietotne ielādē kartes veidošanai nepieciešamos datus: kartes garums, platums, kartes forma. Atzīmē valsts teritorijas un to robežas.
Apstrāde
Sistēma ielādē karti lietotnē.
Izvaddati
Kartes attēls parādās uz ekrāna.
Paziņojumi
Nav

1.3.1.2. Darbības izsaukšana, pēc kartes inicializēšanas

Identifikators: MapInitToPerformAction
Mērķis
Izsaukt darbību ar karti, kad karte ir pilnībā inicializēta (ir ielādēti visi kartei nepieciešami resursi). Pārbauda, vai karte ir veiksmīgi inicializēta ik pēc 300 milisekundēm un, ja inicializācija bija veiksmīga, izsauc padoto darbību.
Ievaddati
Funkcija darbībai ar karti, izveidotie dati no initD3map funkcijas.
Apstrāde
Funkcija pārbauda, vai karte tika veiksmīgi inicializēta, un veiksmīgas inicializācijas gadījumā izsauc argumentā padoto darbību.
Izvaddati
Tiek palaista darbība ar karti.
Paziņojumi
Nav

1.3.1.3. Kartē esošo punktu un loku dzēšana

Identifikators: clearMapFromSupplyChains
Mērķis
Funkcija dod iespēju izdzēst kartē esošos punktus un lokus, lai atzīmētu jaunus.
Ievaddati
Lietotājs nospiež uz citu veikalu.
Apstrāde
Sistēma izdzēs iepriekš kartē attēlotos lokus un punktus, lai attēlotu jaunus.
Izvaddati
No kartes tiek izdzēsti kartē esošie punkti un loki.

Paziņojumi
Nav

1.3.1.4. Loka zīmēšana

Identifikators: drawArc
Mērķis
Funkcija dod iespēju kartē attēlot loku starp divām koordinātām.
Ievaddati
Loka sākuma un loka beigu koordinātas.
Apstrāde
Tiek apstrādāti padoto koordinātu dati, un loks tiek attēlots uz kartes.
Izvaddati
Tiek attēlots loks.
Paziņojumi
Nav

1.3.1.5. Klienta atrašanas vietas detektēšana

Identifikators: dataPosition
Mērķis
Funkcija dod iespēju kartē parādīt lietotāju atrašanās vietu.
Ievaddati
No piedāvātā saraksta jāizvēlas veikals un jānospiež uz tā.
Apstrāde
Tiek apstrādātas klienta atrašanās vietas koordinātas. Ja klienta atrašanās vietas dati ir pieejami, klienta atrašanās punkts tiek pievienots attēlošanas masīvam. Ja klienta atrašanās vietas dati ir atslēgti, masīvam klienta atrašanās vietu nav iespējams attēlot.

Izvaddati
Masīvs ar klienta atrašanās koordinātām.
Paziņojumi
Gadījumā, kad klientam ir atslēgta ģeopozicionēšana vai sistēma nevar identificēt viņa atrašanās vietu, pārlūkprogrammas konsolē tiks izvadīts paziņojums: <i>“User location detection error has occurred!”</i>

1.3.1.6. Veikala atrāšanas vietas detektēšana

Identifikators: dataStore
Mērķis
Funkcija dod iespēju parādīt veikala atrašanās vietu uz kartes.
Ievaddati
No piedāvātā saraksta jāizvēlas veikals un jānospiež uz tā.
Apstrāde
Tiek pārbaudīts, vai izvēlētajam veikalam ir aizpildīti koordinātu lauki. Ja veikala atrašanās vietas dati ir pieejami, tā atrašanās punkts tiek pievienots koordinātu masīvam. Ja veikala atrašanās vietas dati nav ievadīti, tad veikala atrašanās vietu nav iespējams attēlot kartē.
Izvaddati
Masīvs ar pievienotām veikala koordinātām.
Paziņojumi
Nav

1.3.1.7. Noliktavas atrašanas vietas detektēšana

Identifikators: dataWarehouse
Mērķis
Funkcija dod iespēju parādīt noliktavas atrašanās vietu kartē.
Ievaddati
No piedāvātā saraksta jāizvēlas veikals un jānospiež uz tā.
Apstrāde
Tiek pārbaudīts, vai ar veikalu sasaistītajām noliktavām ir aizpildīti koordinātu lauki. Ja noliktavas atrašanās vietas dati ir pieejami, tās atrašanas punkts tiek pievienots koordinātu masīvam. Ja noliktavas atrašanās vietas dati ir atslēgti, noliktavas atrašanas vietu nav iespējams attēlot kartē.
Izvaddati
Masīvs ar pievienotām noliktavas koordinātām.
Paziņojumi
Nav

1.3.1.8. Piegādes ķēdes attēlošana

Identifikators: chainLoop
Mērķis
Funkcijas dod iespēju parādīt ķēdi, kas savieno visas apstrādātās atrašanās vietas.
Ievaddati
No piedāvātā saraksta jāizvēlas veikals un jānospiež uz tā. Funkcijai tiks padots koordinātu masīvs.
Apstrāde
Atkarībā no pieejamajiem datiem tiek attēlotā preces piegādes ķēde (koordinātas koordinātu masīvā tiek attēlotas kartē un savienotas ar lokiem).

Izvaddati
Kartē parādās vizualizēta piegādes ķēde.
Paziņojumi
Nav

1.4. Nefunkcionālās prasības

1.4.1. Veiktspējas prasības

Platformai pieslēgtajai datubāzei jābūt spējīgai nodrošināt noliktavu un veikalu koordinātu glabāšanu, lai tās būtu iespējams attēlot kartē.

Sistēmai jānodrošina iespēja 700 aktīviem lietotājiem vienlaicīgi izmantot sistēmu.

1.4.2. Pieejamība

Sistēmas pieejamībai jābūt ne zemākai par 95%.

Ja sistēmā tiks konstatēta pēkšņa kļūme, kuras dēļ sistēmas darbība tiks pārtraukta, tad tās darbības atjaunošana ir jānodrošina vienas nedēļas laikā.

1.4.3. Drošība

Sistēma nodrošinās drošu datu pārraidi starp serveri un klientu izmantojot HTTPS protokolu.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Programmatūras projektējuma apraksts (turpmāk, PPA) ir visu PPS minēto prasību attēlojums. Tas tiek radīts analīzes, plānošanas, implementēšanas un lēmumu pieņemšanas atvieglojumam. PPA kalpo kā sistēmas uzmetums jeb modelis.

2.1.2. Darbības sfēra

Lietotne paredzēta produkta glabāšanas vietas un piegādes iespējamā ceļa izsekošanai attiecībā pret klienta atrašanās vietu.

2.1.3. Definīcijas un saīsinājumi

Definīcijas, akronīmi un saīsinājumi tika aprakstīti nodaļā “Apzīmējumu saraksts”.

2.1.4. Saistība ar citiem dokumentiem

Programmas projektējuma apraksts ir izstrādāts saskaņā ar standartu LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

Dokuments ir saistīts ar šīs sistēmas programmatūras prasību specifikāciju, kura ir aprakstīta un trešajā nodaļā ar nosaukumu “Programmatūras prasību specifikācija”.

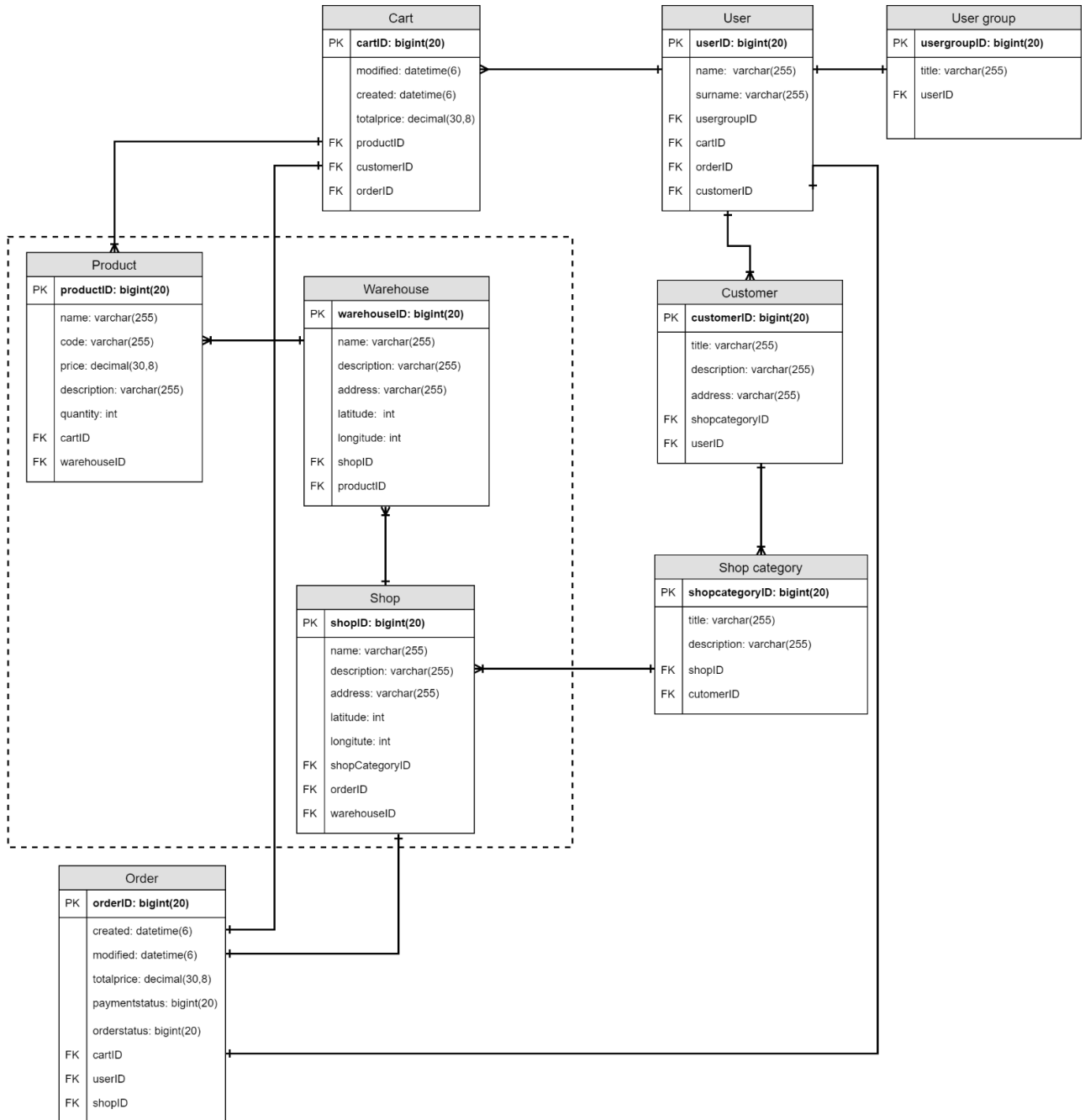
2.2. Dekompozīcijas apraksts

Sistēmas dati tiek glabāti MySQL datu bāzē. Datu bāze sastāv no deviņām tabulām:

1. User – tabulā tiek glabāti lietotāju dati
2. Customer – tabulā tiek glabāti dati par esošajām klientu grupām
3. Shop_category – tabula ar veikala kategoriju aprakstošiem datiem
4. Shop – tabula ar datiem par katru konkrēto veikalu
5. Warehouse – tabula ar datiem par konkrēto noliktavu
6. Product – tabula, kas satur produkta datus
7. Cart – tabula, kurā glabājas lietotāju veikalā izveidotā groza dati
8. User_group – tabula ar lietotāju grupu datiem
9. Order – tabula ar datiem par klienta izdarīto pasūtījumu un tā statusu

2.3. Atkarību apraksts

Datu atkarību var aplūkot datu bāzes ER modelī (skatīt 2.1. attēlu). Ar pārtraukto līniju tiek apzīmēta datubāzes daļa, kas ir pielietota un attēlota tieši lietotnē.



2.1. att. Datubāzes fiziskais ER modelis

2.4. Datu bāzes projektējums

Šajā apakšnodaļā tiek aprakstīts sistēmas datu bāzes projektējums.

Datubāzes tabulās varchar laukiem tiek lietots utf-8_bin kodējums.

2.4.1. Detalizēts datu bāzes tabulu projektējums

2.4.1.1. Tabula User

Kolonna	Datu tips	PK	FK	Apraksts	NULL
userID	bigint(20)	✓		unikālais identifikators	
name	varchar(255)			lietotāju vārds	
surname	varchar(255)			lietotāju uzvārds	
usergroupID	bigint(20)		UserGroup(usergroupID)	nosaka lietotāju grupu	
cartID	bigint(20)		Cart(cartID)	nosaka lietotāja izveidoto grozu	
orderID	bigint(20)		Order(orderID)	nosaka lietotāja izveidoto pasūtījumu	
customerID	bigint(20)		Customer(customerID)	nosaka lietotāja piederību klientu grupai	

2.4.1.2. Tabula Customer

Kolonna	Datu tips	PK	FK	Apraksts	NULL
customerID	bigint(20)	✓		unikālais identifikators	
title	varchar(255)			klientu grupas nosaukums	
description	varchar(255)			klientu grupas apraksts	

address	varchar(255)			nosaka klienta adresi pasūtījumam	
shopcategoryID	bigint(20)		ShopCategory(shopcategoryID)	nosaka veikala kategoriju, kas atbilst dotajai klientu grupai	
userID	bigint(20)		User(userID)	nosaka lietotāju, kas pieder klientu grupai	

2.4.1.3. Tabula Shop_category

Kolonna	Datu tips	PK	FK	Apraksts	NULL
shopcategoryID	bigint(20)	✓		unikālais identifikators	
title	varchar(255)			veikala kategorijas nosaukums	
description	varchar(255)			nosaka veikala kategorijas aprakstu	
shopID	bigint(20)		Shop(shopID)	nosaka konkrēta veikala piederību kategorijai	
customerID	bigint(20)		Customer(customerID)	nosaka klienta grupas piederību veikala kategorijai	

2.4.1.4. Tabula Shop

Kolonna	Datu tips	PK	FK	Apraksts	NULL
shopID	bigint(20)	✓		unikālais identifikators	
name	varchar(255)			veikala nosaukums	
description	varchar(255)			apraksts par veikalu	
address	varchar(255)			veikalam piešķirtā adrese	
latitude	decimal(10,8)			nosaka veikala koordinātas: platumu	
longitude	decimal(11,8)			nosaka veikala koordinātas: garumu	
shopcategoryID	bigint(20)		ShopCategory(shopcategoryID)	nosaka, kādai kategorijas pieder veikals	
orderID	bigint(20)		Order(orderID)	nosaka, no kura veikala tiek izveidots pasūtījums	
warehouseID	bigint(20)		Warehouse(warehouseID)	nosaka, kuras noliktavas ir saistītas	

2.4.1.5. Tabula Warehouse

Kolonna	Datu tips	PK	FK	Apraksts	NULL
warehouseID	bigint(20)	✓		unikālais identifikators	
name	varchar(255)			noliktavas nosaukums	
description	varchar(255)			noliktavas apraksts	
address	varchar(255)			noliktavai piešķirtā adrese	
latitude	decimal(10,8)			nosaka noliktavas koordinātas: platumu	
longitude	decimal(11,8)			nosaka noliktavas koordinātas: garumu	
shopID	bigint(20)		Shop(shopID)	nosaka veikalu, kas ir saistīts ar noliktavu	
productID	bigint(20)		Product(productID)	nosaka produktu, kas atrodas noliktavā	

2.4.1.6. Tabula Product

Kolonna	Datu tips	PK	FK	Apraksts	NULL
productID	bigint(20)	✓		unikālais identifikators	

name	varchar(255)			produkta nosaukums	
code	varchar(255)			produkta unikālais kods	
price	decimal(30,8)			produkta cena	
description	varchar(255)			produkta apraksts	
quantity	int			produkta daudzums	
cartID	bigint(20)		Cart(cartID)	nosaka grozu, kurā tika pievienots produkts	
warehouseID	bigint(20)		Warehouse(warehouseID)	nosaka noliktavu, kurā atrodas produkts	

2.4.1.7. Tabula Cart

Kolonna	Datu tips	PK	FK	Apraksts	NULL
cartID	bigint(20)	✓		unikālais identifikators	
created	datetime(6)			nosaka laiku, kurā tika izveidots grozs	
modified	datetime(6)			nosaka laiku, kurā grozs tika izmainīts	X
totalprice	decimal(30,8)			nosaka groza satura pilnu cenu	
productID	bigint(20)		Product(productID)	nosaka produktus, kas ir pievienoti grozā	

customerID	bigint(20)		Customer(customerID)	nosaka klientu, kas izveidoja grozu	
orderID	bigint(20)		Order(orderID)	nosaka, kurš pasūtījums tika izveidots no groza	

2.4.1.8. Tabula User_group

Kolonna	Datu tips	PK	FK	Apraksts	NULL
usergroupID	bigint(20)	✓		unikālais identifikators	
title	varchar(255)			lietotāju grupas nosaukums	
userID	bigint(20)		User(userID)	nosaka piederīgos lietotājus	

2.4.1.9. Tabula Order

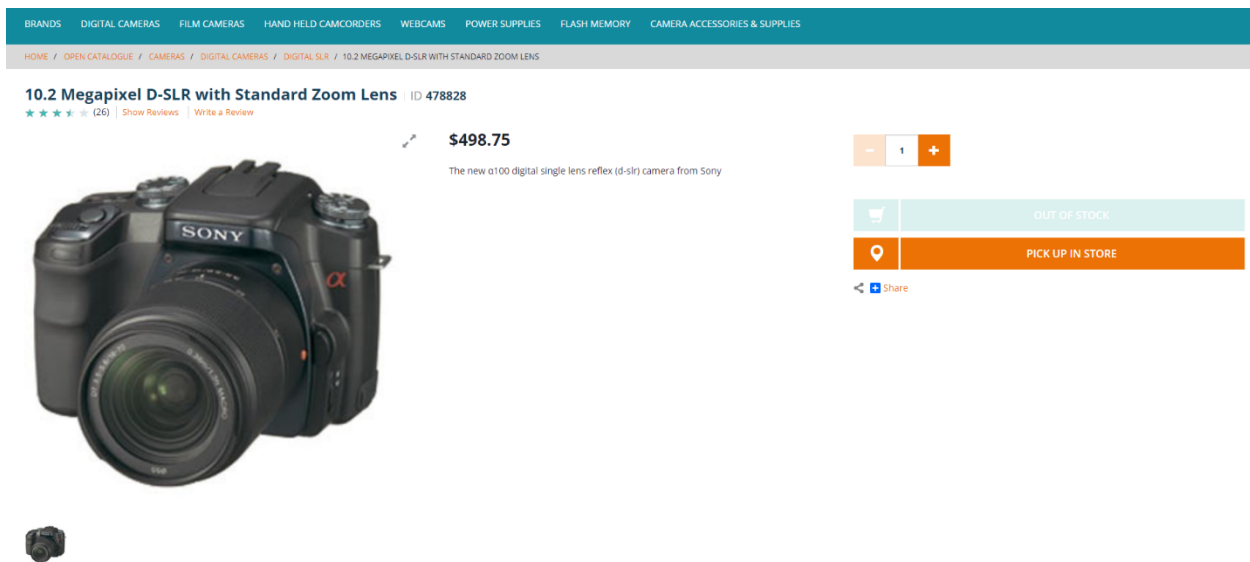
Kolonna	Datu tips	PK	FK	Apraksts	NULL
orderID	bigint(20)	✓		unikālais identifikators	
created	datetime(6)			nosaka laiku, kurā tika izveidots pasūtījums	
modified	datetime(6)			nosaka laiku, kurā pasūtījums tika mainīts	

totalprice	decimal(30,8)			nosaka pasūtījuma pilno cenu	
paymentstatus	bigint(20)			nosaka pasūtījuma apmaksas statusu	
orderstatus	bigint(20)			nosaka pasūtījuma statusu	
cartID	bigint(20)		Cart(cartID)	nosaka grozu, no kura tika izveidots pasūtījums	
userID	bigint(20)		User(userID)	nosaka lietotāju, kas veica pasūtījumu	
shopID	bigint(20)		Shop(shopID)	nosaka veikalu, kurā tika veikts pasūtījums	

2.5. Saskarnes apraksts

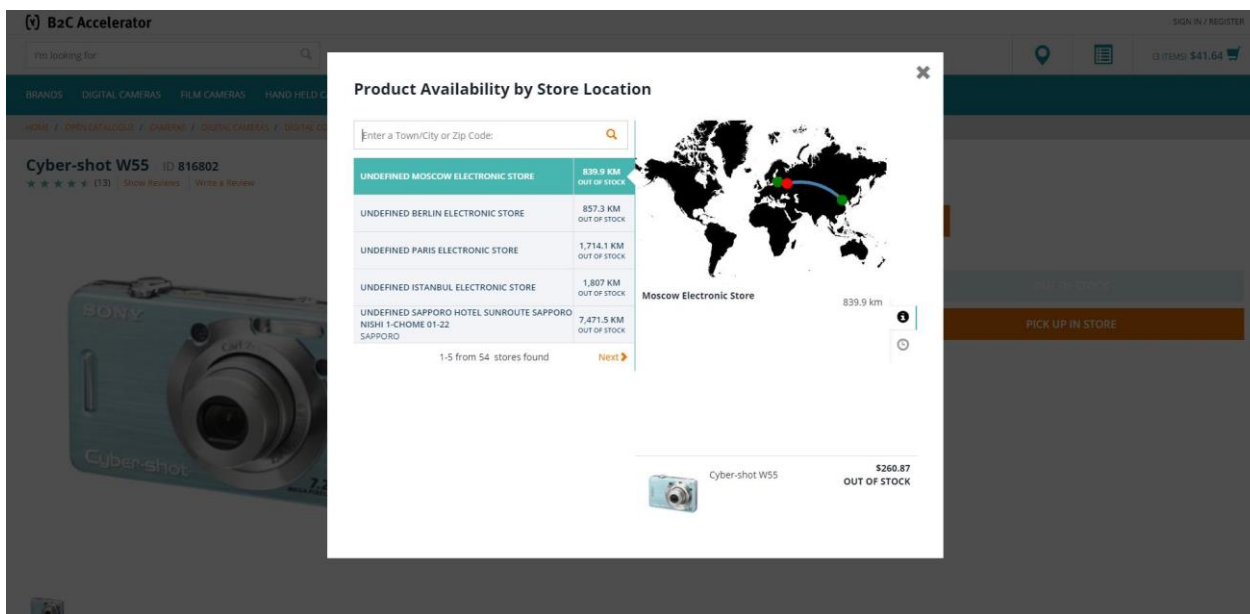
Lietotnes saskarne ir redzama attēlos 2.2., 2.3., 2.4., 2.5. un 2.6.

Attēlā nr. 2.2. ir redzama lietotāja saskarne, kad lietotājs ir atvēris veikala lapu un izvēlējis preci, kas viņam patīk, bet vēl nav izvēlējis, kur to saņemt, noklikšķinot uz PICK UP IN STORE pogas.



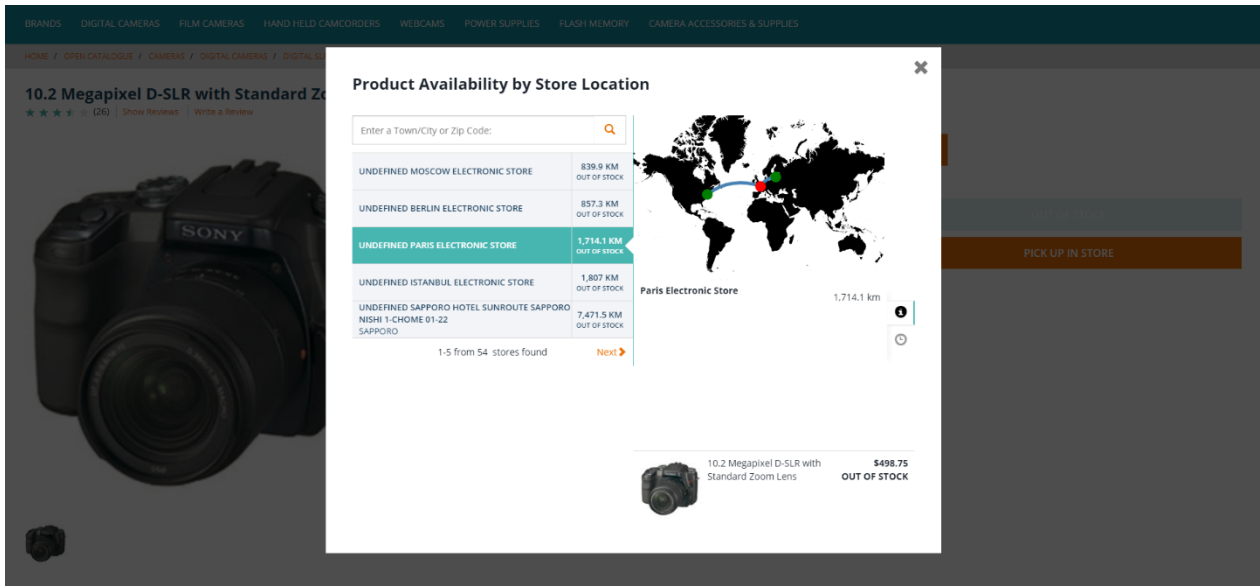
2.2 att.. Lietotāja pamatsaskarne ar izvēlēto produktu

Attēlā 2.3. ir redzama lietotāja saskarne pēc pogas “PICK UP IN STORE” nospiešanas. Lietotājam pēc noklusējuma tiek piedāvāta tuvākā preces atrašanās vieta.



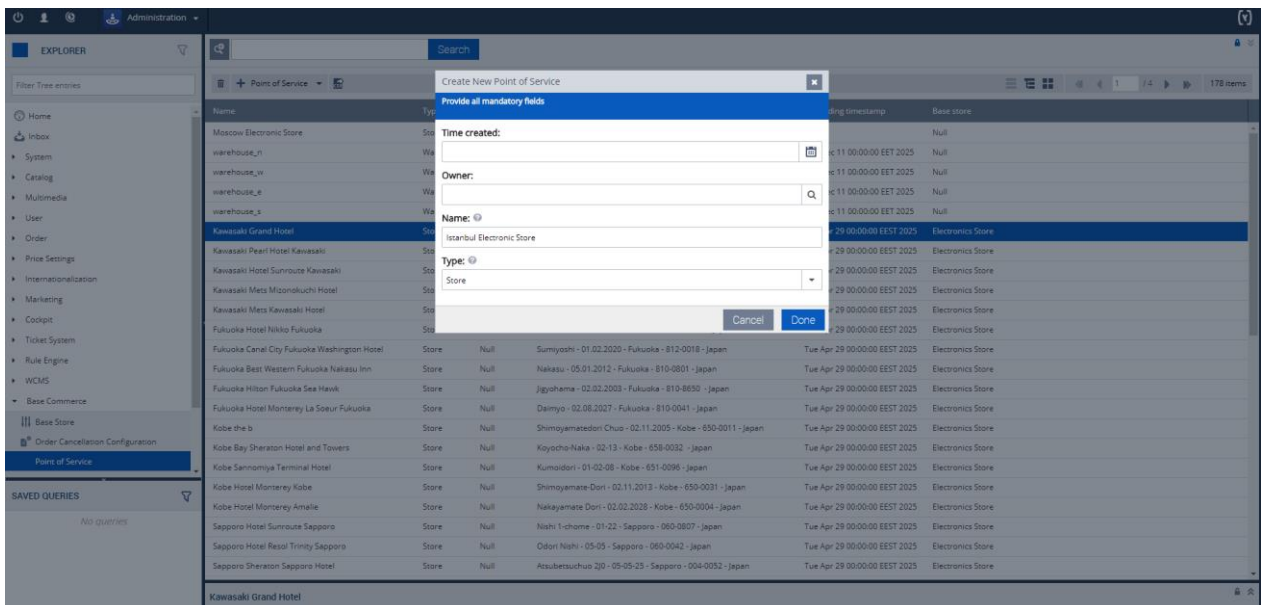
2.3.att. Lietotāju saskarne pēc pogas “PICK UP IN STORE”

Attēlā nr. 2.4. ir redzama lietotāju saskarne, kad lietotājs ir noklikšķinājis uz pogas “PICK UP IN STORE” un ir izvēlējies veikalu, kurā viņš vēlas saņemt preci. Uz ekrāna parādās karte, kurā tiek atzīmēta lietotāju atrašanās vieta, veikala, caur kuru tiks veikta piegāde, atrašanās vieta un produkta noliktavas atrašanās vieta.



2.4.att. Lietotāju saskarne pēc izvēlēta veikala

Attēlā nr. 2.5. un 2.6. ir redzama administratora Backoffice saskarne, kad viņš vēlas pievienot jaunu veikalu vai noliktavu un norādīt tām nepieciešamās koordinātas.



2.5. att. Administrators saskarne ar jaunu izveidotu veikalu

Administration

EXPLORER

Filter Tree entries

- Home
- Inbox
- System
- Catalog
- Multimedia
- User
- Order
- Price Settings
- Internationalization
- Marketing
- Cockpit
- Ticket System
- Rule Engine
- WCMS
- Base Commerce
 - Base Store
 - Order Cancellation Configuration
 - Point of Service**

SAVED QUERIES

No queries

Search

Points of Service 181 items

Name	Type	Description	Address	Geocoding timestamp	Base store
Paris Electronic Store	Store	Null	Null	Null	Electronics Store
Berlin Electronic Store	Store	Null	Null	Null	Electronics Store
Istanbul Electronic Store	Store	Null	Null	Null	Electronics Store
Moscow Electronic Store	Store	Null	Null	Null	Electronics Store
warehouse_n	Warehouse	Null	null - 1-1-22 Kita 7 Jonishi - Seppora - 060-0807 - Japan	Thu Dec 11 00:00:00 EET 2025	Null
warehouse_w	Warehouse	Null	Chuo Ward - 1-11-4 Daimyo - Fukuoka - 810-0041 - Japan	Thu Dec 11 00:00:00 EET 2025	Null
warehouse_e	Warehouse	Null	null - 1-5-16 Takadanobaba - Tokio - 169-0075 - Japan	Thu Dec 11 00:00:00 EET 2025	Null
warehouse_s	Warehouse	Null	Senjara Ward - 2-3-26 Momochihama - Fukuoka - 814-0001 - Japan	Thu Dec 11 00:00:00 EET 2025	Null
Kawasaki Grand Hotel	Store	Null	Myamoto-cho - 06-02 - Kawasaki - 210-0004 - Japan	Tue Apr 29 00:00:00 EEST 2025	Electronics Store
Kawasaki Pearl Hotel Kawasaki	Store	Null	Ogawa cho - 16-02 - Kawasaki - 210-0023 - Japan	Tue Apr 29 00:00:00 EEST 2025	Electronics Store

Istanbul Electronic Store

Location Administration

Geocoding timestamp Latitude Longitude Nearby Store Radius

Documents Assigned Cockpit Item Templates Comments

Dependent catalog versions Source catalog versions

2.6. att. Administratora saskarne, kad pievieno veikala/noliktavas koordināšu datus

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Testēšanas metodika

Lietotnes darbības testēšanai tika izmantota funkcionālā testēšana un vienbtestēšana. Testēšana tika veikta pēc katras funkcijas implementēšanas.

3.2. Testēšanas žurnāls

Nr.	Testa darbība	Sagaidāmais rezultāts	Rezultāts
1.	Nospiež pogu PICK UP IN STORE	Parādās pieejamo veikalu saraksts un tiek ielādēta karte	Izpildās
2.	Nospiež pogu PICK UP IN STORE un sagaida kartes attēlu	Lietotāja, veikala un noliktavas koordinātas pakāpeniski parādās kartē ar punktiem	Izpildās
3.	Nospiež pogu PICK UP IN STORE un sagaida kartes attēlu	Tiek zīmētas arkas, kas savieno koordinātu punktus	Izpildās
4.	Nospiež uz citu veikalu sarakstā	Tiek izdzēsti iepriekšējā veikala koordinātu un ķēdes attēlojumi un tiek attēloti izvēlēta veikala dati	Neizpildās, dažreiz joprojām tiek ielādēti un attēloti iepriekšējā veikala dati, ja uz veikaliem klikšķina ļoti ātri un negaida pilnas ķēdes parādīšanos
5.	Ja Backoffice veikalam nav pievienota noliktava un tiek izsaukta karte	Kartē tiek attēlota un savienota tikai lietotāja atrašanās vieta un veikala atrašanās vieta	Izpildās

6.	Ja lietotāju koordinātas ir atslēgtas un tiek izsaukta karte	Piegādes ķēdes netiek attēlotas. Netiek uzrādītas ne noliktavas koordinātas ne veikalu atrašanās vietas	Izpildās
7.	Ja karte tiek izsaukta izmantojot VPN (Virtuālo privāto tīklu)*	Kartes tiek attēlotas piegādes ķēdes ar lietotāju falsificētām koordinātām.	Neizpildās, tiek attēlots tikai veikalu saraksts ar datiem par attālumu, bet kartē nekas neparādās

*Testēšanai tika izmantota programma Sophos SSL VPN Client un tika testēts, izmantojot Vācijas valsts koordinātas.

4. PROJEKTA ORGANIZĀCIJA

Sakumā projekts tika veidots, par pamatu ņemot ūdenskrituma modeli. Tika plānots, kāda tēma varētu tikt apskatīta projektā, kas šobrīd būtu aktuāls un uz kādas platformas to varētu realizēt.

Tika izvēlēta SAP Hybris Commerce platforma un izlemts, ka nepieciešams izveidot piedāvājuma ķēdes vizualizāciju. Bija nepieciešams laiks, lai izstrādātu projekta koncepciju, saprastu sistēmas pamatfunkcijas, kā arī bija jāapgūst platformas darbības princips.

Nākamais solis bija izstrādāt prasības specifikāciju, kas izstrādes procesā tika rediģēta . Tai sekoja programmatūras projektējuma izstrāde, kas arī tika pielāgota un pārveidota paralēli lietotnes izstrādei.

Pēc izstrādes bija nepieciešams veikt izveidotās programmas testēšanu un pabeigt dokumentāciju.

Koda izstrādes laikā projekta autore tika iepazīstināta arī ar SCRUM metodoloģiju, ko ļoti plaši pielieto autores prakses vietā. Notika biežas tikšanās, un darba vadītājam regulāri tika iesniegtas atskaites par paveikto. Tad arī tika lemts, kādi uzdevumi vai izmaiņas būs jāveic turpmāk.

Lietotne tika izstrādāta SAP Hybris Commerce platformā, izmantojot tai pieslēgto MYSQL datu bāzi.

5. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu kvalitatīvu projekta realizāciju, gan dokumentācijas rakstīšanā, gan programmatūras izstrādes daļā tika izmantotas dažādas saistītas vadlīnijas:

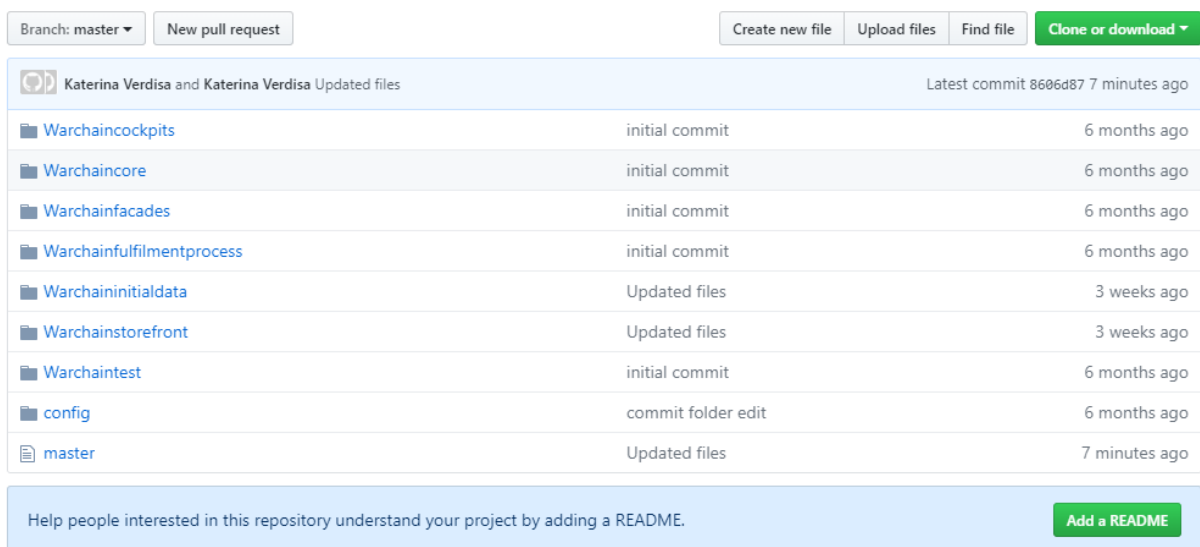
- Lietotnes veidošanai tika ņemtas vērā SAP Hybris Best Practice vadlīnijas¹;
- Rakstot kodu, izmantotas SAP Hybris kodu rakstīšanas standarts²;
- Tika izmantota GitHub git versiju kontroles sistēma;
- Sistēmas koda komentāri tika rakstīti angļu valodā;
- Programmatūras prasību specifikācija un programmatūras projektējuma apraksts tika izstrādāts, balstoties uz Latvijas Valsts standarta.

¹ Best Practice Guidelines - <https://help.hybris.com/6.7.0/hcd/9db9a70f591742b4b70cab3da1ad3ce.html> - [20.05.2018]

² Coding Standards - <https://wiki.hybris.com/display/hybrisALF/Coding+Standards> – [20.05.2018]

6. KONFIGURĀCIJU PĀRVALDĪBA

Projekta versiju pārvaldībai un kvalitātes nodrošināšanai tika izmantots GitHub repositorijs. Tajā tika izveidots projekta autores personīgais zars. Projekta izmaiņas tika ielādētas repositoriņā, izmantojot Eclipse IDE vides iebūvēto funkcionalitāti.



The screenshot shows a GitHub repository interface. At the top, there are navigation buttons: "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". Below this is a header for the repository "Katerina Verdisa and Katerina Verdisa Updated files" with the latest commit hash "8606d87" and time "7 minutes ago". A table lists the repository's contents:

File/Folder	Commit Type	Time Ago
Warchaincockpits	initial commit	6 months ago
Warchaincore	initial commit	6 months ago
Warchainfacades	initial commit	6 months ago
Warchainfulfilmentprocess	initial commit	6 months ago
Warchaininitialdata	Updated files	3 weeks ago
Warchainstorefront	Updated files	3 weeks ago
Warchaintest	initial commit	6 months ago
config	commit folder edit	6 months ago
master	Updated files	7 minutes ago

At the bottom of the repository view, there is a prompt: "Help people interested in this repository understand your project by adding a README." with a green "Add a README" button.

7.1. GitHub glabātie projekta faili.

Savukārt projekta dokumentācija tika glabāta uz ārējā cietajā diska.

7. DARBIETILPĪBAS NOVĒRTĒJUMS

Projekta darbietilpības novērtēšanai tika izmantota eksperta metode. Sākumā ar darba vadītāju tika plānota darba būtība – projekta tēma, kas pašlaik ir aktuāla, un kādus rīkus būs nepieciešams izmantot. Tad tika izvēlēta SAP Hybris Commerce platforma un tika nolemts projekta saturs. Ņemot vērā, ka autorei nebija iepriekšējas pieredzes darbā ar platformu, daudz laika tika atvēlēts tieši tās apgūšanai, dokumentācijas lasīšanai un nepieciešamo programmēšanas valodu apguvei. Tas kopā ar programmēšanu un dokumentācijas rakstīšanu aizņēma aptuveni trīs personmēnešus.

Prasību specifikācijas izveidei tika atvēlētas divas nedēļas, tomēr tā aizņēma ilgāku laiku, jo projekta rakstīšanas laikā tika mainītas kartes izvēles un tās attēlošanas realizācija.

Programmas koda rakstīšana atšķirībā no dokumentācijas aizņēma mazāk laika. Tomēr jau testēšanas laikā tika pamanītas kļūdas, kas nozīmē, ka bija jāvelta vairāk laika.

Apraksts	Plānotais laiks	Reālais patērētais laiks
Uzdevuma izpēte	3 dienas	3 dienas
Prasību specifikācijas izveide un apstiprinājuma iegūšana	14 dienas	21 dienas
Izmantojamās vides un valodas apguve	30 dienas	30 dienas
Programmas koda rakstīšana	60 dienas	42 dienas
Testēšana	2 dienas	2 dienas

SECINĀJUMI

Sap Hybris produktu uzglabāšanas un piedāvājuma ķēdes lietotne tika izstrādāta pēc programmatūras prasību specifikācijas projektējuma apraksta.

Pateicoties šim projektam, autore guva lielu pieredzi darbā ar populāru e-kommercijas izstrādes platformu SAP Hybris Commerce. Neskatoties uz to, ka projekta beigās tika izstrādāts piegādes ķēdes attēlošanas modulis, lielāko daļu projekta laika aizņēma nevis programmas koda rakstīšana, bet iepazīšanās ar platformu un tās darba principiem. Projekta realizācijai bija nepieciešams arī paaugstināt autores iemaņas programmēt JavaScript valodā.

Grūtības, strādājot pie projekta, sagādāja tieši koordinātas izvadīšana uz kartes, jo platformas UI tiek realizēts ar colorbox bibliotēkas palīdzību un, lai saprastu to ielādēšanas kārtību, bija nepieciešams ilgi papētīt sistēmu, lai atrastu pareizo vietu, kur pievienot nepieciešamo kodu. Grūtības sagādāja arī ģeogrāfisko koordinātu attēlošana uz plāksnēm (bija nepieciešams izmantot ģeogrāfiskās projekcijas, lai pareizi attēlotu nepieciešamās koordinātas uz plaknes).

Nākotnē tiek plānots realizēt šo lietotni reālos klienta projektos ar reāliem datiem. Ir plānots lietotnei pievienot funkciju, kas izrēķinās iespējamo piegādes laiku un kartes tālummaiņas īpašības. Alternatīvā ir plānots realizēt šīs lietotnes funkcijas, izmantojot Google Maps / Open Street Maps kartes, lai klientiem sniegtu plašāku izvēli ģeogrāfisko karšu bibliotēku izvēlē.

IZMANTOTĀ LITERATŪRA

1. LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis, Latvijas Valsts standarts.
2. LVS 72:1996 Ieteicamā prakse PPA aprakstīšanai, Latvijas Valsts standarts.
3. JavaScript un HTML dokumentācija - <http://www.w3schools.com/>
4. Ar programmēšanu saistīta diskusiju vietne: Stack Overflow - <http://stackoverflow.com/>
5. Best Practice Guidelines -
<https://help.hybris.com/6.7.0/hcd/9db9a70f591742b4b70cab3da1ad3ce.html>
6. Coding Standards - <https://wiki.hybris.com/display/hybrisALF/Coding+Standards>
7. Backoffice Administration Cockpit - <https://www.hybris.com/en/downloads/product-collateral/backoffice-administration-cockpit/253>
8. Kā atšķiras UX no UI? - <https://www.linkedin.com/pulse/k%C4%81-at%C5%A1%C4%B7ir%C4%81s-ux-ui-armands-antons>
9. Konceptija par elektronisko komerciju - <https://www.vestnesis.lv/ta/id/20451>

PIELIKUMI

Pielikums nr.1. acc.supplychain.js

```
1 var svg, projection, arcs, points;
2
3 ACC.supplychain = {
4   initD3Map : function(mapContainer) { //map initialization
5     svg = d3.select(mapContainer),
6     width = svg.attr("width"),
7     height = svg.attr("height");
8
9     projection = d3.geoMercator()
10    .scale(width / (2 * Math.PI)) //creating map projection
11    .translate([width / 2, height / 1.5]);
12
13    var path = d3.geoPath() //add projection on the map
14    .projection(projection);
15
16    d3.json("/Marchainstorefront/_ui/responsive/common/js/d3data/world-50m.json", function(error, world) { //loading country borders
17      if (error) throw error;
18
19      svg.insert("path") //adding filled countries map
20      .datum(topojson.feature(world, world.objects.land))
21      .attr("class", "land")
22      .attr("d", path);
23
24      svg.insert("path") //adding countries borders
25      .datum(topojson.mesh(world, world.objects.countries, function(a, b) {
26        return a !== b;
27      }))
28      .attr("class", "boundary")
29      .attr("d", path);
30    });
31  },
32
33  waitForMapInitToPerformAction: function(action) { //function wait for map upload, it is recursive and recalls itself
34    if ($(".land").length) { // in period 300 milliseconds and only then begins action
35      action();
36    } else {
37      setTimeout(function(){ACC.supplychain.waitForMapInitToPerformAction(action);}, 300);
38    }
39  },
40
41  clearMapFromSupplyChains: function(){ //function of clearing arcs and points
42    svg.selectAll(".supply-chain")
43    .transition()
44    .duration(500)
45    .style("opacity", 0)
46    .remove();
47  },
48
49  drawArc: function(arcOrigin, arcDestination, firstArc, lastArc) { //arc drawing function
50    var coordinates = [
51      projection(arcOrigin),
52      projection(arcDestination)
53    ];
54
55    var line = arcs.append("path")
56    .datum(coordinates)
57    .attr("d", function(c) {
58      var d = {
59        source: c[0],
60        target: c[1]
61      };
62      var dx = d.target[0] - d.source[0],
63          dy = d.target[1] - d.source[1],
64          dr = Math.sqrt(dx * dx + dy * dy);
65      horizontalShape = (dx < 0) ? "0 " : "1 ";
66      return "M" + d.source[0] + "," + d.source[1] + "A" + dr + "," + dr +
67      " 0 0," + horizontalShape + d.target[0] + "," + d.target[1];
68    })
69    .style("stroke", "steelblue")
70    .style("stroke-width", 5)
71    .style("fill", "none")
72    .transition()
73    .duration(3000)
74    .attrTween("stroke-dasharray", function() {
75      var len = this.getTotalLength();
76      return function(t) {
77        return (d3.interpolateString("0," + len, len + ",0"))(t)
78      };
79    })
80    .on('end', function(d) {
81      var c = coordinates[1];
82      points.append('circle')
83      .attr('cx', c[0])
84      .attr('cy', c[1])
85      .attr('r', 0)
86      .style('fill', lastArc ? 'green' : 'red')
87      .style('fill-opacity', '0.7')
```

```

88     .transition()
89     .duration(1000)
90     .attr('r', 20)
91     .on('end', function(d) {
92         d3.select(this)
93             .transition()
94             .duration(1000)
95             .attr('r', 8)
96             .style('fill-opacity', '1');
97     });
98 });
99
100 if (firstArc) { //check if it is the first arc, then adding a point at the start
101     var c = coordinates[0];
102     points.append('circle')
103         .attr('cx', c[0])
104         .attr('cy', c[1])
105         .attr('r', 0)
106         .style('fill', lastArc ? 'green' : 'green')
107         .style('fill-opacity', '0.7')
108         .transition()
109         .duration(1000)
110         .attr('r', 20)
111         .on('end', function(d) {
112             d3.select(this)
113                 .transition()
114                 .duration(1000)
115                 .attr('r', 8)
116                 .style('fill-opacity', '1');
117         });
118     }
119 },
120
121 chainLoop: function(i, supplyChain, delay) { //function of all chainloop drawing
122     setTimeout(function () {
123         if (i < supplyChain.length) {
124             ACC.supplychain.drawArc(supplyChain[i-1], supplyChain[i], (i == 1) ? true : false, (i+1 == supplyChain.length) ? true : false);
125             i++;
126             ACC.supplychain.chainLoop(i, supplyChain, delay);
127         }
128     }, (i == 1) ? 0 : delay)
129 }
130 }

```

Pielikums nr.2. acc.pickupinstore.js

```
428     var supplyChain = [];  
429     var selectedStore = $(this);  
430  
431     if (navigator.geolocation) {  
432         navigator.geolocation.getCurrentPosition(function(position) {  
433             //checking if user's geolocation is turn on or not  
434             //getting user's current position  
435             if (position.coords.latitude != "" && position.coords.longitude != "") {  
436                 supplyChain.push([position.coords.latitude, position.coords.longitude]);  
437             }  
438             if (selectedStore.data("store-lat") != "" && selectedStore.data("store-long") != "") {  
439                 supplyChain.push([selectedStore.data("store-long"), selectedStore.data("store-lat")]);  
440             }  
441             if (selectedStore.data("warehouse-lat") != "" && selectedStore.data("warehouse-long") != "") {  
442                 supplyChain.push([selectedStore.data("warehouse-long"), selectedStore.data("warehouse-lat")]);  
443             }  
444         });  
445     }  
446     } else {  
447         //if geolocation isn't turned on  
448         supplyChain = [[selectedStore.data("store-long"), selectedStore.data("store-lat")],  
449             [selectedStore.data("warehouse-long"), selectedStore.data("warehouse-lat")]];  
450         //use store geolocation  
451         //use warehouse geolocation  
452     }  
453     if (supplyChain != null) {  
454         ACC.supplychain.waitForMapInitToPerformAction(function(){  
455             // if there are datas, then the map is being drawn  
456             //wait till map will be loaded, and do next steps  
457             arcs = svg.append('g').attr("class", "supply-chain arcs");  
458             points = svg.append('g').attr("class", "supply-chain points");  
459             //add graphic collection with arcs and points  
460             setTimeout(function(){ACC.supplychain.chainLoop(1, supplyChain, 3000);}, 100);  
461             //draw chainloop for 100 milliseconds  
462         });  
463     }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```

Kvalifikācijas darbs „*SAP Hybris produktu uzglabāšanas un piedāvājuma ķēdes lietotne*”
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: *Katerīna Verdiša* _____ .05.2018.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs/a: *Mg. comp. sc. Jurijs Nikolajevs* _____ .05.2018.

Recenzents: *M.soc.zin. Lauris Raipulis*

Darbs iesniegts 28.05.2018.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2018. prot. Nr. _____

Komisijas sekretārs(-e): _____