



**LATVIJAS UNIVERSITĀTES  
DATORIKAS FAKULTĀTE**

**Android lietotne - Draudzīgie maksājumi  
Kvalifikācijas darbs**

**Darba autors: Egīls Paulovskis**

**Stud. apl. Nr: ep14016**

**Darba vadītājs: doc. Imants Gorbāns**

**RĪGA 2018**

## **Anotācija**

Kvalifikācijas darba mērķis ir izveidot Android lietotni ar nosaukumu "Friendly Payments". Autors uzskata, ka ar tās palīdzību lietotājs varētu labāk pārskatīt savu finansiālo līdzekļu lietošanas paradumus un veidot pirkumu atskaites CSV formātā, kā arī dalīties ar attiecīgo pirkumu informāciju, to izvadot, kā teksta ziņu.

Lietotnē iespējams veidot pirkumus, kur katrs pirkums sastāv no apraksta un precēm, kuras jāizveido lietotājam, kā arī kontaktiem, kurus iespējams pievienot, ja tas bijis kopīgs pirkums. Pēc pirkuma izveides, lietotājs to var apskatīt un labot pēc paša ieskatiem, mainot tā aprakstu, vai pievienojot, labojot vai dzēšot tajā esošās preces un vai kontaktus. Lietotājs var arī pirkuma informāciju aplūkot soli dziļāk, apskatot tā priekšskatu, kas parādās teksta veidā un ir nokopējams un vai nosūtāms izmantojot ierīcē esošās lietotnes, kā arī eksportējams CSV formātā.

Lietotnes dati tiek uzglabāti lokāli, lietotāja ierīcē, SQLite datubāzē. Lietotne izstrādāta ar Android Studio palīdzību, produkta izstrādē izmantotas programmēšanas valodas Java un XML.

**Atslēgvārdi:** Android, SQLite, GitHub, Android Studio.

## **Abstract**

### **ANDROID APPLICATIONS - FRIENDLY PAYMENTS**

The aim of this qualification work is to create an Android application called “Friendly Payments”. According to the author, this would help the user to better review their use of financial resources and to create purchase reports in CSV format, as well as to share information about the purchase in question, as a text message.

In the app, you can create purchases, where each purchase consists of a description and the items to be created by the user, as well as contacts that the user can add in case of a joint purchase. After creating a purchase, the user can view and edit it at his own discretion by changing its description, or by adding, editing or deleting the items and contacts in it. The user can also view the information of the purchase a step further by looking at its preview, which appears in plain text and is copyable and can be transmitted using the applications that are already available in the device, the purchase can be exported in CSV format as well.

The data of the application is stored locally, on the user device, in a SQLite database. The application is designed with Android Studio, and the programming languages used were Java and XML.

**Keywords:** Android, SQLite, GitHub, Android Studio.

# SATURS

<b>Anotācija</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>SATURS</b>	<b>3</b>
<b>Vārdnīca</b>	<b>6</b>
<b>Ievads</b>	<b>7</b>
<b>1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA</b>	<b>8</b>
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	8
1.2.1. Produkta perspektīva	8
1.2.2. Produkta funkcijas	9
1.2.3. Lietotāju raksturiezīmes	9
1.2.4. Vispārējie ierobežojumi	9
1.2.5. Pieņēmumi un atkarības	9
1.3. Funkcionālās prasības	10
1.3.1. Pievienot kontaktu	10
1.3.2. Rediģēt kontaktu	10
1.3.3. Dzēst kontaktu	10
1.3.4. Pievienot preci	10
1.3.5. Rediģēt preci	11
1.3.6. Dzēst preci	11
1.3.7. Pievienot pirkumu	11
1.3.8. Rediģētu pirkumu	11
1.3.9. Dzēst pirkumu	11
1.3.10. Pirkumu apskate	12
1.3.11. Pirkumu eksportēšana	12
1.3.12. Izmaksu kopsavilkumu apskate	12
1.3.13. Paroles pārbaude	12
1.4. Ārējās saskarnes prasības	12

1.4.1. Lietotāja saskarne	12
1.5. Nefunkcionālās prasības	12
1.5.1 Datu drošības prasības	12
<b>2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS</b>	<b>13</b>
2.1 Ievads	13
2.1.1 Nolūks	13
2.1.2 Darbības sfēra	13
2.1.3 Saistība ar citiem dokumentiem	13
2.1.4 Pārskats	13
2.2. Dekompozīcijas apraksts	13
2.2.1. Pievienot kontaktu	13
2.2.2. Rediģēt kontaktu	14
2.2.3. Dzēst kontaktu	14
2.2.4. Pievienot preci	14
2.2.5. Rediģēt preci	14
2.2.6. Dzēst preci	14
2.2.7. Pievienot pirkumu	15
2.2.8. Rediģēt pirkumu	15
2.2.9. Dzēst pirkumu	15
2.2.10. Pirkumu apskate	15
2.2.11. Pirkumu eksportēšana	15
2.2.12. Izmaksu kopsavilkumu apskate	16
2.2.13. Paroles pārbaude	16
2.3. Datu atkarības	16
2.3.1. Datu atkarības	16
2.3.2. Datu bāzes tabulu detalizēts apraksts	17
2.3.2.1. Tabula “Items”	17
2.3.2.2. Tabula “Contact”	17
2.3.2.3. Tabula “Purchase”	17
2.3.2.4. Tabula “HelperTable”	18
2.4. Lietotnes skatu relācijas	18
2.4.1. Funkciju kopa “Jauns pirkums”	19
2.4.1.1. Jauns pirkums	20
2.4.1.2. Pievienot / Rediģēt preci	20
2.4.1.3. Pievienot / Rediģēt kontaktu	20

2.4.1.4. Pirkums	20
2.4.2. Funkciju kopa “Vēsture”	21
2.4.2.1. Vēsture	22
2.4.2.2. Pirkums	22
2.4.2.3. Rediģēt pirkumu	22
2.4.2.4. Priekšskatījums	22
2.4.3. Funkciju kopa “Kontakti”	23
2.4.3.1. Kontakti	23
2.4.3.2. Pieveinot kontaktu	23
2.4.3.3. Rediģēt kontaktu	23
2.4.4. Kopsavilkums	24
2.4.4.1. Kopsavilkums	24
2.4.4.2. Kopsavilkuma priekšskats	24
<b>3. TESTĒŠANA</b>	<b>25</b>
3.1. Saistības ar citiem dokumentiem	25
3.2. Funkciju darbības skaidrojumi un pieņemšanas kritēriji	25
3.2.1. Preču testēšana	25
3.2.2. Kontaktu testēšana	26
3.2.3. Pirkumu testēšana	26
3.2.4. Drošības / Kopsavilkuma testēšana	27
3.3. Kļūdu novēršana	27
<b>4. PROJEKTA ORGANIZĀCIJA</b>	<b>28</b>
4.1. Konfigurāciju pārvaldība	28
<b>5. KVALITĀTES NODROŠINĀŠANA</b>	<b>29</b>
<b>6. DARBIETILPĪBAS NOVĒRTĒJUMS</b>	<b>30</b>
6.1 Projekta darbietilpības plānošana	30
6.2 Secinājumi	30
<b>7. NOBEIGUMS</b>	<b>31</b>
<b>8. IZMANTOTIE AVOTI</b>	<b>32</b>
<b>10. PIELIKUMI</b>	<b>33</b>
10.1. Programmatūras koda piemēri.	33
<b>Dokumentārā lapa</b>	<b>35</b>

## Vārdnīca

**PPA** – Programmatūras projektējuma apraksts. Dokumentēts “Android Lietotne - Friendly Payments” tehniskais projektējums.

**PPS** - Programmatūras prasību specifikācija. Dokumentēta “Android Lietotne - Friendly Payments” funkcionalitāte un citas prasības.

**SQL** - vaicājumu valoda, kas paredzēta datu manipulēšanai relāciju datubāžu pārvaldības sistēmās.

**SQL injekcija** - datorsistēmu ielaušanās veids, kurā uzbrucējs ar klienta tiesībām mēģina piekļūt datubāzei.

**XML** – Extensible Markup Language. Plaši izmantota datu strukturizēšanas iezīmēšanas valoda.

**SQLite** – atvērta pirmkoda datu bāzes vadības sistēma.

**DB** – Datu Bāze jeb datu krātuve, kur glabāsies lielākā daļa no sistēmā ievietotās informācijas.

**CRUD** – Create, Read, Update, Delete (Akronīms, četras sistēmas pamat funkcijas – izveidot, lasīt, atjaunināt, dzēst).

**MVC** – Model View Controller arhitektūra ir veids, kā strukturizēt projekta kodu, lai sadalītu to trīs pamata komponentēs.

**View (Skats)** – lietotājam redzamās daļas un iezīmēšanas valodu koda ģenerēšanas daļa.

**SVG** - Scalable Vector Graphics XML formātā veidoti attēli, kuriem nesamazinas izšķirtspēja.

**OS** - Operētājsistēma, šajā darbā, attiecināts uz Android.

**CSV** - Comma Separated Values, datu uzglabāšanas tips, kur katra kolona atdalīta ar komatu.

## Ievads

Kvalifikācijas darba tēma tika izvēlēta autora paša interešu ieskatos, ņemot vērā iepriekš esošās zināšanas un vēlmi izveidot kādu lietotni. Lietotne “Friendly Payments” tika veidota pēc “Agile” izstrādes metodes.

Lietotnes “Friendly Payments” mērķis ir lietotājam sniegt iespēju veidot pirkumus, kuri darbotos kā atskats lietotāja finansiālo līdzekļu lietotšanas paradumiem, jeb maksājumu / pirkumu grāmatiņa, kur lietotājs var apskatīt iepriekš veidotus pirkumus, tos rediģēt, dzēst, kā arī izvadīt to informāciju teksta ziņas vai CSV formātā.

Lietotne veidota izstrādes vidē Android Studio, ar papildus attēlu resursu failiem no vietnes TheNounProject.

# 1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

## 1.1. Ievads

### 1.1.1. Nolūks

Programmatūras prasību specifikācijā (PPS), detalizēti tiek aprakstītas programmatūras prasības, visas programmatūras funkcijas, to pieejamība un sistēmas ierobežojumi.

### 1.1.2. Darbības sfēra

Lietotne tika izveidota, lai lietotājs varētu ērtāk noformēt maksājumus, kā arī atskatīties uz to, kā ir lietoti finansiālie līdzekļi. Tā atvieglo nepieciešamās informācijas atrašanu, kā piemēram bankas konta numurs, kurš, pēc autora domām, ir pietiekami sarežģīts, lai lietotājs to nemēģinātu apgūt no galvas.

### 1.1.3 Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standartu LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” un LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai” prasības.

### 1.1.4. Pārskats

Šis dokuments ir organizēts šādās daļās:

- Dokumenta ievads – raksturots dokumenta nolūks, darbības sfēra un saistība ar citiem dokumentiem, kā arī aprakstīta dokumenta struktūra;
- Vispārējais apraksts – aprakstīti produkta vispārējie faktori, kas ietekmē produktu un tā prasības (produkta perspektīva, produkta funkcijas, lietotāja raksturiezīmes, vispārējie ierobežojumi. Pieņēmumi un atkarības);
- Funkcionālās prasības – raksturotas produkta funkcionālās prasības, specificējot funkcijas ievadi, apstrādi un izvadi;
- Ārējās saskarnes prasības – raksturo lietotāju saskarnes prasības;
- Nefunkcionālās prasības – raksturotas produkta veiktspējas, datu drošības, pieejamības prasības;

## 1.2. Vispārējs apraksts

### 1.2.1. Produkta perspektīva

Lietotnes izveide attiecināta pārsvarā uz studentiem, kuri mēdz kopīgi veidot pasākumus vai svētkus. Tā nosacīti darbojas kā grāmatvedības programma, ar lietotnes palīdzību tiek atvieglota izmaksu

uzskaitīšana, izdalīšana. Papildus, ar lietotnes palīdzību, iespējams veikt atskatu uz to kā lietoti finansiālie līdzekļi pēc kopsavilkuma izveides, kas tiek noteikts ar dažādiem kritērijiem.

### **1.2.2. Produkta funkcijas**

Izstrādātajā sistēmā tiek nodrošināta sekojoša funkcionalitāte:

- Pievienot kontaktu;
- Rediģēt kontaktu;
- Dzēst kontaktu;
- Pievienot preci;
- Rediģēt preci;
- Dzēst preci;
- Pievienot pirkumu;
  - Pievienot vai rediģēt preci;
  - Pievienot vai rediģēt kontaktu;
  - Pievienot aprakstu;
- Rediģēt pirkumu;
- Dzēst pirkumu;
- Iepriekš veiktu pirkumu apskate;
- Pirkumu eksportēšana;
- Izmaksu kopsavilkumu apskate;
- Paroles pārbaude

### **1.2.3. Lietotāju raksturiezīmes**

Lietotnei jābūt viegli saprotamai, lai jebkura persona ar mobilo ierīču lietošanas pamatiemaņām spētu to izmantot.

### **1.2.4 Vispārējie ierobežojumi**

Lietotnes izmantošanu nodrošina mobilā ierīce, kurai ir Android OS, ar 4.1.0 (“Jelly Bean”), vai augstāku versiju.

### **1.2.5. Pieņēmumi un atkarības**

Lietotājs nodrošina prasībām atbilstošu ierīci lietotnes izmantošanai. Lietotāji nemēģinās manuāli veikt SQL injekcijas, kas ietekmētu to datu drošību.

## **1.3. Funkcionālās prasības**

### **1.3.1. Pievienot kontaktu**

Kontakts tiek uzglabāts tabulā Contacts, kura sastāv no laukiem:

- ContactID;
- Name;
- Surname;
- Number;
- Email;
- BankAccountNumber;
- BankName;
- Password;

Kontakti tiek veidoti manuāli ievadot informāciju attiecīgā lietotnes skatā. No kontaktiem tiek izmantots vārds un uzvārds, kontaktu atlasei. Bankas konta numurs un nosaukums ir nepieciešams tikai pašam lietotājam, kas nav obligāts. Pārējiem pirkuma kontaktiem jeb dalībniekiem šo informāciju ievadīt nav iespējams. Parole nepieciešama lietotājam, datu nodrošināšanai un lietotnes piekļuvei.

### **1.3.2. Rediģēt kontaktu**

Rediģēšanai, kontakts tiek atrasts caur lietotājam pieejamu sarakstu, kurš uzglabā izveidotos vai pirkumam pievienotos kontaktus. Rediģēšanas laikā, lietotājs var mainīt visus pieejamos tabulas laukus attiecīgajam kontaktam, izņemot tā primārās atslēgas vērtību (ContactID).

### **1.3.3. Dzēst kontaktu**

Kontaktu iespējams dzēst kontakta rediģēšanas laikā, ja tas tiek rediģēts caur pirkumu, kontakts tiek noņemts kā dalībnieks pie pirkuma, ja kontakts tiek rediģēts caur kontaktu skatu, tad tas tiek izdzēsts no visām tabulām (pirkumiem) kurās tas atrodas lietotnē.

### **1.3.4. Pievienot preci**

Prece tiek uzglabāta atsevišķā tabulā un sastāv no šādiem laukiem:

- ItemID;
- Name;
- Price;
- Quantity;
- Category;

ItemID, preces identifikācijas numurs, Name, preces nosaukums, Price, preces vienas vienības cena, Quantity, preces vienību skaits, Category, kategorija, kas tiek izmantota izmaksu kopsavilkumu veidošanai.

### **1.3.5. Rediģēt preci**

Prece tiek atlasīta, caur lietotājam pieejamu sarakstu, kurā atrodas preces, un uzspiežot uz attiecīgo pirkumu. Atveroties precei, lietotājs var mainīt visus tās laukus, izņemot tās identifikācijas numuru (ItemID).

### **1.3.6. Dzēst preci**

Preci iespējams izdzēst, tās rediģēšanas laikā.

### **1.3.7. Pievienot pirkumu**

Pirkums sastāv no divām tabulām ar attiecīgiem laukiem:

#### **Purchases**

- PurchaseID;
- Description;

Pirmā tabula satur PurchaseID, pirkuma identifikācijas numurs un Description, pirkuma apraksts, jeb iemesls, kādēļ tas ticis veidots.

#### **HelperTable**

- PurchaseID;
- ItemID;
- ContactID;

Otrā tabula satur iepriekš veidoto objektu identifikācijas numurus, kuri tiek izmantoti datu atlasei un ziņas veidošanai, kura tiktu izmantota pēc lietotāja izvēles. Laukumu nozīme apskatāma iepriekšējos punktos.

### **1.3.8. Rediģētu pirkumu**

Pirkums tiek rediģēts, atverot to no pirkumu vēstures, kuri tiek uzglabāti attiecīgā lietotnes skatā vai no attiecīgā pirkuma priekšskata. Rediģēšanā iespējams rediģēt preces, kontaktus, kā arī pirkuma aprakstu.

### **1.3.9. Dzēst pirkumu**

Pirkumu iespējams dzēst atverot to caur pirkumu vēsturi no attiecīga lietotnes skata.

### **1.3.10. Pirkumu apskate**

Pirkumus iespējams apskatīt no sākuma skata Vēsture. Pirkumi tiek uzglabāti režģa skatā, kur katrs pirkums tiek atdalīts pēc tā kārtas numura un apraksta. Pirkuma veidotā ziņa apskatāma priekšskatā - "Preview".

### **1.3.11. Pirkumu eksportēšana**

Pēc pirkuma izveides, lietotājs spēj izvadīt ziņu, kas veidojas no attiecīgajam pirkumam veidotā apraksta, precēm, kā arī dalībniekiem, ja tādi tajā ir. Iespēja eksportēt pirkuma ziņu CSV faila formātā.

### **1.3.12. Izmaksu kopsavilkumu apskate**

Lietotājs spēj apskatīt kā ir tērējis savus finansiālos līdzekļus, veicot speciālu atlasu izmantojot dažādus laukus no izveidotajiem pirkumiem.

### **1.3.13. Paroles pārbaude**

Lietotājam jāievada parole lietotnes izmantošanai. Skat. 1.5.1. Datu drošības prasības.

## **1.4. Ārējās saskarnes prasības**

### **1.4.1. Lietotāja saskarne**

Lietotāja saskarne veidota ar Android Studio iebūvētā Layout Editor palīdzību, kas atvieglo lietotāja saskarnes modificēšanu, kā arī piedāvā automātiski modificēt skata izkārtojumu attiecīgi pret ierīces izmēra.

## **1.5. Nefunkcionālās prasības**

### **1.5.1 Datu drošības prasības**

Saglabātie dati ir pieejami tikai ierīces īpašniekam, jo tie tiek uzglabāti lokāli SQLite datubāzē, kura ir sasniedzama tikai izmantojot lietotni. Priekš datu pastiprinātas aizsardzības, tiek plānots ieviest paroles pieprasījumu ik pēc notikta laika posma, tā tiek uzstādīta pirmajā lietotnes uzsākšanas reizē, ar iespēju to nomainīt caur lietotāja informācijas rediģēšanu, kura piekļuvei arī nepieciešams ievadīt paroli.

## 2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

### 2.1 Ievads

#### 2.1.1 Nolūks

Programmatūras prasību specifikācijā (PPS), detalizēti tiek aprakstītas programatūras prasības, tiek aprakstītas visas programmatūras funkcijas, to pieejamība un sistēmas ierobežojumi.

#### 2.1.2 Darbības sfēra

Sistēmas “Darbības sfēra” ir aprakstīta 1.1.2 nodaļā Programmatūras prasību specifikācijā.

#### 2.1.3 Saistība ar citiem dokumentiem

Šī dokumenta veidošanā tika izmantotas LVS 72:1996 "Ieteicamā prakse programmatūras projektējuma aprakstīšanai" standarta vadlīnijas. Dokumenta veidošanā izmantota augstāk minētā PPS, tāpēc dokumentu ieteicams lietot kopā ar PPS.

#### 2.1.4 Pārskats

Lietotnes “Friendly Payments” PPA sastāv no 4 nodaļām:

- Ievads – aprakstīts dokumenta nolūks, norādīts, kur atrast produkta darbības sfēru un aprakstīta saistība ar citiem dokumentiem, kā arī parādīta dokumenta struktūra;
- Dekompozīcijas apraksts – Uzskaitītas un aprakstītas datu bāzes tabulas (specificējot datu nosaukumus, tipus, aprakstus, saistītās atslēgas);
- Datu atkarības – attēlotas datu atkarības ER modelī;
- Lietotnes skatu relācijas - Aprakstītas lietotnes saskarnes skatu saistības;

## 2.2. Dekompozīcijas apraksts

### 2.2.1. Pievienot kontaktu

Mērķis	Izveidot jaunu kontaktu.
Funkcija	Izveido ierakstu tabulā ”Contacts”, attiecīgi no kurienes tas tiek izsaukts, pievieno to arī pirkumam, palīgtabulā “HelperTable”. Izmanto skatu “Add Contact”.
Darbību loks	Funkcija pieejama skatos : Contacts, New Purchase, Edit Purchase.

### 2.2.2. Rediģēt kontaktu

Mērķis	Rediģēt esoša kontakta informāciju.
Funkcija	Atvērt esošu kontaktu un izmainīt tā informāciju. Izmanto skatu "Edit Contact".
Darbību loks	Funkcija pieejama no skata "Edit Contact".

### 2.2.3. Dzēst kontaktu

Mērķis	Dzēst eksistējošu kontaktu.
Funkcija	Izdzēst kontakta informāciju no tabulas "Contacts", kā arī no tabulas "HelperTable". Izmanto skatu "Edit Contact".
Darbību loks	Funkcija pieejama no skata "Edit Contact."

### 2.2.4. Pievienot preci

Mērķis	Izveidot jaunu pirkuma preci.
Funkcija	Izveidot preci tabulā "Items" un pievienot to pirkumam palīgtabulā "HelperTable". Izmanto skatu "Add Item".
Darbību loks	Funkcija pieejama skatos : New Purchase, Edit Purchase.

### 2.2.5. Rediģēt preci

Mērķis	Rediģē eksistējošu preci.
Funkcija	Atvērt esošu preci un izmainīt tās informāciju. Izmanto skatu "Edit Item".
Darbību loks	Funkcija pieejama no skata "Edit Item".

### 2.2.6. Dzēst preci

Mērķis	Izdzēst preci.
Funkcija	Izdzēst preci no tabulas "Items", kā arī no tabulas "HelperTable". Izmanto skatu "Edit Item".
Darbību loks	Funkcija pieejama no skata "Edit Item".

### 2.2.7. Pievienot pirkumu

Mērķis	Izveidot pirkumu.
Funkcija	Izveidot pirkumu tabulā "Purchases" jauna pirkuma izveides uzsākšanas laikā, vēlākai izmantošanai priekš kontaktu un preču pievienošanas, kā arī ievadīt pirkuma aprakstu. Izmanto skatu "New Purchase".
Darbību loks	Funkcija pieejama no sākuma skata.

### 2.2.8. Rediģēt pirkumu

Mērķis	Rediģēt eksistējošu pirkumu.
Funkcija	Atvērt pirkumu un ielādēt tā informāciju, tās rediģēšanai. Izmanto skatu "Edit Purchase".
Darbību loks	Funkcija pieejama no skata : "Edit Purchase"

### 2.2.9. Dzēst pirkumu

Mērķis	Dzēst eksistējošu pirkumu.
Funkcija	Izdzēst atvērto pirkumu pēc lietotāja izvēles no pirkumu Vēstures, izdzēst "atceltu" pirkumu (pirkumu, kurš netiek saglabāts pie jauna pirkuma izveides). Funkcija izmantojama skatos : "New Purchase" un "Purchase".
Darbību loks	Funkcija pieejama skatos : New Purchase, Purchase.

### 2.2.10. Pirkumu apskate

Mērķis	Atvērt izveidotu pirkumu.
Funkcija	Atvērt eksistējošu pirkumu caur pirkumu vēsturi izvēloties to pēc tā apraksta. Izmanto skatu "Purchase".
Darbību loks	Funkcija pieejama no skata "History".

### 2.2.11. Pirkumu eksportēšana

Mērķis	Izvadīt pirkuma informāciju.
Funkcija	Izvadīt pirkuma informāciju kā teksta ziņu vai CSV failu. Izmanto skatu "Preview".
Darbību loks	Funkcija pieejama no skata "Purchase"

### 2.2.12. Izmaksu kopsavilkumu apskate

Mērķis	Izveidot kopsavilkumu par pirkumiem vai precēm veicot speciālu atlasī.
Funkcija	Veikt atlasī izvēloties starp precēm vai kontaktiem, atlasot tos pēc noteiktiem kritērijiem. Preces atlasītas pēc to kategorijas vai kontakti no to saraksta.
Darbību loks	Funkcija pieejama skatā "Summary".

### 2.2.13. Paroles pārbaude

Mērķis	Datu drošība.
Funkcija	Atverot lietotni, lietotājam pieprasa ievadīt paroli, datu drošības nolūkos.
Darbību loks	Funkcija pieejama sākuma skatā.

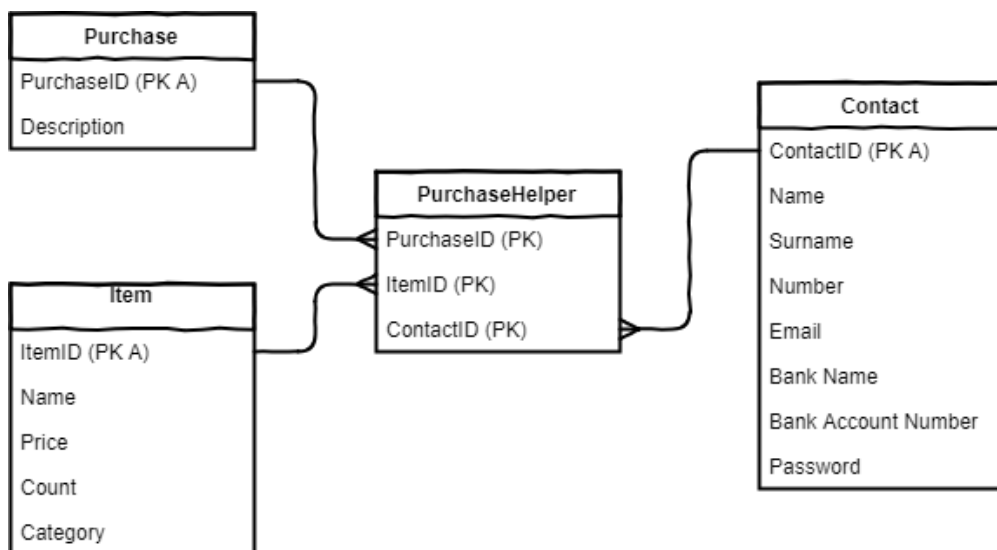
## 2.3. Datu atkarības

Izmoantotie saīsiājmi apakšnodaļās:

PK - Primary Key - Primārā atslēga;

A - Autoincrement - Automātiski aizpildīts, pieaugošs laukums.

### 2.3.1. Datu atkarības



Attēls 1. Realizācijas ER modelis.

## 2.3.2. Datu bāzes tabulu detalizēts apraksts

### 2.3.2.1. Tabula “Items”

Tabula satur informāciju par precēm.

Nosaukums	Tips	Attribūti	Apraksts
ItemID	Integer	PK, A	Preces identifikācijas numurs, preces atlasei
Name	Text		Preces nosaukums
Price	Float		Preces vienības izmaksas
Quantity	Integer		Preces vienību skaits
Category	Text		Preces kategorija

### 2.3.2.2. Tabula “Contact”

Tabulas satur informāciju par kontaktu informāciju

Nosaukums	Tips	Attribūti	Apraksts
ContactID	Integer	PK, A	Kontakta identifikācijas numurs, kontakta atlasei
Name	Text		Kontakta vārds
Surname	Text		Kontakta uzvārds
Number	Integer		Kontakta tālrunis
Email	Text		Kontakta epasta adrese
BankName	Text		Lietotāja izmantotās bankas nosaukums
BankAccountNumber	Text		Lietotāja bankas konta numurs
Password	Text		Lietotāja izvēlēta parole, datu aizsardzībai

### 2.3.2.3. Tabula “Purchase”

Tabulā uzglabājas informācija par pirkuma aprakstu.

Nosaukums	Tips	Attribūti	Apraksts
PurchaseID	Integer	PK, A	Pirkuma identifikācijas numurs, tā atlasei
Description	Text		Pirkuma apraksts jeb iemesls

### 2.3.2.4. Tabula “HelperTable”

Tabula satur apkopotu informāciju par to, kuras preces un kontakti atrodas pie attiecīgā pirkuma.

Nosaukums	Tips	Attribūti	Apraksts
PurchaseID	Integer	PK	Pirkuma identifikācijas numurs, tā atlasei
ItemID	Integer	PK	Preces identifikācijas numurs, tās atlasei
ContactID	Integer	PK	Kontakta identifikācijas numurs, tā atlasei

## 2.4. Lietotnes skatu relācijas

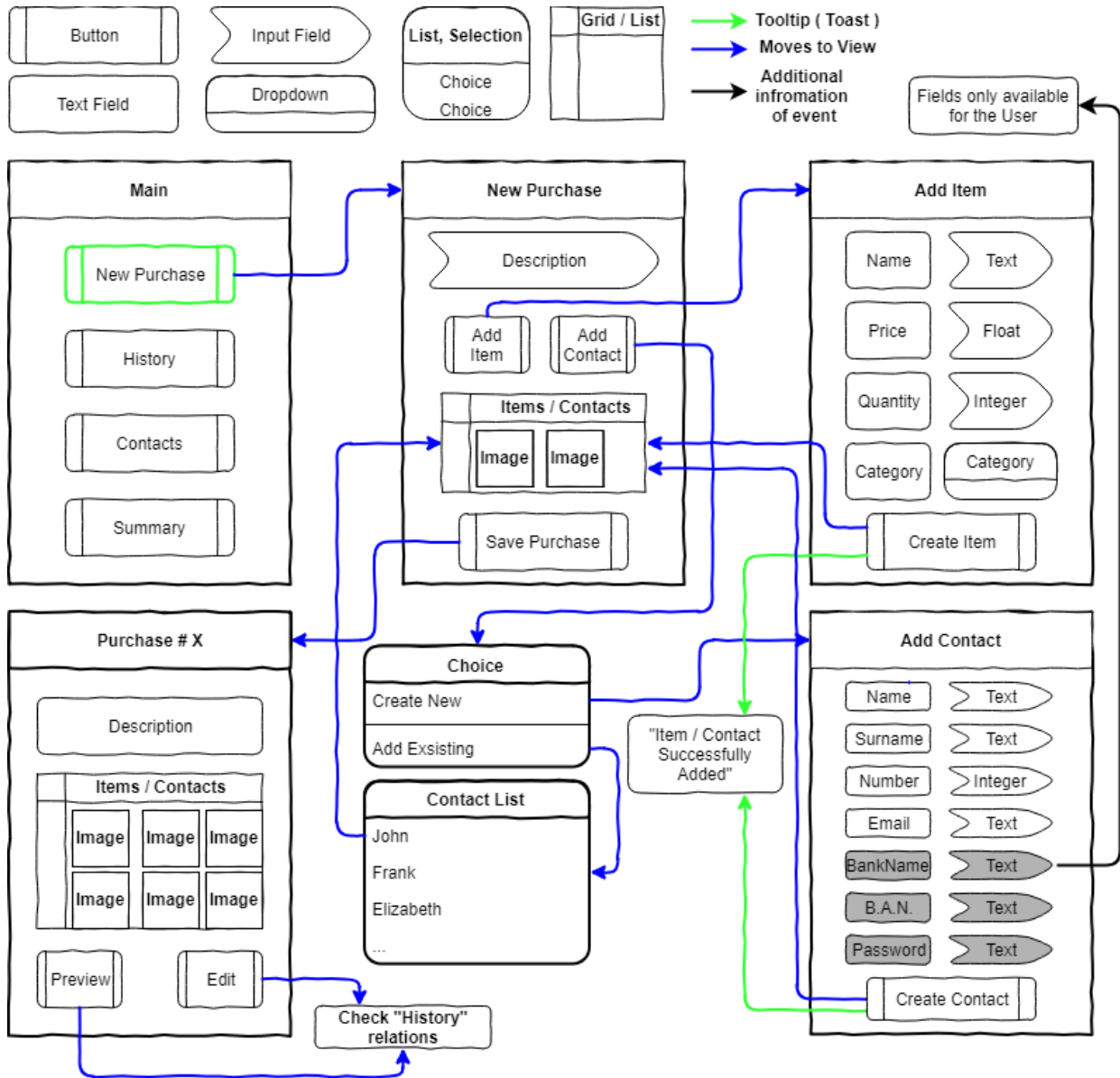
Lietotne pamatā sastāv no 4 darbībām:

- Jauns Pirkums (New Purchase);
- Vēsture (History);
- Kontakti (Contacts);
- Kopsavilkums (Summary);

### 2.4.1. Funkciju kopa “Jauns pirkums”

Kopa satur 6 skatus, potenciālais saskarnes izskats apskatāms (skat. Att. 2.):

- Jauns Pirkums (New Purchase);
- Pievienot Preci (Add Item);
- Rediģēt Preci (Edit Item);
- Pievienot Kontaktu (Add Contact);
- Rediģēt Kontaktu (Edit Contact);
- Pirkums (Purchase);



Att. 2. Jauna pirkuma modelis

### **2.4.1.1. Jauns pirkums**

Jauna pirkuma skats sastāv no ievades laukuma pirkuma aprakstam jeb iemeslam, trim pogām un diviem režģiem, kas uzglabā pirkuma preces un dalībniekus. Pogas attiecīgi aprakstītas nākamajos punktos.

### **2.4.1.2. Pievienot / Rediģēt preci**

Ar pogu “Add Item”, lietotājs tiek aizvadīts uz jaunas preces izveidi, kura satur četrus ievades laukus ar blakus attiecīgiem paskaidrojošiem laukiem kam tie parādzēti un pogu preces saglabāšanai. Saglabāšanas poga izveido preces ierakstu attiecīgā tabulā, pievieno to iepriekšējā skata pirkumam, izvada izgaistošu tekstu, par sekmīgu preces izveidošanu un atgriež lietotāju Jauna pirkuma skatā.

Preci iespējams rediģēt uzspiežot uz tās attiecīgās režģa rūts, kurā ir SVG attēls attiecīgi preces izvēlētajai kategorijai un preces nosaukums un tā kopīgā summa.

### **2.4.1.3. Pievienot / Rediģēt kontaktu**

Poga “Add Contact” savukārt ar uznirstoša loga palīdzību piedāvā lietotājam izveidot jaunu kontaktu, vai izvēlēties to no jau izveidoto kontaktu saraksta. Ja tiek izvēlēta eksistējošu kontaktu pievienošana, tad uznirstošais logs tiek aizstāts ar citu uznirstošu logu, kurš ir aizpildīts ar jau iepriekš izveidotajiem kontaktiem, bet ja lietotājs vēlas izveidot jaunu kontaktu. Tad līdzīgi kā preces izveidē, lietotājs tiek nogādāts uz attiecīgu skatu, kur tas var aizpildīt četrus ievades laukus (Vārds, Uzvārds, Numurs, Epasts), un saglabāt kontaktu, kas tad automātiski pievienojas pirkumam. Pie saglabāšanas arī parādas izgaistoša ziņa paziņojot par sekmīgu kontakta pievienošanu pirkumam.

Ja lietotājs vēlas rediģēt kontaktu, tad kontakts jāatver caur režģa rūti, kurā ir kontakta vārds un uzvārds. Pēc atvēršanas atrodamī tie paši lauki kuri tika izmantoti kontakta izveidei, kā arī poga kontakta dzēšanai, tā izdzēs kontaktu no pirkuma un paziņo ka kontakts noņemts no pirkuma veiksmīgi.

### **2.4.1.4. Pirkums**

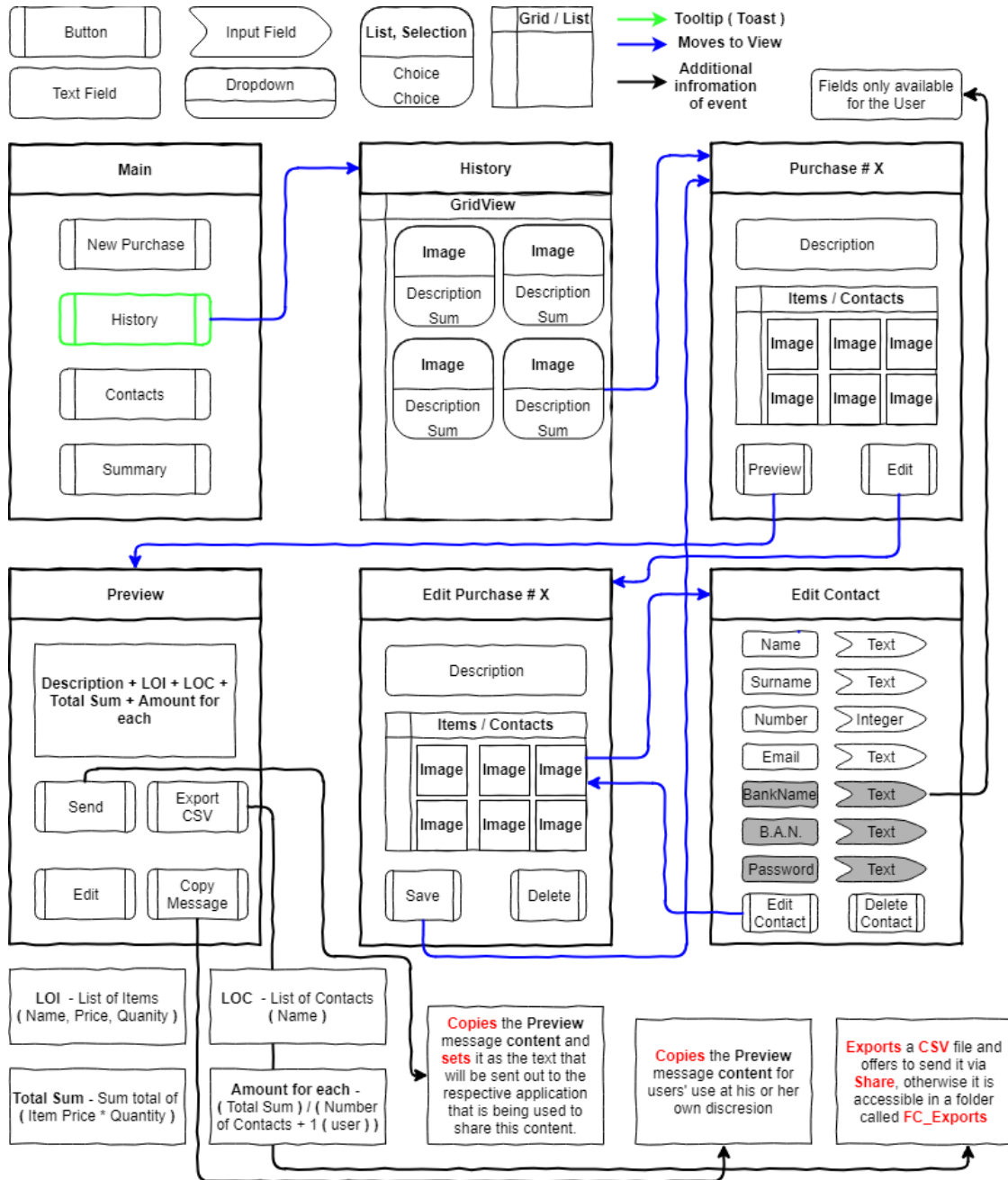
Pirkuma saglabāšanas poga to ne saglabā, bet atjaunina, jo pirkums tiek izveidots atveroties “New Purchase” skatam. Citādāk, neeksistē atslēga, ar kuru savienot pirkumu ar precēm un kontaktiem. Saglabāšanas poga atjaunina pirkuma aprakstu un aizvada lietotāju uz izveidotā pirkuma skatu “Purchase”, paziņojot ka pirkums veiksmīgi izveidots, pēc izveides, tas ir apskatāms Vēsturē. Ja saglabāšanas poga netiek nospiesta un lietotājs atgriežas iepriekšējā skatā, vai aizver lietotni, pirkums tiek atcelts un tā attiecīgā informācija tiek izdzēsta.

Pirkuma skats satur teksta laukumu tā aprakstam un divus režģus pirkuma preču un kontaktu atainošanai, kur preču attēlu atainošanai tiek izmantota funkcija no atvērtā pirmkoda bibliotēkas “Picasso”, kā arī divas pogas, pirkuma priekšskatījumam “Preview” un tā rediģēšanai “Edit”, kuru apraksti apskatāmi nākamajā nodaļā.

## 2.4.2. Funkciju kopa “Vēsture”

Attēls satur 4 skatus, potenciālais saskarnes izskats apskatāms (skat. Att. 3.):

- Vēsture (History);
- Pirkums (Purchase);
- Rediģēt Pirkumu (Edit Purchase);
- Priekšskatījums (Preview);



Attēls 3. Vēstures modelis

### **2.4.2.1. Vēsture**

Skats "History" satur režģi, kurš tiek piepildīts ar izveidotajiem pirkumiem. Pirkumi atverami, tiem uzspiežot, pirkumi tiek atdalīti režģa rūtīs pēc to kārtas numura un attiecīgi izveidotā apraksta. Gadījumos kad pirkuma apraksts ir garāks, režģa rūtī redzama tikai daļa no tā.

### **2.4.2.2. Pirkums**

Skat. nodaļā 2.4.1.4. Pirkums.

### **2.4.2.3. Rediģēt pirkumu**

Pirkuma rediģēšanas skats ļauj labot tā aprakstu, preces, kontaktus, kā arī pievienot jaunus kontaktus un preces. Pirkuma rediģēšanā arī ir iespējams pirkumu izdzēst, pēc kā lietotājs tiek atgriezts sākuma skatā.

### **2.4.2.4. Priekšskatījums**

Skats sadalīts 4 daļās, kur pirmā sadaļa parādzēta pirkuma aprakstam, otrā sadaļa preču režģim, trešā sadaļa kontaktu režģim un ceturta sadaļa funkciju pogām. Skats sadalīts gadījumam, ja kāda no pirmajām trim sadaļām ir pārāk apjomīga, tad to ir iespējams ritināt uz leju.

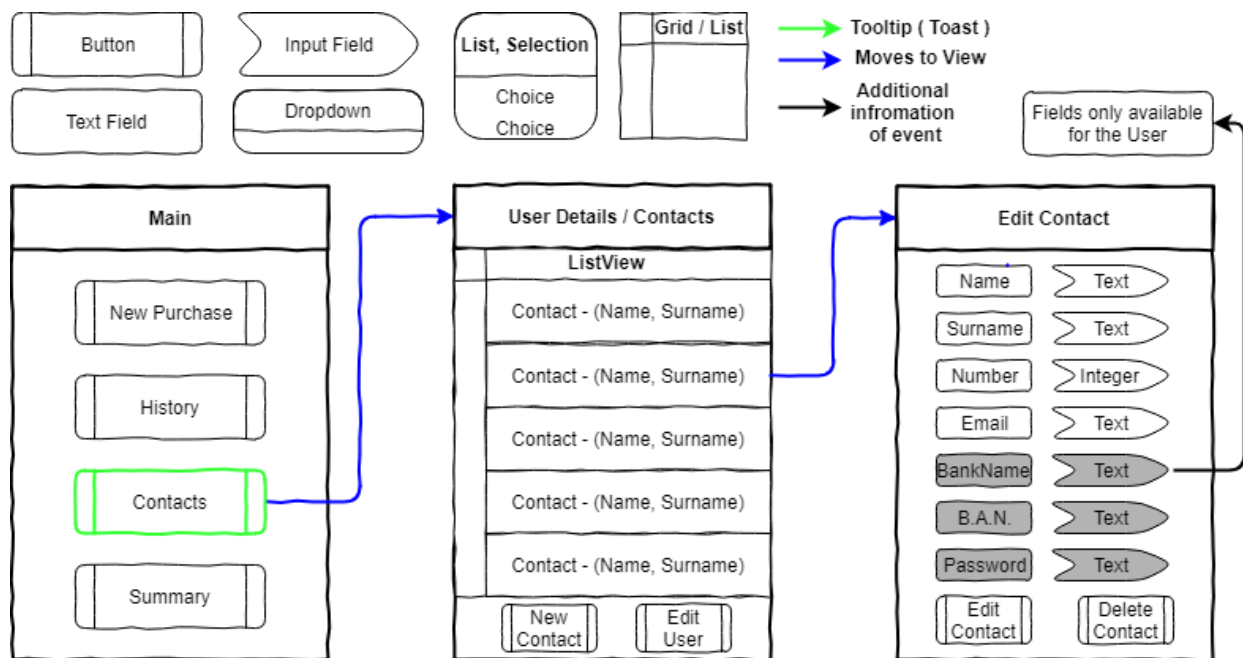
Priekšskatījuma skata ceturtajā sadaļā atrodas četras pogas, "Edit", "Copy Message", "Send / Share" un "Export CSV". Pirkuma rediģēšana ir apskatāma iepriekšējā punktā. Poga "Copy Message", tiek nokopēta ziņas informācija starpliktuvē - apraksts, preces, dalībnieki un kopīgā summa, kuru lietotājs var izmantot kā vēlas. Poga "Send" satur "Copy Message" funkciju, bet piedāvā lietotājam ar to dalīties izmantojot Android iebūvēto "Share" opciju.

"Export CSV", izveido jaunu CSV failu, kas pirmajā rindā satur preces kolonas, kā arī apraksta lauku. Nākamās rindas tiek aizpildītas ar preču informāciju un pēdējā rindā tiek pievienots pirkuma apraksts.

### 2.4.3. Funkciju kopa “Kontakti”

Kopa satur 3 skatus, potenciālais saskarnes izskats apskatāms (skat. Att. 4.):

- Kontakti (Contacts);
- Pievienot Kontaktu (Add Contact);
- Rediģēt Kontaktu (Edit Contact);



Attēls 4. Lietotāja / Kontaktu modelis

#### 2.4.3.1. Kontakti

Skats sastāv no trim elementiem - saraksts un divām pogām. Saraksts tiek aizpildīts ar lietotāja izveidotajiem kontaktiem, kuri tiek atainoti uzrādot to vārdu un uzvārdu. Pogas aizvada lietotāju uz attiecīgu skatu, kas ir aprakstīts nākamajos punktos.

#### 2.4.3.2. Pievienot kontaktu

Skat. nodaļā 2.4.1.3. Pievienot / Rediģēt kontaktu.

#### 2.4.3.3. Rediģēt kontaktu

Skat. nodaļā 2.4.1.3. Pievienot / Rediģēt kontaktu.

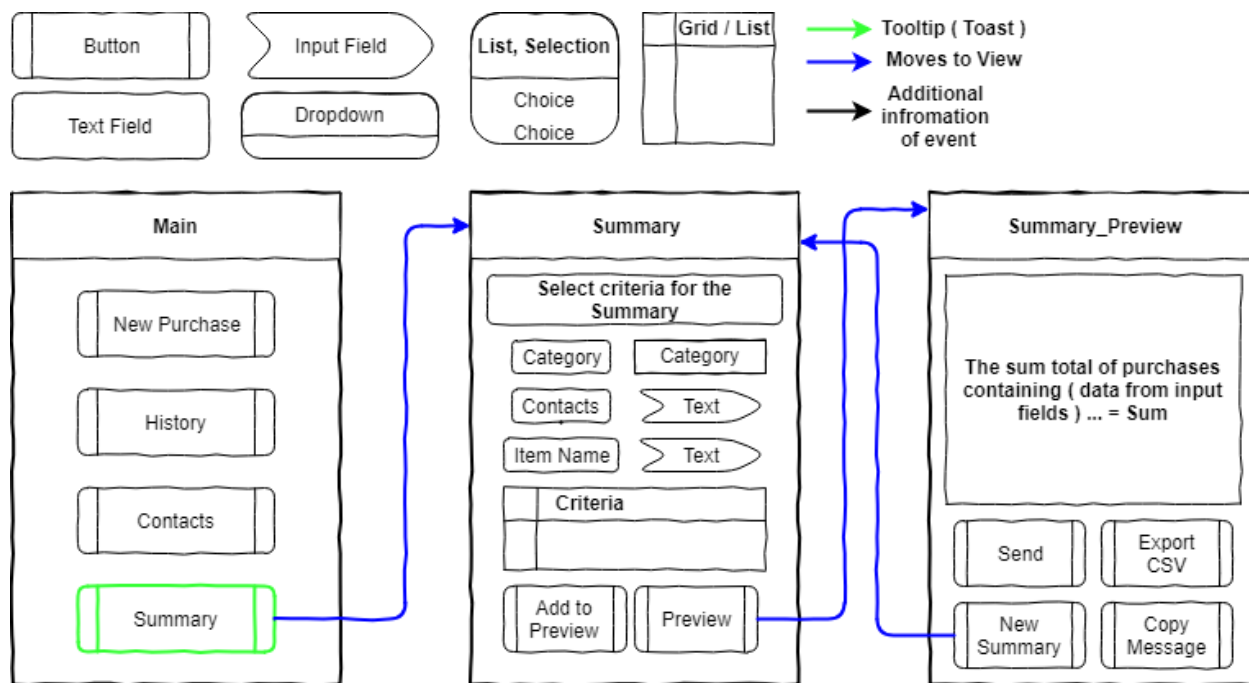
Mēģinot izdzēst kontaktu pilnībā, lietotājs tiek brīdināts ar uznirstošu logu, par to, ka kontakts tiks dzēsts arī no visiem pirkumiem kuros tas ir pievienots, logā atrodas papildus poga kontakta dzēšanai.

Poga “Edit User” ļauj mainīt lietotāja informāciju, kas sākotnēji ir tukša, bet satur papildus laukus, kuri parādzēti ziņas izveides papildus informācijai, ja pirkumā ir pievienoti dalībnieki.

## 2.4.4. Kopsavilkums

Kopa satur 2 skatus, potenciālais saskarnes izskats apskatāms (skat. Att. 5.):

- Kopsavilkums (Summary);
- Kopsavilkuma priekšskats (Summary Preview);



5. Attēls Kopsavilkuma modelis

### 2.4.4.1. Kopsavilkums

Kopsavilkumā lietotājs spēj veikt atlasī starp preču kategorijām, kontaktiem un preču nosaukuma.

Atlases kritēriji tiek pievienoti režģī ar pogas palīdzību, tie redzami režģī, tos iespējams noņemt uz tiem uzspiežot. Poga priekšskats lietotāju aizvada uz attiecīgu skatu. Ja kritēriju nav, tiek atlasīti visi pirkumi.

### 2.4.4.2. Kopsavilkuma priekšskats

Skatā tiek izvadīta kopsumma no pirkumiem, kuri satur visus no atlasītajiem kritērijiem. Kopsavilkumu iespējams kopēt, sūtīt vai eksportēt kā aprakstīts punktā “Priekšskatījums” (skat. 2.4.2.4. Priekšskatījums). Vai arī atgriezties uz iepriekšējo skatu, kurā izveidot jaunu kopsavilkumu.

### 3. TESTĒŠANA

Produkts tika izstrādāts ar spējās metodikas metodoloģiju, tādējādi pēc katras funkcijas ieviešanas, produkts tika intensīvi testēts turpmāku kļūdu novēršanai. Testēšanu veica produkta izstrādātājs, konsultējoties ar darba vadītāju.

#### 3.1. Saistības ar citiem dokumentiem

Testēšana veikta balstoties uz PPS funkcionālajām prasībām un to kritērijiem. Testēšanas laikā tika ņemts vērā attiecīgais sistēmas stāvoklis, kurā dažas no funkcijām nebija pieejamas.

#### 3.2. Funkciju darbības skaidrojumi un pieņemšanas kritēriji

Tiek aprakstīta funkcijas darbība, tās pieņemšanas kritēriji un sniegts vērtējums. Ir iespējami trīs dažādi vērtējumi (saīsinājumā Vērt.) :

- “✓” - Veiksmīgi izpildīts tests, kritēriju nosacījumi sasniegti;
- “✗” - Neveiksmīgi izpildīts tests, kritēriju nosacījumi netiek sasniegti;

##### 3.2.1. Preču testēšana

Funkcija	Darbība	Kritērijs	Vērt.
Pievienot preci	Lietotājs aizpilda laukumus, kuri tiek nolasīti tabulas ieraksta izveidei. Ja lauki tiek atstāti tukši, tad to vērtības tiek aizstātas ar to noklusējuma vērtībām (Name - Null, Price - 0, Quantity - 0, Category - Food).	Izvadās izgaistoša ziņa par izveidotu preci, lietotājs tiek atgriezts uz pirkumu, un prece tiek pievienota pirkuma režģī.	✓
Rediģēt preci	Atverot rediģēšanu preces lauki aizpildās automātiski ar atteicīgu informāciju, izņemot kategoriju, kura pēc noklusējuma ir “Food”.	Izvasdas ziņa par sekmīgu preces labošanu, lietotājs tiek atgriezts iepriekšējā skatā un preces informācija tiek atjaunota.	✓
Dzēst preci	Preces dzēšana ir pieejama preces rediģēšanas laikā, nospiežot uz pogu “Delete Item”.	Lietotājs tiek atgriezts iepriekšējā skatā un prece tiek izdzēsta no visām tabulām, kur tā bija sastopama, kas savukārt noņem to no preču režģa, kā arī parādās ziņa par tās dzēšanu.	✓

### 3.2.2. Kontakta testēšana

Funkcija	Darbība	Kritērijs	Vērt
Pievienot kontaktu	Atvērts kontakta pievienošanas skats. Lietotājs aizpilda laukumus, kuri tiek nolasīti tabulas ieraksta izveidei.	Izvadās ziņa par izveidotu kontaktu, lietotājs tiek atgriezts uz iepriekšējo skatu, un kontakts tiek pievienots režģī vai sarakstā.	✓
Rediģēt kontaktu	Kontakts tiek atvērts caur režģi vai sarakstu, lietotājs var labot tā laukumus.	Izvadīta ziņa par kontakta labošanu, un potenciāli atjauno informāciju režģī vai sarakstā.	✓
Dzēst kontaktu	Kontakta dzēšanai, tas jāatver caur režģi vai sarakstu.	Parādās ziņa par kontakta dzēšanu un tas tiek izdzēsts.	✓

### 3.2.3. Pirkumu testēšana

Funkcija	Darbība	Kritērijs	Vērt
Pievienot pirkumu	Lietotājs nospiež pogu "New Purchase"	Atveras jauna pirkuma izveides skats un izveidojas pirkums, kuru lietotājs var rediģēt.	✓
Rediģēt pirkumu	Lietotājs atver pirkumu caur to vēsturi un spēj to rediģēt pēc rediģēšanas pogas nospiešanas.	Atveras rediģēšanas skats, kurš aizpildās ar pirkuma aprakstu, tam pievienotajām precēm un kontaktiem, kuru lietotājs var rediģēt.	✓
Dzēst pirkumu	Lietotājs atver pirkuma rediģēšanas skatu un nospiež pogu "Delete".	Pirkums izdzēšas, lietotājs tiek aizvadīts uz pirkumu vēsturi.	✓
Apskatīt pirkumu	Lietotājs atver pirkumu caur to vēsturi un nospiež pogu "Preview".	Atveras pirkuma skats un tajā tiek izvadīta apkopota ziņa par pirkuma sastāvu	✓
Eksportēt pirkumu	Pirkuma priekšskatā lietotājs var izvēlēties eksportēt vai dalīties ar pirkuma sastāvu.	Skat. 2.2.11. Pirkumu eksportēšana	✓

### 3.2.4. Drošības / Kopsavilkuma testēšana

Funkcija	Darbība	Kritērijs	Vērt
Paroles pārbaude	Atverot lietotni, lietotājam tiek pieprasīts ievadīt paroli.	Parādās logs paroles ievadišanai, pēc kura lietotājs var izmantot lietotni.	✘
Kopsavilkums	Lietotājs veic kritēriju atlasī, pēc kuriem tiek veikta specifiska pirkumu vai preču atlase.	Atveras kopsavilkuma priekšskats, kurā tiek atlasīta informācija pēc attiecīgās atlases	✘

### 3.3. Kļūdu novēršana

Funkcija	Darbība kritēriju pieņemšanai
Paroles pārbaude	Izveidot uznirstošu logu paroles ievadišanai (ja tāda jau eksistē) un salīdzina to ar saglabāto vērtību, ja tās nesakrīt, tad pēc trešās nepareizās paroles ievades reizes, lietotne tiek aizvērta. Iespējams uzstādīt laika atskaiti pēc sistēmas laika, lai lietotājs nevar atkārtoti atvērt lietotni un atkal mēģināt 3 reizes ievadīt paroli. Izveidot funkciju paroles pārbaudei un enkriptēšanai.
Izmaksu kopsavilkuma apskate	Izveidot skatu kritēriju ievadei, funkciju ievades apstrādei, skatu izvadei.

## 4. PROJEKTA ORGANIZĀCIJA

Projekts tika izstrādāts pēc spējās “Agile” izstrādes metodoloģijas. Projekta izstrādē piedalījās divi cilvēki, tā autors un darba vadītājs kā projekta konsultants. Prasības izstrādāja viens cilvēks, konsultējoties ar darba vadītāju, projekta prasības neregulāri tika mainītas. Sākotnējās projekta prasības tika pieteiktas no pasūtītāja, kas sastāv no autora un viņa draugiem.

Projektu izstrādāja vairākos posmos, implementējot, pēc autora domām, produktam svarīgākās funkcijas vispirms. Tas tika izstrādāts un uzglabāts uz autora personīgajām ierīcēm. Projektam ieviesta GitHub versiju kontrole.

Projekta izstrādātā lietotne tā beigās tiks uzstādīta uz autora ierīces, kā arī tiks publicēta kādā brīvas piekļuves vietnē priekš citām personām, kas arī to vēlēšies izmantot. Programmatūras kods arī tiks publicēts, pēc tā kārtīgas izpētes un optimizēšanas pēc projekta laikā iegūtajām iemaņām.

### 4.1. Konfigurāciju pārvaldība

Autors ieviesa versiju kontroli ar “GitHub.com” palīdzību, jo tā ir atbalstīta programmatūra iekš Android Studio izstrādes platformas. Tādējādi nodrošinot projekta atgūšanu gadījumā, ja kas atgadās ar izstrādātāja ierīci vai atgriezt uz kādu iepriekšēju versiju projekta, kad projekts darbojās efektīvāk nekā šobrīd.

Sākotnēji versiju kontrole netika ieviesta, jo autors uzskatīja, ka tā nav nepieciešama, ņemot vērā to, ka projektu izstrādā tikai viena persona un nebija iedomājies par datu zaudēšanas varbūtību. GitHub vietne tika izvēlēta vienas no lasīto grāmatu dēļ. (DiMarzio, 2015)

## 5. KVALITĀTES NODROŠINĀŠANA

Projektā kvalitātes nodrošināšanai tika veiktas šādas rīcības:

- PPS izstrāde;
- PPA izstrāde;
- Testēšana un tās dokumentācija;
- Aprakstīta projekta organizācija;
- Veidots darbietilpības novērtējums;
- Programmēšana;

Programmatūras kods tika veidots vadoties pēc programmēšanas “labās prakses”, tā funkcijām, klasēm un mainīgajiem elementiem piešķirot paskaidrojošus nosaukumus, vai vismaz paskaidrojot kam attiecīgais mainīgais elements ir paradzēts. Kā arī tika komentētas funkcijas, to pielietojuma labākai izpratnei. Tādējādi veidojot programmatūras kodu “pašdokumentējošu”.

Pēc katras funkcionalitātes ieviešanas, lietotne tika testēta un tajā tika atrisinātas saskartās kļūdas.

## 6. DARBIETILPĪBAS NOVĒRTĒJUMS

### 6.1 Projekta darbietilpības plānošana

Darbietilpības plāns tika veidots pēc autora paša ieskatiem un prognozēm, bez iepriekšējas pieredzes un tikai ar daļēji noformētu aprakstu produktam.

Darbs	Pašnovērtējums	Aprēķinātais	Pavadītais laiks
PPS izveide	10	9	9
PPA izveide	10	11	13
Datubāzes izveide	4	6	7
Kontrolieru un Skatu izstrāde	6	7	6
Funkciju izstrāde	20	25	26
Testēšana un kļūdu labošana	5	6	6
Testēšanas dokumentācijas apkopošana	5	6	6
Atlikušās dokumentācijas izstrāde	10	11	12
<b>Kopā :</b>	<b>70</b>	<b>81</b>	<b>85</b>

Pēc vairāku soļu izpildes, autors atskatoties uz darba sākumā veikto novērtējumu saprata, ka jāveic atkārtota darbietilpības novērtēšana pēc formulas  $\frac{\text{optimālais} + \text{pesimistiskais} + 4(\text{realistiskais})}{6} = \text{darbietilpība}$

PPS izveidei tika pavadītas 9 dienas, lai tiktu noformulēta galēja ideja, ko ir nepieciešams sasniegt gala rezultātā. PPA izveidē, nācās pavadīt 13 dienas, kas tika veidota palaujoties uz autora veidoto PPS. Pašas lietotnes izstrādei tika atvēlētas 44 dienas, kuru laikā būtu jāizstrādā projekta funkcionālās prasības, kā arī jāveic to testēšana. Atlikušās dienas tika atvēlētas projekta un testēšanas dokumentācijas izstrādei un apkopošanai.

### 6.2 Secinājumi

Autors pārāk optimistiski sākotnēji novērtējis projekta izstrādes plūsmu, neierēķinot tajā aizkavēšanos kļūdu dēļ un to sekas. Autoram vajadzēja vēltīt vairāk laika gala produkta apraksta noformēšanai, lai tam nebūtu pārāk lielu iespēju atšķirties not tā sākotnēji plānotās funkcijas.

Programmatūras mērķis sākotnēji bija domāts maksājumu uzskaites un atmaksas atvieglošanai, bet pēc konsultēšanās ar darba vadītāju, tas tika mainīts.

## 7. NOBEIGUMS

Darba rezultāts ir funkcionējoša Android lietotne, pagaidām beta versijā, kuru iespējams izmantot kā lokālu grāmatvedības blociņu, tajā uzskaitīti tam pievienotie pirkumi un to izmaksas. Izstrādes laikā tika secināts ka programmēšanas “labā prakse” ir ļoti noderīga programmatūras koda izpratnei, pat pašam veidotājam, kad tas kādu laiku nav bijis apskatīts.

Pēc projekta izstrādes tika secināts ka jebkuram projektam nepieciešams konkrēts mērķis, no kura atkāpeniski jāveido apraksts ar katru soli precīzāk iedziļinoties tā prasībās un regulāri jāveic tā testēšana.

Autors izsaka pateicību “**The Noun Project**” dalībniekiem par to izstrādātajiem SVG attēliem, kuri tiek izmantoti lietotnē.

Nākotnē, lietotnē plānots ieviest:

- Pievienot trešo izvēli kontakta izveidošanā / pievienošanā caur lietotni “Kontakti”;
- Ieviest laika un datuma laukumus tabulās priekš:
  - Kopsavilkumu kritērijiem;
  - Ikmēneša kopsavilkuma paziņojuma;
  - Paroles pārbaude ik pēc noteikta laika posma;
- Kontaktu iedalījums 2 daļās :
  - Samaksājuši / Nesamaksājuši
- Pirkumu nolasišana no attēla (čeka);

## 8. IZMANTOTIE AVOTI

### Literatūra

Grant Allen & Mike Owens, “*The Definitive Guide to SQLite, Second Edition*”, 2010

J. F. DiMarzio, “*Android Studio Game Development: Concepts and Design*”, Nov. 2015

Sunny Kumar Aditya & Vikash Kumar Karn, “*Android SQLite Essentials*”, 2014

### Tīmekļa resursi

*Android Developers* - Dokumentācija [tiešsaiste],

Pieejams: <https://developer.android.com/>, apmeklējums - [11.04.2018]

*Android Open Source Project (AOSP)* - Dokumentācija [tiešsaiste],

Pieejams: <https://source.android.com/setup/start/build-numbers>, apmeklējums - [07.04.2018]

*Letonika - Angļu - latviešu vārdnīca* [tiešsaiste],

Pieejams: <https://www.letonika.lv/groups/default.aspx?g=2&r=10331062>,  
apmeklējums [26.04.2018]

*Stack Overflow* [tiešsaiste],

Pieejams: <https://stackoverflow.com/>, apmeklējums - [15.05.2018]

*TheNounProject - Icons for everything* [tiešsaiste],

Pieejams: <https://thenounproject.com/>, apmeklējums - [01.05.2018]

### Latvijas Valsts standarti

**Latvijas Valsts Standarts LVS 68:1996** ”Programmatūras prasību specifikācijas ceļvedis”.

**Latvijas Valsts Standarts LVS 72:1996** „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

## 10. PIELIKUMI

### 10.1. Programmatūras koda piemēri.

Pirkuma preču režģa aizpildīšana, daļa no faila : ItemAdapter.java

```
34      @Override
35      public View getView(int pos, View convertView, ViewGroup parent) {
36
37          int actual_ID = correctitem[pos];
38
39          // Define fields of the current grid frame
40          if (convertView == null) {
41              final LayoutInflater inflater = LayoutInflater.from(mContext);
42              convertView = inflater.inflate(R.layout.item_frame, root: null);
43
44              final TextView name = convertView.findViewById(R.id.item_name);
45              final TextView totalsum = convertView.findViewById(R.id.item_sum);
46              final ImageView cat = convertView.findViewById(R.id.cat_image);
47
48              final ViewHolder viewHolder = new ViewHolder(name, totalsum, cat);
49              convertView.setTag(viewHolder);
50          }
51
52          final ViewHolder viewHolder = (ViewHolder) convertView.getTag();
53
54          DBHelper db = new DBHelper(this.mContext, name: null, factory: null, version: 1);
55
56          viewHolder.nameTextView.setText(db.loadItemName(actual_ID));
57
58          DecimalFormat dec = new DecimalFormat("###.##");
59          int quantity = db.loadItemQuantity(actual_ID);
60
61          // Calculating the total sum of the items
62          float value = (Float.valueOf(dec.format(db.loadItemPrice(actual_ID)))) * quantity;
63          String total_item_sum = String.valueOf(value);
64
65          // Makes sure that there are 2 numbers after the decimal point for the output
66          String fixed_sum = smallCheck(total_item_sum);
67          viewHolder.sumTextView.setText(fixed_sum);
68
69          // Selects an SVG image for the grid frame
70          String Category = db.loadItemCat(actual_ID);
71          SelectCat(viewHolder, Category);
72
73          db.close();
74
75          return convertView;
76      }
77
```

## CSV faila izveides koda fragments no faila : Preview.java

```
public void Export() {
    DBHelper db = new DBHelper( context: this, name: null, factory: null, version: 1);

    String Columns = "\"Item\", \"Price\", \"Quantity\", \"Category\", \"Description\"";
    String Data = Columns + System.getProperty("line.separator");
    String fixedFloat = "";
    for (int i = 0; i < correctitem.length; i++) {
        if (correctitem[i] != 0) {
            fixedFloat = ItemAdapter.smallCheck(String.valueOf(db.loadItemPrice(correctitem[i])));
            Data += "\"" + db.loadItemName(correctitem[i])
                + "\", \"" + fixedFloat
                + "\", \"" + db.loadItemQuantity(correctitem[i])
                + "\", \"" + db.loadItemCat(correctitem[i])
                + "\", \"" // + db.getDesc(correctitem[i]) - don't need the description in every line
                + System.getProperty("line.separator");
        }
    }

    String EndRes = Data + "\"\", \"\", \"\", \"\", \"\" + db.getDesc(purchase_identification_number) + "\"";
    db.close();

    File file = null;
    File dir = new File( pathname: Environment.getExternalStorageDirectory().getAbsolutePath() + "/FP_Exports");
    if (!dir.exists()) {
        dir.mkdir();
    }

    String ExportName = "Purchase_" + purchase_identification_number + ".csv";
    file = new File(dir, ExportName);
    try {
        file.createNewFile();
    } catch (IOException e) {
    }
    FileOutputStream out = null;
    try {
        out = new FileOutputStream(file);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    try {
        out.write(EndRes.getBytes()); // writes the string into the .csv file
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        out.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    Uri u1 = Uri.fromFile(file);
    Sharing(u1);
}
```

## Dokumentārā lapa

Kvalifikācijas darbs „*Android lietotne - Draudzīgie maksājumi*” 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: *Egīls Paulovskis* \_\_\_\_\_ .05.2018.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *doc. Imants Gorbāns* \_\_\_\_\_ .05.2018.

Recenzents: *M. dat. Egons Šolmanis*

Darbs iesniegts 28.05.2018.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* \_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_\_.06.2018. prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_