

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ _____
подпись инициалы, фамилия
« ____ » _____ 20 __ Г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование направления

Программная система организации и учёта покупок

тема

Пояснительная записка

Руководитель

подпись, дата

должность, ученая степень

Д. А. Швец
инициалы, фамилия

Выпускник

подпись, дата

Д. А. Частоступов
инициалы, фамилия

Нормоконтролер

подпись, дата

должность, ученая степень

Д. А. Швец
инициалы, фамилия

Красноярск 2020

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ _____
подпись инициалы, фамилия
« ____ » _____ 20 ____ Г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Частоступов Дмитрий Александрович
фамилия, имя, отчество

Группа КИ16-07Б Направление (специальность) 09.03.01
номер код

Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы Программная система организации и учёта покупок

Утверждена приказом по университету № _____ от _____

Руководитель ВКР Д. А. Швец, канд. техн. наук., доцент, кафедра ИВТ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: провести анализ известных мобильных приложений по хранению дисконтных карт и учёту покупок, разработать мобильное приложение для организации и учёта покупок с возможностью хранения дисконтных карт.

Перечень разделов ВКР: анализ предметной области и анализ существующих решений, проектирование приложения, реализация приложения, инструкция пользователя.

Перечень графического материала: презентация в формате PowerPoint.

Руководитель ВКР

подпись

Д. А. Швец
инициалы и фамилия

Задание принял к исполнению

подпись

Д. А. Частоступов
инициалы и фамилия студента

« ____ » _____ 2020 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программная система организации и учёта покупок» содержит 50 страниц текстового документа, 1 таблицу, 42 иллюстрации, 11 использованных источников.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, СПИСКИ ПОКУПОК, ШТРИХ-КОДЫ, QR КОДЫ, ПОКУПКИ, БАЗА ДАННЫХ, ОС ANDROID.

Цель выпускной квалификационной работы – разработка мобильного приложения для организации и учёта покупок с учётом имеющихся дисконтных карт.

В результате выполнения выпускной квалификационной работы была определена структура для мобильного приложения по организации, учёту покупок и хранению дисконтных карт, установлены функциональные требования, выбраны средства реализации, произведена разработка мобильного приложения, которое позволяет добавлять дисконтные карты в базу данных, составлять списки покупок, сканировать QR коды с чеков после покупок добавляя их в базу данных, для последующего учёта расходов за определённые промежутки времени.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ.....	4
1 Анализ предметной области.....	6
1.1 Постановка задачи	6
1.2 Цель создания системы	6
1.3 Функциональные возможности	7
1.4 Анализ существующих решений.....	7
1.4.1 Stocard	7
1.4.2 PINbonus.....	8
1.4.3 Кошелёк	9
1.5 Вывод по главе	10
2 Проектирование приложения.....	11
2.1 Определение функциональных требований.....	11
2.1.1 Работа с картами	12
2.1.2 Работа с заметками	13
2.1.3 Работа с расходами	14
2.1.4 Настройки приложения	15
2.1.5 Диаграмма прецедентов	16
2.2 Структура приложения.....	16
2.2.1 Определение структуры приложения.....	16
2.2.2 Структура представлений	18
2.2.3 Структура моделей представления	19
2.2.4 Структура модели	20

2.3	Выбор средства реализации	21
2.3.1	Выбор модели представления данных.....	21
2.3.2	Выбор интегрированной среды разработки	23
2.4	Вывод по главе	24
3	Реализация приложения.....	25
3.1	Разработка модели	26
3.2	Разработка части приложения по работе с картами.....	26
3.3	Разработка части приложения по работе со списками покупок .	31
3.4	Разработка части приложения по работе с расходами.....	35
3.5	Реализация настроек приложения	38
3.6	Вывод по главе	39
4	Инструкция пользователя.....	40
4.1	Взаимодействие с картами.....	40
4.2	Взаимодействие со списками покупок	43
4.3	Взаимодействие с расходами.....	44
4.4	Настройки приложения	47
	ЗАКЛЮЧЕНИЕ	48
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49

ВВЕДЕНИЕ

В современном обществе потребления человек приобретает всё больше и больше товаров в совершенно разных магазинах, которые зачастую имеют свои программы лояльности и в следствии дисконтные карты. Из-за этого у людей скапливается всё больше и больше дисконтных карт, иногда их число может переваливать за сотню у конкретного человека, что может, как минимум, доставлять дискомфорт из-за постоянного ношения с собой огромного количества дисконтных карт, а как максимум, может привести к потере денег и времени из-за забытой, но нужной дисконтной карты во время очередной покупки в магазине. Так же из-за увеличения количества приобретаемых товаров менеджмент личных расходов становится всё более важной частью современной жизни человека, обычно для этого люди стараются составлять списки необходимых для приобретения товаров и ведут учёт потраченных, после каждой покупки, средств. Если же начать относиться к этой части современной жизни не подобающе, то чаще всего, наступают негативные последствия – потеря личных средств на ненужные вещи, взамен первостепенных, нарастающий стресс и вследствие, ухудшение здоровья.

Благодаря активному развитию мобильных операционных систем и рынка смартфонов, сейчас каждый человек имеет в кармане универсальное устройство способное на установку огромного количества сторонних приложений, фотосъёмку, хранение данных, передачу данных через различные сети и множество других операций. Всё это привело к большому количеству мобильных приложений главной задачей которых является как хранение дисконтных карт, так и менеджмент расходов.

К менеджменту расходов можно отнести составление списка покупок, контроль расходов с помощью сохранения чеков от покупок и подсчёт потраченных средств за определённый период с помощью сохранённых чеков. Но с хранением чеков есть проблема, они не долговечны – даже если они не подвергались внешним воздействиям, со временем краска на чеках начинает

выцветать и чеки теряют читаемость. Такие чеки теряют свою актуальность и невозможно подсчитать свои расходы за определённый период.

Так как многие доступные приложения могут либо подсчитывать расходы, либо только хранить сканировать и хранить дисконтные карты, либо только составлять списки покупок. Всё это подтолкнуло меня на создание мобильного приложения, которое бы совмещало в себе все вышеперечисленные функции.

Целью данной работы является создание мобильного приложения по организации и учёту покупок с учётом имеющихся дисконтных карт.

Для достижения этой цели необходимо выполнить следующие задачи:

- Провести анализ схожих по функционалу существующих мобильных приложений;
- Определить функциональные требования к приложению;
- Определить структуру приложения;
- Выбрать средства реализации приложения;
- Разработать мобильное приложение.

1 Анализ предметной области

1.1 Постановка задачи

Целью текущей бакалаврской работы является разработка мобильного приложения для организации и учёта покупок с учётом имеющихся дисконтных карт. Разработанное мобильное приложение должно уметь хранить список запланированных покупок и дисконтных карт, а также вести учёт расхода средств.

Система должна удовлетворять следующим требованиям:

- Интерфейс программы должен быть легко воспринимаемым;
- Программа должна быть отказоустойчивая;
- Программа не должна потреблять слишком много ресурсов смартфона;
- Данные о дисконтных картах, составленных списках покупок и расходах должны храниться в базе данных.

Система должна выполнять следующие задачи:

- Сканировать штрих-коды дисконтных карт;
- Сканировать QR коды с чеков после покупок;
- Хранить дисконтные карты в базе данных;
- Хранить составленные списки запланированных покупок в базе данных;
- Уметь экспортировать и импортировать базу данных;
- Учитывать потраченные средства за промежутки времени.

1.2 Цель создания системы

Целью является создание мобильного приложения по организации и учёта покупок и расходов с возможностью хранить дисконтные карты.

1.3 Функциональные возможности

- Возможность составления списков покупок по категориям;
- Возможность сканирования QR кодов для считывания информации о покупке для учёта потраченных средств;
- Возможность сканирования дисконтных карт для хранения их в базе данных;
- Возможность экспортировать базу данных для восстановления отсканированных дисконтных карт, составленных списков покупок и сохранённых расходах на другом мобильном устройстве.

1.4 Анализ существующих решений

1.4.1 Stocard

Интерфейс:

Интерфейс программы Stocard [1] выполнен в светлом стиле, на главном экране в центральной области находятся, ранее добавленные, дисконтные карты, в нижней части интерфейса кнопка по добавлению новой карты и три вкладки по ней — это список карт, рекламные предложения и настройки.



Рисунок 1 – Интерфейс приложения Stocard

Дополнительные возможности:

Приложение так же может синхронизировать сохранённые карты с сервером, для этого надо создать учётную запись с помощью электронной почты или привязать аккаунт Google или Facebook.

Во вкладке “Offers” находятся рекламные предложения от магазинов, где можно узнать о актуальных акциях.

Кроссплатформенность:

Приложение Stocard доступно как на Android, так и на IOS.

1.4.2 PINbonus

Интерфейс:

Интерфейс программы PINbonus [2] также выполнен в светлом стиле, на главном экране в центральной области находятся, ранее добавленные, дисконтные карты, в нижней части интерфейса кнопка по добавлению новой карты, настройки по выбору языка и авторизация находятся в меню типа “гамбургер”.

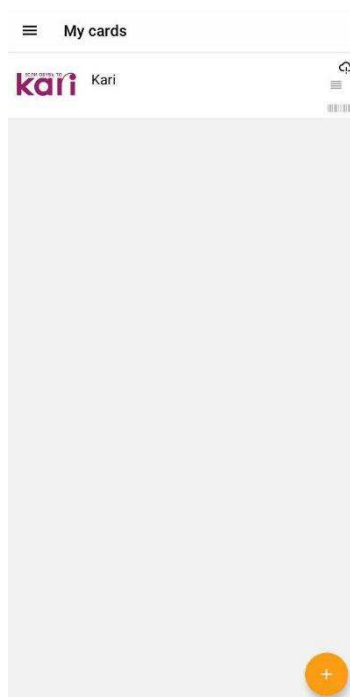


Рисунок 2 – Интерфейс приложения Stocard

Дополнительные возможности:

Приложение так же может синхронизировать сохранённые карты с сервером, для этого надо создать учётную запись с помощью привязки аккаунта Google или VK.

Кроссплатформенность:

Приложение PINbonus доступно как на Android, так и на IOS.

1.4.3 Кошелёк

Интерфейс:

Интерфейс программы Кошелёк [3] также выполнен в тёмном стиле, на главном экране вверху находятся три кнопки, по нажатию первой открывается окно, где можно выпустить, как дисконтные, так и банковские карты, с нажатием второй можно отсканировать свою карту, по нажатию третьей кнопки открывается окно с актуальными акциями. В центральной области находятся, ранее добавленные, дисконтные и банковские карты. В меню типа “гамбургер” можно войти в учётную запись, в настройки и в меню по управлению картами.

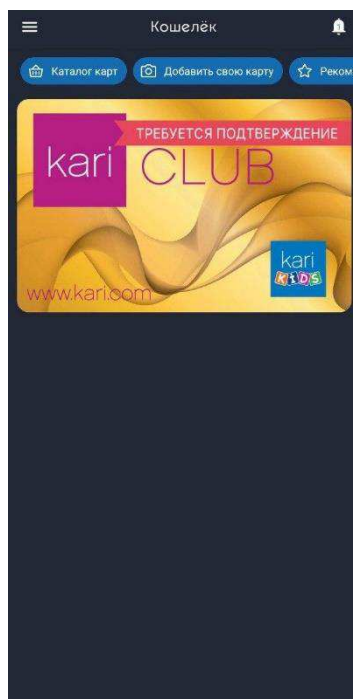


Рисунок 3 – Интерфейс приложения Кошелёк

Дополнительные возможности:

Приложение так же может синхронизировать сохранённые карты с сервером, для этого надо создать учётную запись с помощью номера телефона и почты.

Кроссплатформенность:

Приложение Кошелёк доступно как на Android, так и на IOS.

1.5 Вывод по главе

В данной главе были поставлены цели и задачи, которые должны быть выполнены при реализации данной выпускной квалификационной работы. Обзор аналогов позволил получить представление какой должна быть система и какой функционал она должна реализовывать.

2 Проектирование приложения

2.1 Определение функциональных требований

При рассмотрении аналогичных приложений и систем, а также при более детально рассмотрении технического задания были сформированы следующие требования к функционалу:

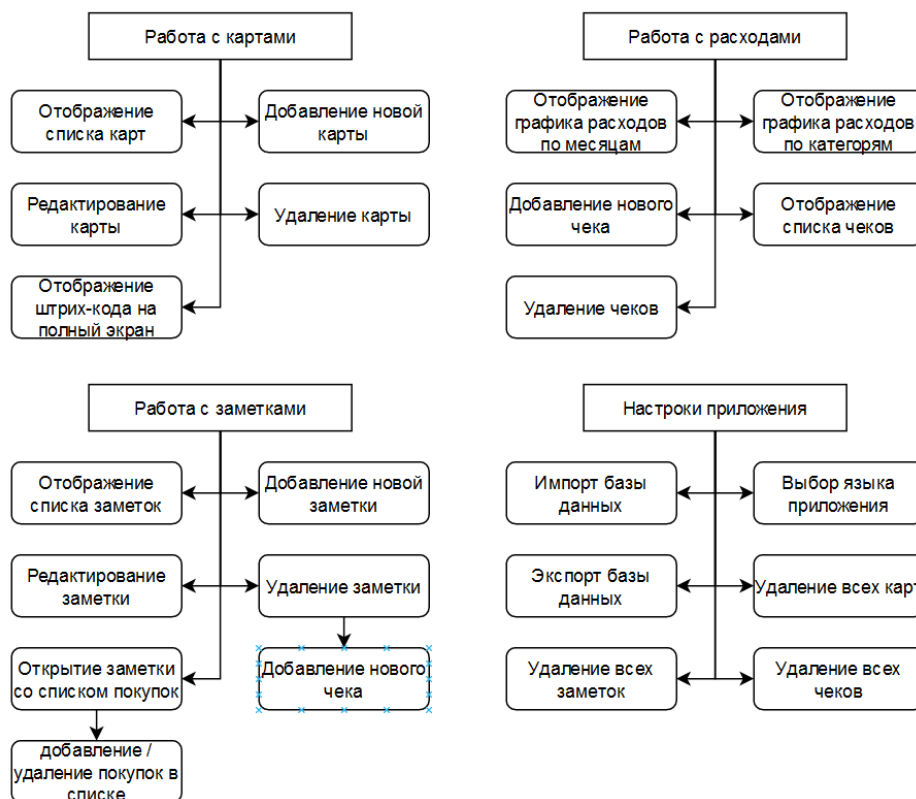


Рисунок 4 – Наглядное отображение функционала

Так функционал разделится на следующие модули:

- a) Работа с картами
 - 1) Отображение списка карт
 - 2) Добавление новой карты
 - 3) Редактирование карты
 - 4) Удаление карты
 - 5) Отображение штрих-кода карты на полный экран
- b) Работа с заметками
 - 1) Отображение списка заметок

- 2) Добавление новой заметки
 - 3) Редактирование заметки
 - 4) Удаление заметки
 - 5) Удаление заметки с добавлением нового чека
 - 6) Открытие заметки со списком покупок
 - 7) Добавление покупок в заметке
 - 8) Удаление покупок в заметке
- с) Работа с расходами
- 1) Отображение гистограммы расходов по месяцам
 - 2) Отображение графика расходов по категориям
 - 3) Добавление нового чека
 - 4) Отображение списка чеков
 - 5) Удаление чеков
- d) Настройки приложения
- 1) Выбор языка приложения
 - 2) Импорт базы данных
 - 3) Экспорт базы данных
 - 4) Удаление всех карт
 - 5) Удаление всех заметок
 - 6) Удаление всех чеков

2.1.1 Работа с картами

Приложение должно осуществлять работу с дисконтными картами – добавлять новые карты, удалять и изменять существующие. Обработка включает в себя следующие функции:

– Отображение списка карт. Существующие в базе данных дисконтные карты должны отображаться на экране в виде списка, при взаимодействии с которым можно манипулировать существующими картами. Каждая карта в этом списке должна содержать следующие поля:

- а) Имя карты;

- b) Фотография карты;
- c) Номер из штрих-кода карты;
- d) Описание карты.

– Добавление новой дисконтной карты. При добавлении карты сканирование штрих-кода карты должно осуществляться с помощью камеры смартфона, при этом нужно учесть, что штрихкоды имеют форматы типа «EAN_13» и «CODE_128». Так же каждая карта должна иметь фотографию и имя, описание карты опционально. Хранение фотографий карт так же будет в базе данных, однако можно хранить фотографии в файловой системе смартфона, а в базе данных хранить только пути к этим фотографиям, но это не позволит легко эскортировать базу данных.

– Удаление существующей дисконтной карты. При удалении карты пользователю должна быть предоставлена возможность в течении короткого промежутка времени восстановить удалённую карту, чтобы исключить случайное удаление.

– Редактирование существующей дисконтной карты. Возможность редактировать должна быть все поля карты.

– Отображение на полный экран штрих-кода дисконтной карты. При этом ещё должен быть отображен номер, который закодирован в самом штрих коде, сам же штрих код должен быть легко читаем для сканирующих устройств.

2.1.2 Работа с заметками

Второй важной функцией приложения является составление списка покупок. Так же, как и с картами должна быть возможность создавать, удалять, редактировать и отображать эти списки – заметки.

– Добавление новой заметки. При этом каждая заметка обязательно должна содержать поле с её именем, опционально описанием и категорией заметки. Каждая новая заметка должна иметь уникальное имя, создание заметок с одинаковыми именами нужно запретить пользователю.

– Отображаться заметки должны в виде списка.

- Редактировать в заметке должно быть доступно как имя, так и описание, и категорию заметки.
- При открытии заметки должен быть отображен список запланированных покупок, должны быть возможность добавлять и удалять элементы в этом списке, при этом добавление дубликатов покупок должно быть запрещено.
- При удалении заметки у пользователя должна быть возможность отменить удаление, а также сразу после удаления создать новый чек, в котором уже будут заполнены дата покупки и категория покупки.

2.1.3 Работа с расходами

Третьей важной функцией приложения является подсчёт расходов, он будет производиться на основе сохранённых чеков, а результаты будут выводиться в виде графиков, где будут отображаться расходы по каждому месяцу.

- Расходы по месяцам будут отображены в виде гистограммы, где каждый месяц будет отдельным столбцом.
- Расходы в конкретном месяце будут отображены в круговой гистограмме, где будут показываться категории расходов.
- Добавление чеков будет осуществляться в базу данных. На основе сохранённых чеков будет осуществлён расчёт расходов по месяцам. Данные для чеков можно будет считывать с реальных – физических чеков, сканируя QR коды, которые можно найти сейчас на каждом чеке. Из этого QR кода можно получить информацию о дате покупке и цене. Каждый чек должен иметь поля с:
 - а) Датой покупки;
 - б) Ценой;
 - в) Категорией покупки.
- Сохранённые чеки должны отображаться списком в отдельном меню, где возможно их отредактировать и удалить.

2.1.4 Настройки приложения

Приложение должно иметь меню с настройками.

- Настройка языка приложения. В приложение должно быть адаптировано для русскоязычных и англоязычных пользователей. Такой выбор языков обусловлен тем, что английский язык является международным, а моим родным языком является русским и приложение рассчитано в основном на русскоязычных пользователей.
- Экспорт базы данных должен выполняться на внутреннее хранилище смартфона
- Импорт базы данных должен выполняться из внутреннего хранилища смартфона.
- Удаление дисконтных карт. При этом действию должны удаляться все сохранённые карты из базы данных.
- Удаление всех заметок. При этом действию должны удаляться все сохранённые заметки из базы данных.
- Удаление всех чеков. При этом действию должны удаляться все сохранённые чеки из базы данных.

2.1.5 Диаграмма прецедентов

Учитывая весь вышеописанный функционал можно построить следующую диаграмму прецедентов, которая будет отображать какие действия сможет выполнить пользователь, использующий приложение.

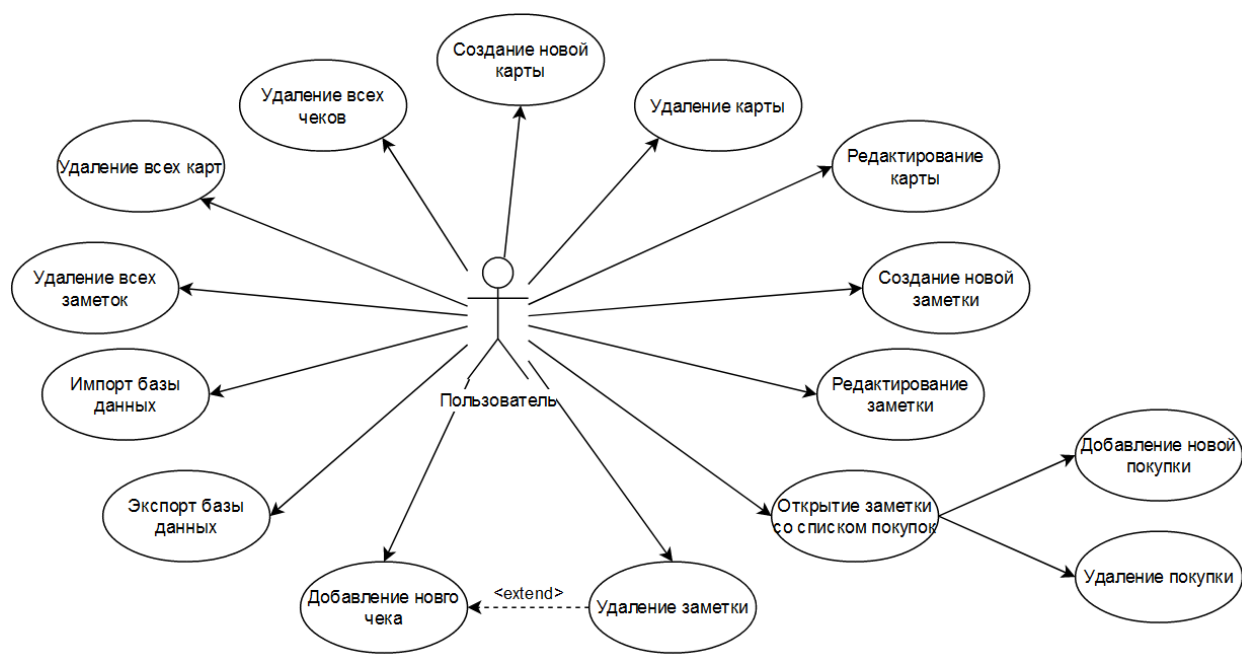


Рисунок 5 – Диаграмма прецедентов

2.2 Структура приложения

2.2.1 Определение структуры приложения

При начале проектирования приложения первичной задачей является разбиение приложения на функциональные модули, необходимые для реализации. Для решения этой задачи применяются шаблоны проектирования. Наиболее распространенными шаблонами являются:

- Model-View-Controller;
- Model-View-Presenter;
- Model-View-ViewModel.

Был выбран MVVM (Model-View-ViewModel) шаблон проектирования так, как представления и данные не должны быть напрямую связаны, а прослойка

между данными и представлениями не должна напрямую воздействовать на представления, а должна лишь предоставлять данные.

Так же следует добавить репозиторий, который будет являться абстракцией над базой данных. Репозиторий поможет сделать код каждого ViewModel более чистым и менее громоздким, а также независимым от реализации базы данных.

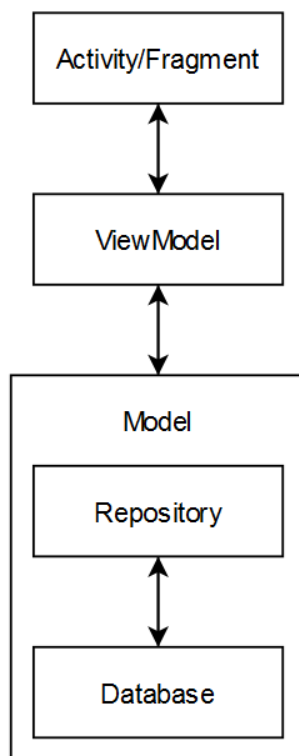


Рисунок 6 – Структурная схема приложения

Следовательно структурная схема приложения включает в себя следующие элементы:

- Активности/ Фрагменты (Views);
- ViewModels;
- Model;
 - a) Репозиторий;
 - b) База данных.

2.2.2 Структура представлений

Представления являются той частью приложения с которой пользователь взаимодействует напрямую.

Из требуемого функционала можно сделать вывод, что необходимы следующие представления:

- `SplashScreenActivity` – активность, показывающая загрузочный экран пользователю перед стартом приложения.

- `MainActivity` – главная активность, которая будет содержать «`FrameLayout`» для помещения в него одного из трёх фрагментов:

- a) `CardsFragment`– фрагмент, в котором будет располагаться список сохранённых дисконтных карт и кнопка для добавления новой карты.

- b) `NotesFragment`– фрагмент, в котором будет располагаться список сохранённых списков покупок и кнопка для добавления нового списка покупок.

- c) `CostsFragment` – фрагмент, в котором будут располагаться кнопка добавления нового чека и две гистограммы, первая будет отображать количество потраченных средств в каждом месяце текущего года, вторая будет отображать расходы, разбитые по категориям в выбранном месяце.

- `CostsList` – активность отображающая список сохраненных чеков.

- `SettingsActivity` – активность с настройками приложения, в которой можно будет сменить язык, импортировать или экспортировать базу данных, а также отчистить её.

- `AddEdit_Card` – активность для добавления новой дисконтной карты в базу данных приложения.

- `BigBarcodeDialog` – диалог показывающий на весь экран штрих-код, при нажатии в `cards_fragment` на любую дисконтную карту.

- AddEdit_PurchaseList – активность открывающаяся при нажатии на любую заметку в notes_fragment и показывающая список покупок, который содержится в этой заметке.
- AddEdit_Note – активность для добавления новой заметки в базу данных приложения.
- AddEdit_QRCheck – активность для добавления нового чека в базу данных приложения.
- Scanner – активность для сканирования штрих-кодов и QR кодов с чеков и дисконтных карт, после сканирования передаёт расшифрованный код из QR кода или штрих-кода в родительскую активность.

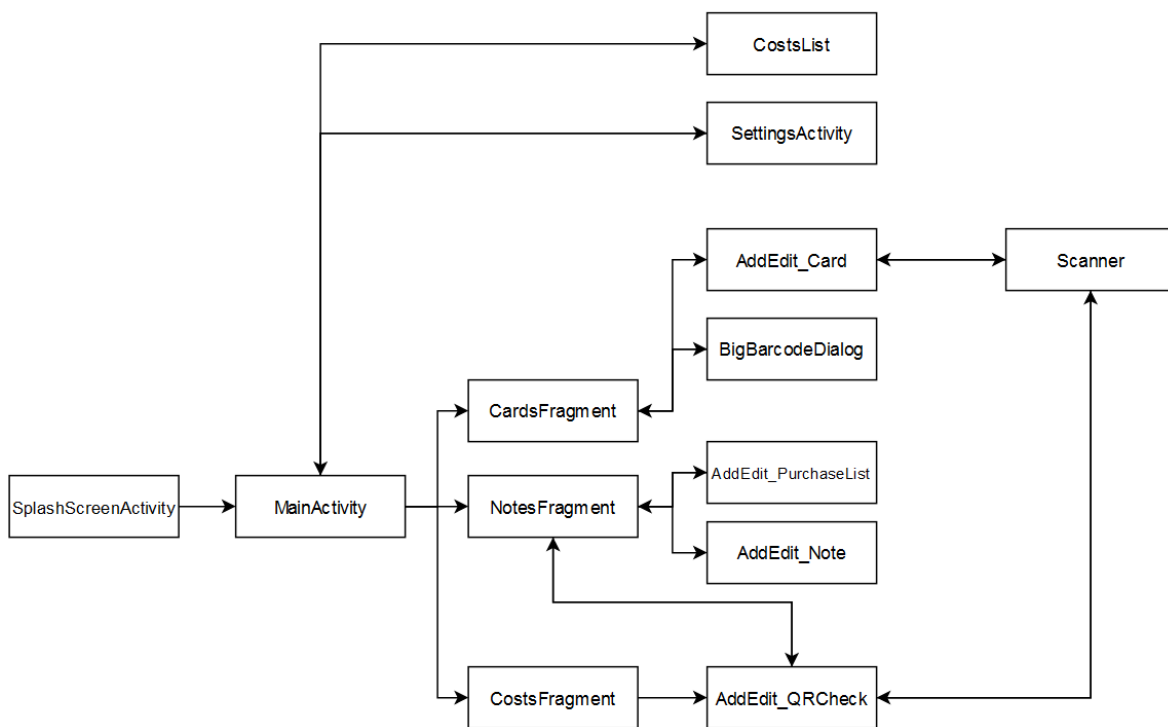


Рисунок 7 – Иерархия представлений

2.2.3 Структура моделей представления

Модели представления являются программной обёрткой для предоставления данных из модели в представление.

Учитывая, что у нас будет четыре таблицы с различными данными в базе данных, то соответственно следует создать четыре модели представления:

– `CardViewModel` – взаимодействует с данными из таблицы «Cards» и предоставляет их для представления `CardsFragment` с использованием хранилища данных `LiveData`.

– `CostsViewModel` – взаимодействует с данными из таблицы «Costs» и предоставляет их для представления `CostsList` и `CostsFragment` с использованием хранилища данных `LiveData`.

– `NotesViewModel` – взаимодействует с данными из таблицы «Notes» и предоставляет их для представления `NotesFragment` с использованием хранилища данных `LiveData`.

– `PurchaseViewModel` – взаимодействует с данными из таблицы «Purchase» и предоставляет их для представления `AddEdit_PurchaseList` с использованием хранилища данных `LiveData`.

2.2.4 Структура модели

В шаблоне проектирования MVVM (Model-View-ViewModel) модель представляет собой данные и методы работы с ними, которые не зависят от представлений (View) и моделей представления (ViewModel). Модель будет состоять из репозитория и базы данных, которая будет реализована с помощью библиотеки Room [9].

Room – встроенный в Android высокоуровневый интерфейс для низкоуровневых привязок SQLite, который создаёт свой API интерфейс поверх встроенного SQLite API во время компиляции программы, является частью Android Architecture Components. Состоит из трёх основных компонентов: DAO, Entity и Database (Рисунок 8).

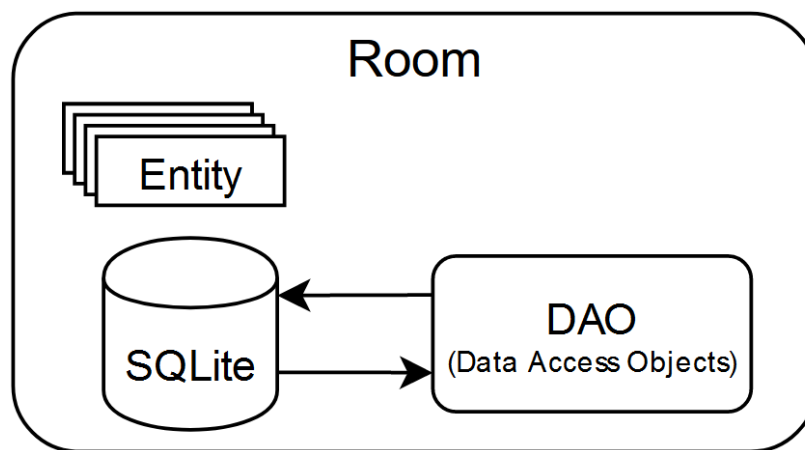


Рисунок 8 – Схема Room

Database – основной класс по работе с базой данных, помечается аннотацией «@Database» и наследуется от класса «RoomDatabase»

Entity – классы описывающие таблицы базы данных и помеченные аннотацией «@Entity». Учитывая, что нужно создать четыре таблицы, то соответственно будет создано четыре Entity класса:

- Card;
- Note;
- Purchase;
- Costs.

DAO – классы помеченные аннотацией «@Dao» описывают методы работы с базой данных. Понадобится создать так же четыре DAO класса:

- CardDao;
- NoteDao;
- PurchaseDao;
- CostsDao.

2.3 Выбор средства реализации

2.3.1 Выбор модели представления данных

При выборе модели представления данных подразумевается, как данные будут храниться в базе данных.

Для разработки Android приложения нам необходима компактная и легко встраиваемая база данных, которая позволит разбить хранимые данные на таблицы.

Под эти требования подходит SQLite [10] – свободная, реляционная система управления базами данных, официально поддерживаемая Android.

База данных приложения будет состоять из следующих таблиц и полей (Рисунок 9):

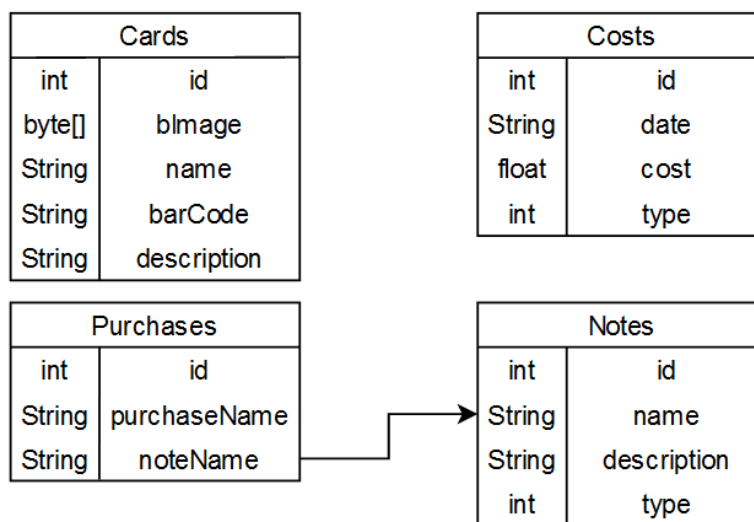


Рисунок 9 – Структура базы данных

- Таблица дисконтных карт
 - a) Id карты;
 - b) Фотография карты;
 - c) Имя карты;
 - d) Штрих-код карты;
 - e) Описание карты.
- Таблица заметок
 - a) Id заметки;
 - b) Имя заметки;
 - c) Описание заметки;
 - d) Тип заметки.
- Таблица с покупками

- a) Id покупки;
 - b) Имя покупки;
 - c) Имя заметки, в которой хранится покупка.
- Таблица чеков
- a) Id чека;
 - b) Дата чека;
 - c) Цена чека;
 - d) Тип покупки.

2.3.2 Выбор интегрированной среды разработки

Для разработки приложений под Android OS существует множество интегрированных сред разработки (IDE) (Таблица 1).

Таблица 1 – Интегрированные среды разработки для Android ОС

Название	Целевые ОС	Поддерживаемые языки	Лицензия	Цена
Android Studio	Android	Java Kotlin C C++	свободная	бесплатно
Eclipse	Android iOS Linux macOS Windows	Java C C++ C# JavaScript Python	Eclipse Public License	бесплатно
IntelliJ IDEA	Любая ОС, которая поддерживает Java	Java Scala Groovy Kotlin JavaScript TypeScript	Apache 2.0	\$500 в год

Android Studio – IDE официально поддерживаемая Google, разработанная на основе программного обеспечения IntelliJ IDEA от JetBrains. Данная IDE является полностью бесплатной и кроссплатформенной и доступна на Windows, macOS и Linux. Из весомых минусов можно отметить только то, что разработка ПО нацелена исключительно на Android ОС.

Eclipse – свободная кроссплатформенная IDE разработанная и развиваемая Eclipse Foundation, поддерживает разработку для всех наиболее популярных платформ (Windows, Linux, macOS, Android, iOS).

IntelliJ IDEA – свободная кроссплатформенная IDE разработанная компанией JetBrains, распространяется под лицензией Apache 2.0, имеет триальную версию сроком на месяц, стоимость полноценного использования составляет 500 долларов в год.

В результате анализа интегрированных сред разработки для Android был выбран следующий программный продукт – Android Studio так, как эта IDE полностью бесплатная, официально поддерживаемая Google и разработанная на основе IntelliJ IDEA.

2.4 Вывод по главе

В результате проектирования приложения были определены архитектурные решения, функциональные требования к приложению и определено какая IDE будет использоваться для разработки приложения.

3 Реализация приложения

Приложение решено было делать в стиле приложения Stocard, с центральной областью под основные элементы приложения (списки с картами и заметками, и гистограммами расходов) и нижнем меню, где располагаются три кнопки переключения между основными вкладками: вкладка с картами, вкладка со списками и вкладка со статистикой расходов. А также с кнопкой, располагающейся над нижнем меню справа, по добавлению новой карты, новой заметки или нового чека, в зависимости какая вкладка открыта.

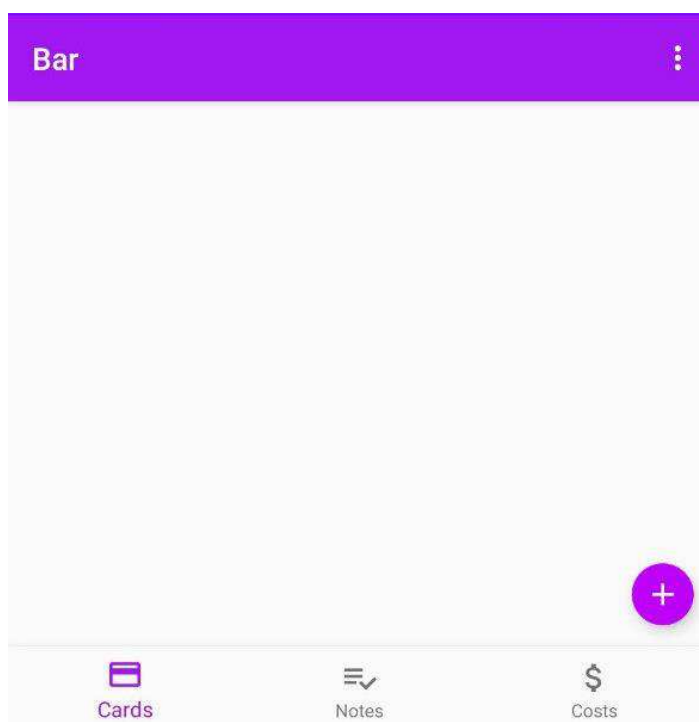


Рисунок 10 – Интерфейс приложения

В центральной основной области MainActivity располагается FrameLayout в который загружаются фрагменты, в зависимости от того какая вкладка выбрана в нижнем меню. Соответственно было сделано три фрагмента: CardsFragment, NotesFragment, CostsFragment.

3.1 Разработка модели

Модель состоит из четырёх основных частей: Entity, Database, DAO и Репозитория.

Было написано четыре класса Entity класса, каждый из них описывает объект, который помещается в базу данных, а именно описываются объекты:

- Card. Служит для описания дисконтной карты;
- Note. Служит для описания одного списка покупок;
- Purchase Служит для описания одной покупки;
- Costs – Служит для описания чека.

Так же для каждого Entity класса было создано по DAO классу: CardDao, NoteDao, PurchaseDao, CostsDao. DAO классы реализуют интерфейсы для работы с базой данных, а именно методы для вставки данных, удаления данных, обновления данных и получения данных из базы данных.

Database – является основным классом по работе с базой данных. В параметрах аннотации указываются версия базы данных и Entity классы, чтобы для каждого из них была создана своя таблица в базе данных. База данных создаётся на внутреннем хранилище смартфона в папке «BarData».

Репозиторий – класс, являющийся прослойкой при работе с базой данных, содержит два типа методов: статические методы асинхронных запросов к базе данных и публичные методы для вызова предыдущих методов.

3.2 Разработка части приложения по работе с картами

Добавленные дисконтные карты отображаются в виде списка карт в фрагменте CardsFragment. Этот список был реализован с помощью RecyclerView (компонент интерфейса, который позволяет нам создавать прокручиваемый список) под именем «card_list». Для card_list потребовалось написать свой адаптер – «CardAdapter», он принимает данные, которые нужно отобразить, и связывает их с card_list. Так же для адаптера был сделан макет для элементов card_list, в который будут помещаться данные карт (Рисунок 11).



Рисунок 11 – Макет для элементов card_list

Кнопка добавления новой карты была реализована с помощью `FloatingActionButton`, при нажатии на неё открывается новая активность - `AddEdit_Card`, в которой идёт добавление новой карты (Рисунок 12).

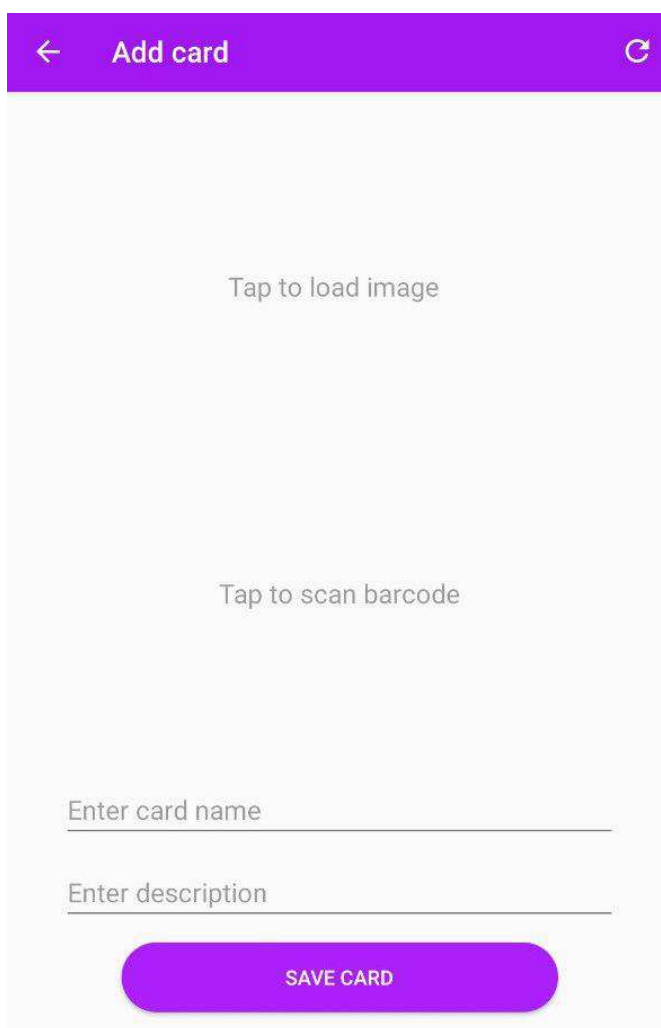


Рисунок 12 – Интерфейс активности `AddEdit_Card`

Каждая новая карта должна содержать отсканированный штрих-код, изображение карты, имя карты и опционально описание. Если новая карта имеет штрих-код, который уже содержится в базе данных, то такая карта не будет добавлена, это сделано для того, чтобы не было дубликатов карт в базе данных. Добавить изображение для карты можно из внутреннего хранилища или с

помощью камеры, после выбора изображения откроется следующая активность, в которой можно будет обрезать изображение под размер карты, эта активность реализуется с помощью библиотеки uCrop [4] от компании Yalantis. Сканирование штрих-кода осуществляется с помощью класса Scanner в котором использовалась библиотека Юрия Будиева - Code-scanner [5], которая поддерживает все актуальные форматы кодов, а также легко имплементируется.

С элементами списка «card_list» можно взаимодействовать (удалять, редактировать, просматривать штрих-код карты на полном экране). Для удаления карты нужно свайпнуть (смахнуть) карту вправо или влево, при этом область экрана, которая находилась под элементом карты подсвечивается красным (Рисунок 13). Для реализации этого жеста был использован служебный класс ItemTouchHelper [8].

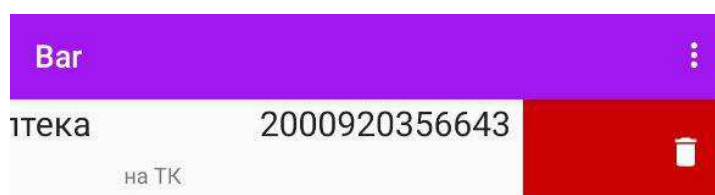


Рисунок 13 – Подсвечивающаяся область экрана при удалении

После удаления карты внизу экрана появится ненадолго Snackbar, сообщающий об удалении карты, в котором содержится кнопка для отмены удаления карты в случае случайного удаления (Рисунок 14).

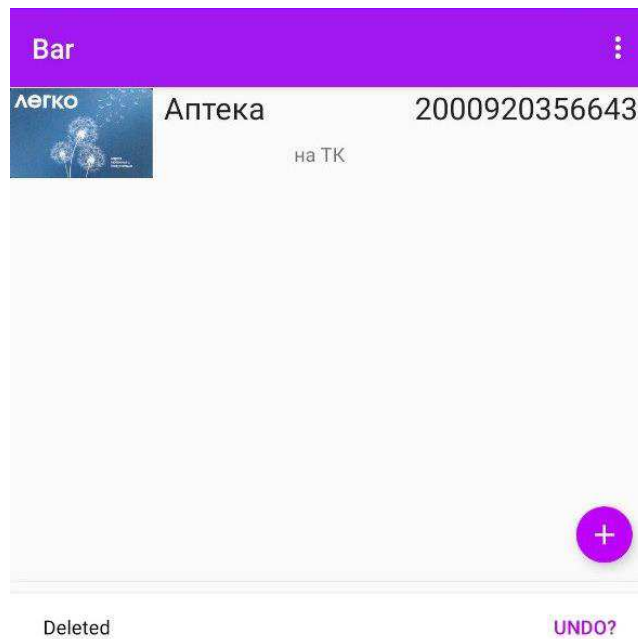


Рисунок 14 – Snackbar предупреждающий об удалении

Для редактирования существующей карты нужно зажать карту пока не появится новая активность, с данными выбранной карты (Рисунок 15), редактировать возможно все параметры карты. Для этого используется `AddEdit_Card`, только при редактировании карты в неё передаются данные редактируемой карты с помощью метода `putExtra` системного класса `Intent`.



Рисунок 15 – Редактирование карты

Для просмотра штрих-кода карты на полный экран нужно нажать на карту (Рисунок 16).



Рисунок 16 – Штрих-код выведенный на полный экран

Для реализации этого был создан класс `BigBarcodeDialog` который реализует создание `AlertDialog`, в нём и отображаются штрих-код и номер, который в нём зашифрован. Для закрытия полноэкранного штрих-кода нужно нажать на любую его область, однако для удобства был добавлен крестик в правый верхний угол, чтобы у пользователя не возникало трудностей с закрытием этого окна.

Так же была создана модель представлений `CardViewModel` для взаимодействия `CardsFragment` и модели. `CardViewModel` содержит методы для добавления карт, обновления карт, удаления карт, проверки существования карты и получения карты по `id`.

3.3 Разработка части приложения по работе со списками покупок

Созданные списки покупок отображаются в виде списка в фрагменте `NotesFragment`. Этот список был так же реализован с помощью `RecyclerView` под именем «`note_list`». Для `note_list` так же потребовалось написать свой адаптер – «`NoteAdapter`», он принимает данные, которые нужно отобразить, и связывает их с `note_list`. Так же для адаптера был сделан макет для элементов `note_list`, в который будут помещаться данные карт (Рисунок 17).

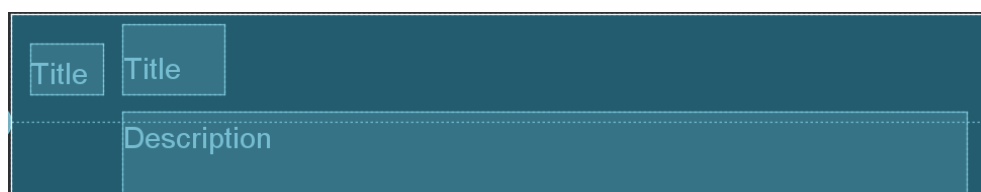


Рисунок 17 – Макет для элементов `note_list`

Кнопка добавления нового списка покупок была реализована с помощью `FloatingActionButton`, при нажатии на неё открывается новая активность - `AddEdit_Note`, в которой идёт добавление нового списка покупок (Рисунок 12).

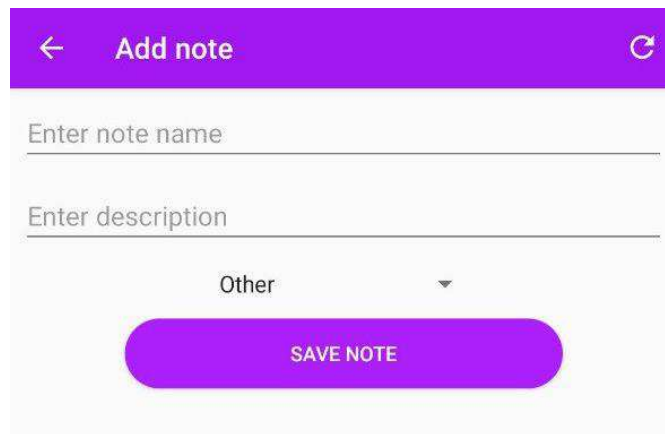


Рисунок 18 – Интерфейс активности AddEdit_Note

Каждая новый список покупок должен иметь имя, опционально описание и тип покупок:

- Еда;
- Электроника;
- Связь (Интернет и телефон);
- Одежда;
- Здоровье;
- Налоги;
- Другое.

Если новый список покупок имеет имя, которое уже содержится в базе данных, то такой список не будет добавлен, это сделано для того, чтобы не было дубликатов списков покупок в базе данных.

С элементами списка «note_list» можно взаимодействовать (удалять, удалять с созданием нового чека и редактировать). Для удаления карты нужно свайпнуть (смахнуть) карту вправо, при этом область экрана, которая находилась под элементом карты подсвечивается красным (Рисунок 13). Для удаления карты с добавлением нового чека нужно свайпнуть (смахнуть) влево, при этом область экрана, которая находилась под элементом карты подсвечивается зелёным (Рисунок 20). Для реализации этих жестов был использован служебный класс ItemTouchHelper.



Рисунок 19 – Подсвечивающаяся область экрана при удалении



Рисунок 20 – Подсвечивающаяся область экрана при удалении с добавлением нового чека

После удаления списка покупок внизу экрана появится ненадолго Snackbar, сообщающий об удалении списка покупок, в котором содержится кнопка для отмены удаления в случае его случайности (Рисунок 21).

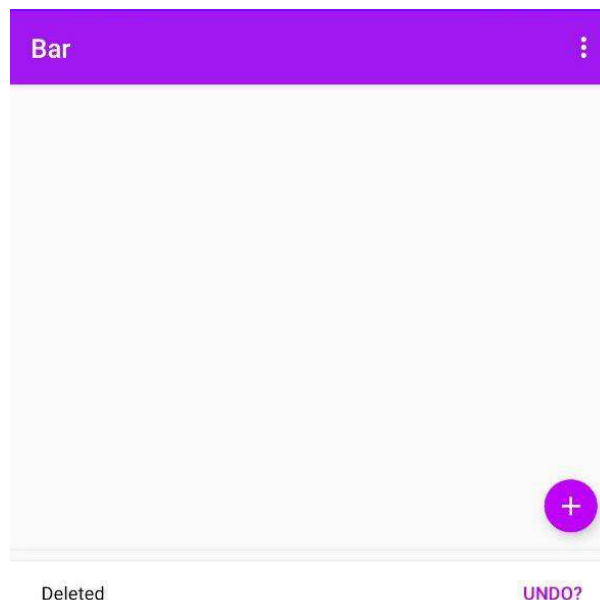


Рисунок 21 – Snackbar предупреждающий об удалении

Однако так как сами покупки хранятся в другой таблице базы данных и для корректного удаления покупок, которые после удаления списка покупок к которому они были привязаны, был создан фоновый сервис

BackgroundCleanerPurchases который удаляет ненужные покупки из таблицы покупок, он начинает свою работу только тогда, когда пропадёт Snackbar, позволяющий отменить удаления списка покупок, а именно через 1500мс.

После удаления списка покупок с созданием нового чека, откроется новая активность AddEdit_QRCheck в которую с помощью метода putExtra системного класса Intent будут переданы данные списка покупок, при этом если новый чек не был создан, то список покупок не удалится.

Для редактирования существующего списка покупок нужно зажать список пока не появится новая активность, с данными выбранного списка редактировать возможно все параметры списка. Для этого используется AddEdit_Note, параметры редактируемого списка передаются так же с помощью метода putExtra.

Для открытия списка покупок нужно нажать на него, при этом откроется новая активность, в которой будет отображен сам список запланированных покупок, в этой же активности можно добавлять и удалять существующие покупки. Для добавления существует поле ввода имени покупки и кнопка добавления, а для удаления так же надо свайпнуть (смахнуть) покупку вправо или влево, это так же реализовано с помощью ItemTouchHelper (Рисунок 22).

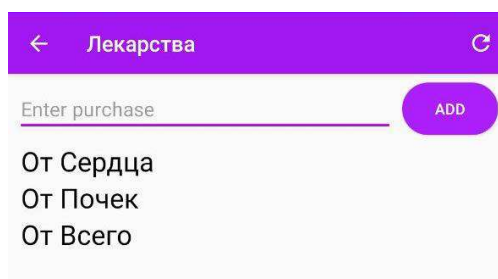


Рисунок 22 – Список запланированных покупок

Так же были созданы модели представлений NotesViewModel и PurchasesViewModel для их взаимодействия моделью. NotesViewModel и PurchasesViewModel содержит методы для добавления списков, обновления списков, удаления списков и проверки существования списка.

3.4 Разработка части приложения по работе с расходами

Фрагмент, отображающий расходы CostsFragment, содержит две гистограммы, которые были построены с помощью библиотеки MPAndroidChart [6]. Первая гистограмма отображает расходы в каждом месяце за последний год (Рисунок 23), вторая гистограмма является круговой и отображает расходы по категориям в выбранном месяце (Рисунок 24).

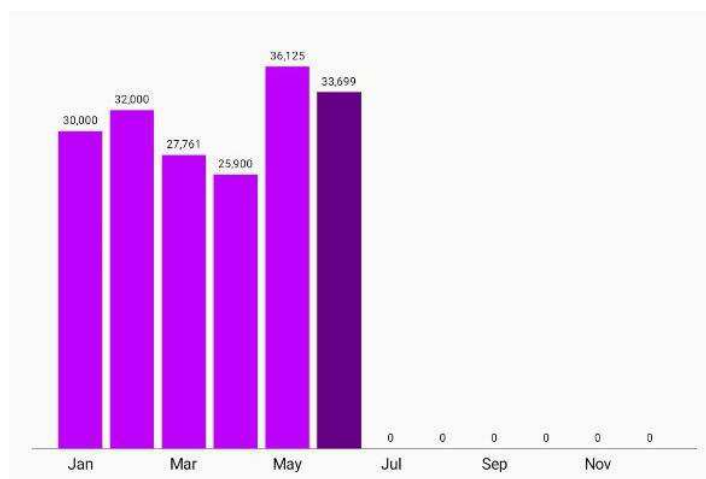


Рисунок 23 – Гистограмма с расходами по месяцам



Рисунок 24 – Круговая гистограмма с расходами за месяц

При первом открытии вкладки с расходами круговая гистограмма будет отображать расходы за текущий месяц.

Кнопка добавления нового чека была реализована с помощью FloatingActionButton, при нажатии на неё открывается новая активность - AddEdit_QRCheck, в которой идёт добавление нового чека (Рисунок 25).

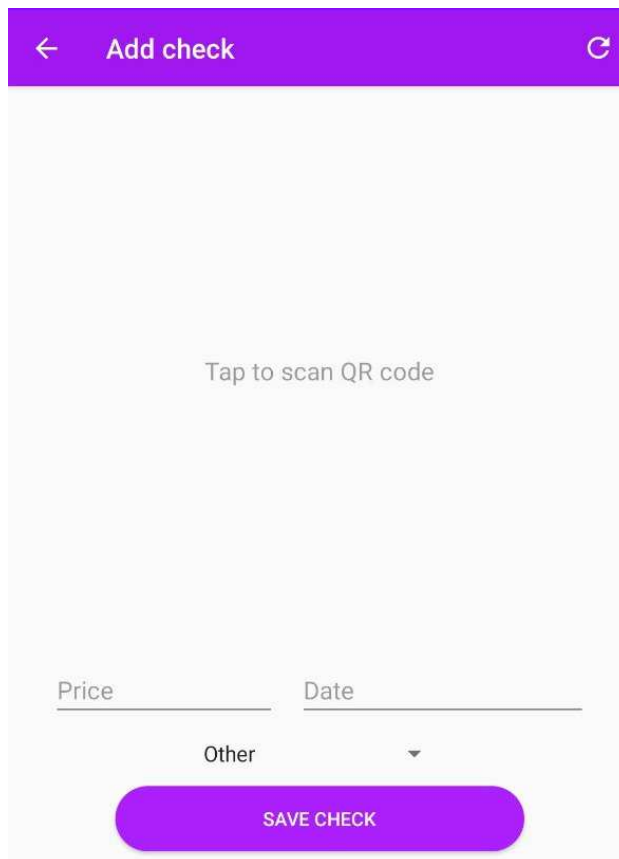


Рисунок 25 – Активность добавления нового чека

При добавлении нового чека, этот новый чек должен обязательно содержать дату покупки, цену чека и тип покупки. Все эти данные можно ввести вручную, однако можно и отсканировать QR код, который находится в конце любого физического чека, и получить цену и дату покупки. Сканированию будет выполняться с помощью класса `Scanner`, который так же использовался при сканировании штрих-кодов с дисконтных карт. Отсканированный QR код будет отрисован с помощью библиотеки `QRGenerator` [7] в верхней части окна добавления нового чека.

Просмотреть список чеков можно нажать на первый пункт меню верхнего бара, при этом откроется новая активность (Рисунок 26), которая содержит `RecyclerView` под именем «`costList`» и для него так же был написан адаптер и создан макет элемента чека (Рисунок 27)

← Bar		
Other:	6.0	11/3/2020
Other:	99.0	12/6/2020
Other:	30000.0	15/1/2020
Clothes:	32000.0	12/2/2020
Health:	27755.0	10/3/2020
Other:	25900.0	17/4/2020
Other:	36125.0	16/5/2020
Communications:	5000.0	8/6/2020
Food:	12000.0	8/6/2020
Health:	3000.0	8/6/2020
Electronics:	10000.0	8/6/2020
Communications:	600.0	20/6/2020
Other:	3000.0	8/6/2020

Рисунок 26 – Список чеков



Рисунок 27 – Макет для элементов costList

Удалить чек из списка можно свайпом вправо или влево, при этом снизу всплывёт Snackbar сообщающий об удалении и содержащий кнопку восстановления чека.

Так же удаление чеков происходит автоматически, удаляются чеки, которые были добавлены в прошлом году, для этого был создан фоновый сервис BackgroundCleanerPurchases, запускающийся каждый раз при старте приложения.

Так же была создана модель представлений CostsViewModel для взаимодействия CostsFragment и модели. CostsViewModel содержит методы для добавления чеков, обновления чеков, удаления чеков и проверки существования чека.

3.5 Реализация настроек приложения

Настройки приложения реализованы с помощью PreferenceFragmentCompat, он создаёт в активности фрагмент, в котором отображается вся иерархия опций, которые вы добавили в XML файле активности. Все эти опции сохраняются в автоматически создаваемый файл SharedPreferences. А загружаются они из этого же файла при старте программы.

Были добавлены настройки по изменению языка, возможности импортировать и экспортировать базу данных, а также полностью отчищать таблицу с картами, с расходами и списками покупок (Рисунок 28).

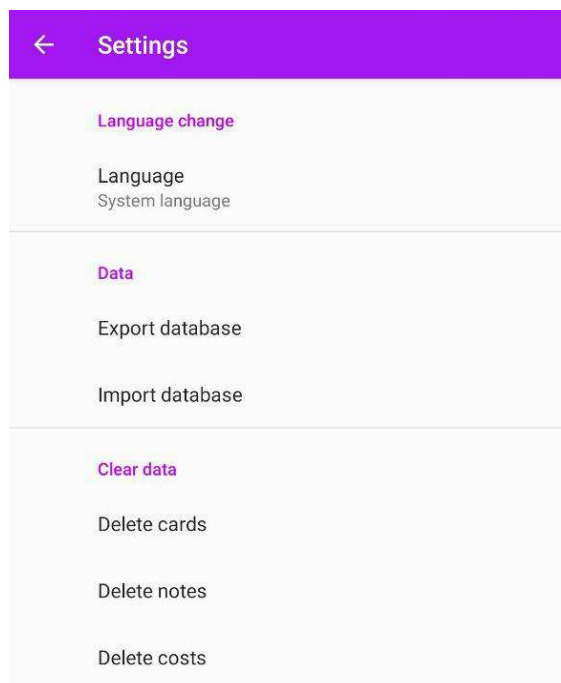


Рисунок 28 – Экран настроек приложения

При изменении языка, приложение сразу меняет язык на выбранный, в активности у каждого элемента меняется текст, а в остальные активности, которые пользователь не видит, перезагружаются.

Экспорт базы данных происходит из папки приложения в корень внутреннего хранилища смартфона. При импорте базы данных открывается системный проводник, в котором нужно выбрать файл базы данных, этот файл обязательно должен иметь название “card_database.db”

3.6 Вывод по главе

Было разработано мобильное Android приложение по сформированным требованиям и функционалу, были выбраны методы и средства разработки требуемого функционала приложения.

4 Инструкция пользователя

При первом запуске приложение потребует предоставить ему разрешения на пользование камерой и внутренним хранилищем (Рисунок 29).

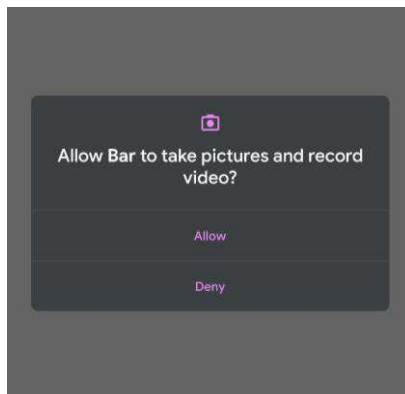


Рисунок 29 – Требование разрешений

Если приложению будет отказано, то при каждом действии в приложении, оно будет запрашивать разрешения, а также будет выведен Snackbar, где можно вывести окно предоставления разрешений (Рисунок 30).

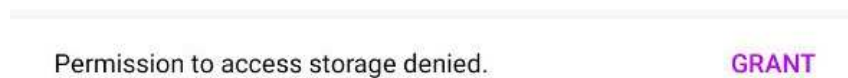


Рисунок 30 – Snackbar с требованием разрешений

После того как пользователь предоставил все необходимые разрешения приложению, пользователь может:

- Взаимодействовать с картами;
- Взаимодействовать со списками покупок;
- Взаимодействовать с расходами.

4.1 Взаимодействие с картами

Так как пользователь только поставил приложение, то в базе данных нет никаких карт, для добавления карты нужно нажать на круглую кнопку в нижнем правом углу (Рисунок 31).

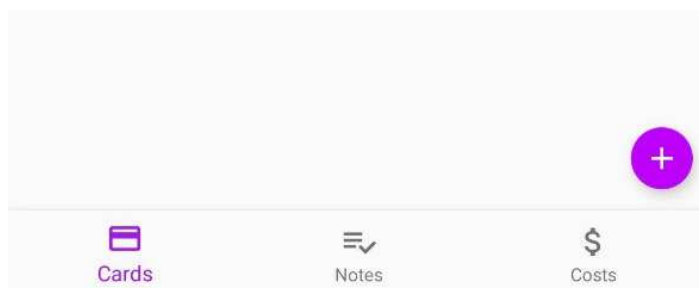


Рисунок 31 – Кнопка для добавления новой карты

После этого откроется новое окно, где нужно ввести данные новой карты, обязательны все поля кроме описания (Рисунок 32).

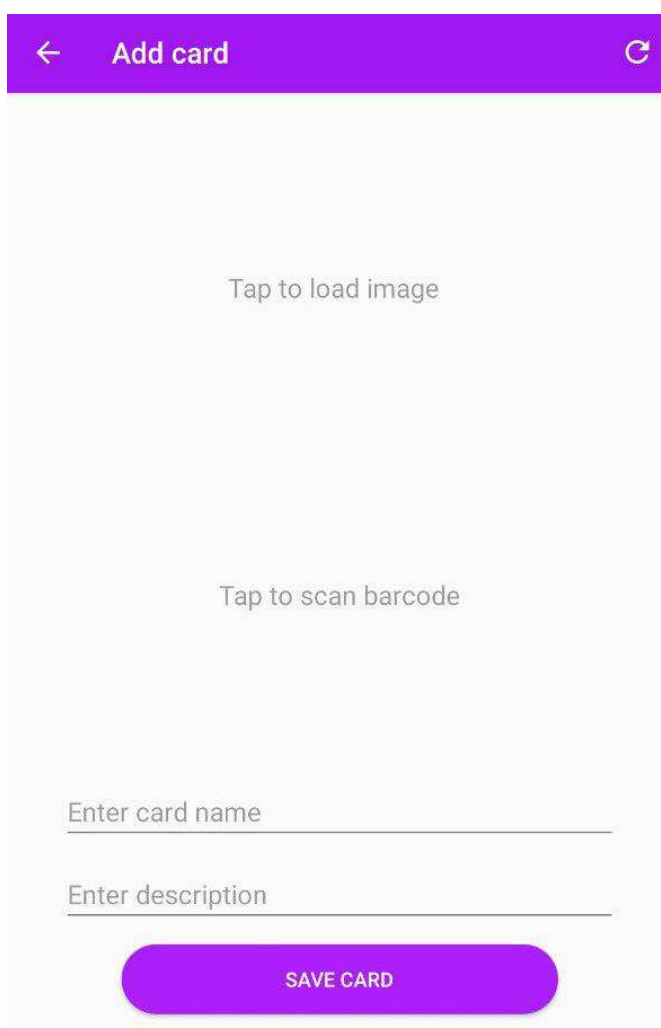


Рисунок 32 – Добавление новой карты

После добавления новой карты, она отобразится на главном экране. С ней можно взаимодействовать: открыть её штрих-код на полный экран, редактировать и удалить. Для открытия штрих-кода на полный экран нужно

однократно нажать на карту. Для редактирования нужно зажать карту, тогда появится окно, где можно изменить данные карты (Рисунок 33).



Рисунок 33 – Редактирование карты

Для удаления нужно свайпнуть (смахнуть) карту вправо или влево. После этого появится Snackbar с сообщением об успешном удалении и кнопкой отмены удаления (Рисунок 34).



Рисунок 34 – Сообщение об удалении карты

4.2 Взаимодействие со списками покупок

Для взаимодействия с списками покупок нужно перейти на вкладку с списками, для этого нужно в нижнем меню выбрать вторую вкладку. Далее можно создать новый список покупок нажав на круглую кнопку в правом нижнем углу экрана. После этого появится окно добавления новой вкладки (Рисунок 35).

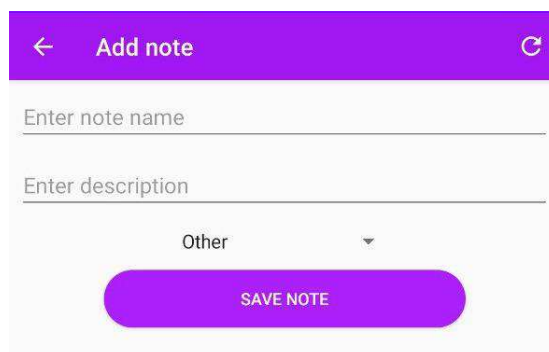


Рисунок 35 – Окно добавления нового списка покупок

В этом окне обязательно ввести имя списка и его категорию, описание не обязательно. Далее, после создания нового списка в окне появится новая строчка с этим списком (Рисунок 36).

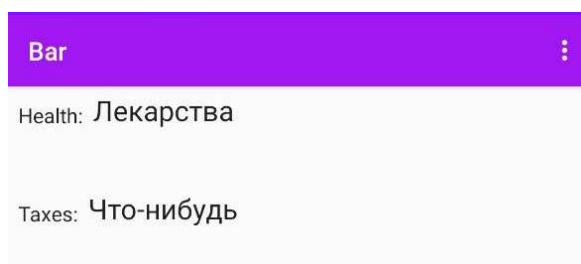


Рисунок 36 – Списки покупок

Эти списки можно: открывать, удалять, удалять с добавлением нового чека и редактировать. Для открытия списка нужно однократно нажать на него, после этого откроется окно, где можно посмотреть и отредактировать содержимое списка (Рисунок 37).

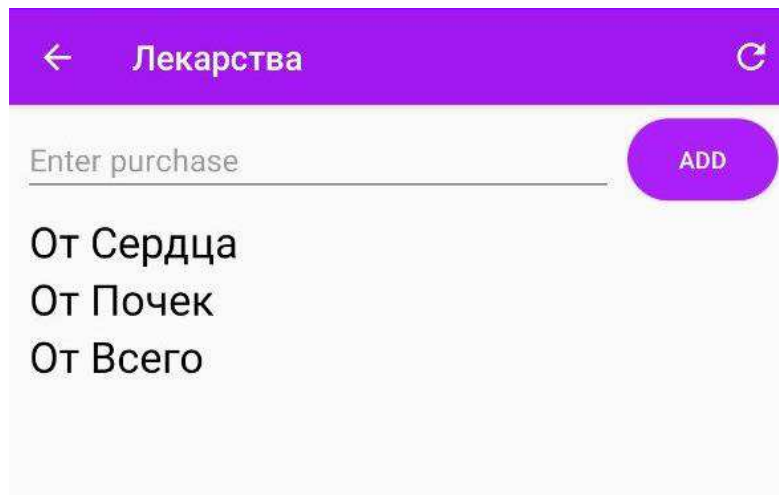


Рисунок 37 – Содержание списка покупок

Для удаления списка покупок нужно свайпнуть вправо, при этом появится снизу Snackbar, где можно отменить удаление (Рисунок 34). Для удаления с добавлением чека нужно свайпнуть влево, после этого появится окно добавления нового чека (Рисунок 40), если же чек не будет добавлен, то и список покупок не будет удалён. Для редактирования названия, типа и описания списка нужно нажать этот список и тогда откроется окно редактирования его параметров (Рисунок 38).

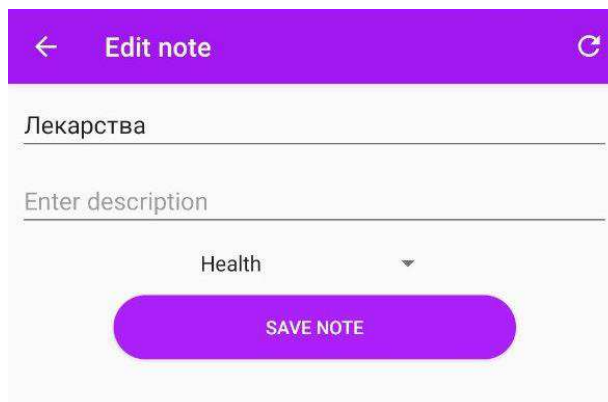


Рисунок 38 – Редактирование параметров списка

4.3 Взаимодействие с расходами

Для взаимодействия с расходами нужно перейти на вкладку с списками, для этого нужно в нижнем меню выбрать третью вкладку. На этой вкладке располагаются две гистограммы одна обычная, а вторая круговая. Первая

отображает кол-во расходов в каждом месяце. Вторая отображает расходы, разбитые по категориям в конкретном месяце (Рисунок 39).

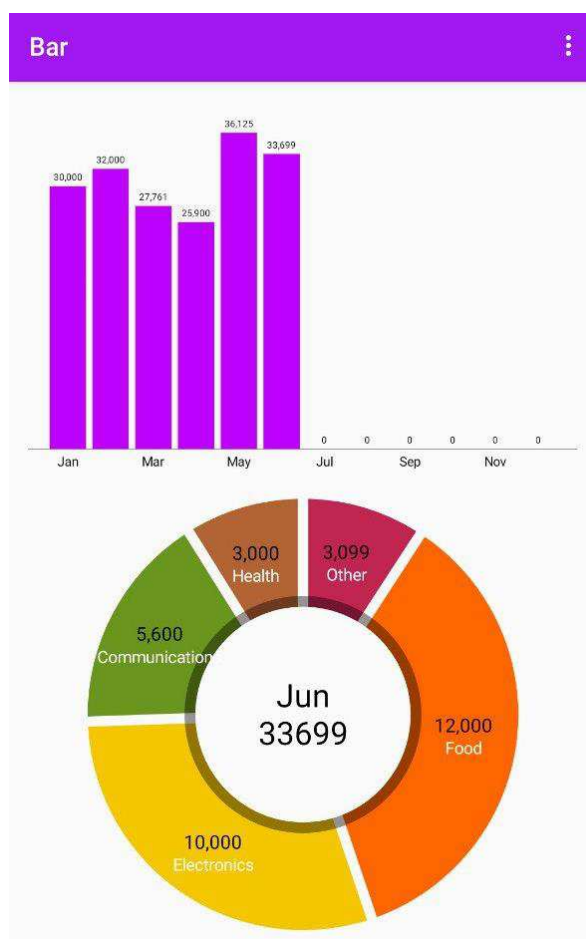


Рисунок 39 – Гистограммы расходов

Чтобы добавить новый чек нужно нажать на правую нижнюю кнопку внизу, после этого откроется окно добавления нового чека (Рисунок 40). Данные чека можно добавить вручную, а можно отсканировать с физического чека. При добавлении чека изменения сразу же будут видны на гистограммах.

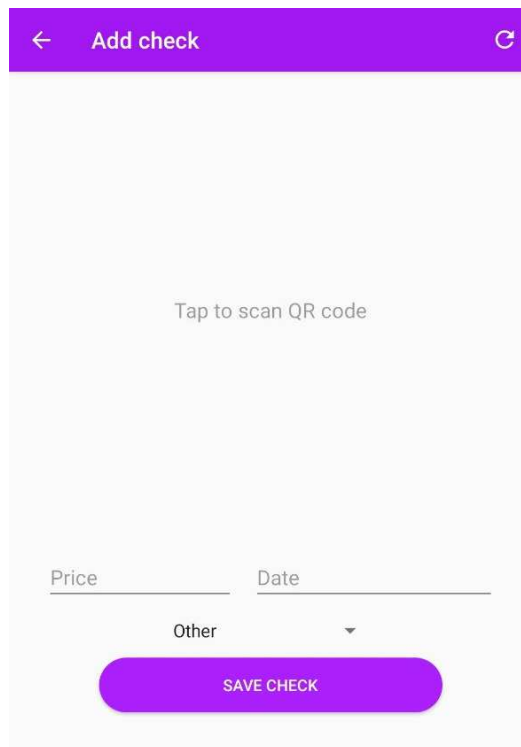


Рисунок 40 – Добавление чека

Для того чтобы открыть список чеков нужно нажать в верхнем баре кнопку меню и выбрать первый пункт, откроется окно со списком всех чеков (Рисунок 41).

	Price	Date
Other:	6.0	11/3/2020
Other:	99.0	12/6/2020
Other:	30000.0	15/1/2020
Clothes:	32000.0	12/2/2020
Health:	27755.0	10/3/2020
Other:	25900.0	17/4/2020
Other:	36125.0	16/5/2020
Communications:	5000.0	8/6/2020
Food:	12000.0	8/6/2020
Health:	3000.0	8/6/2020
Electronics:	10000.0	8/6/2020
Communications:	600.0	20/6/2020
Other:	3000.0	8/6/2020

Рисунок 41 – Список всех чеков

В этом списке их можно удалять свайпом вправо или влево, а также редактировать долгим зажатием.

4.4 Настройки приложения

Настройки приложения можно открыть если нажать на меню в верхнем баре и нажать на второй пункт (Рисунок 42).

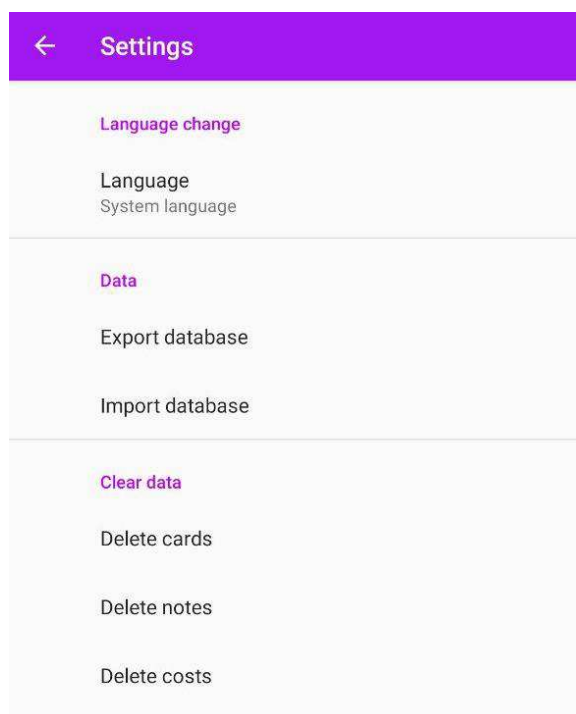


Рисунок 42 – Окно настроек приложения

Здесь можно поменять язык (Русский, Английский), экспортировать или импортировать базу данных, а также отчистить её.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы была достигнута поставленная цель – создание приложения мобильного приложения для организации и учёта покупок с учётом имеющихся дисконтных карт. Для её достижения были выполнены следующие задачи:

- был проведен обзор аналогов, благодаря которому был определен список функциональных требований и получено представление о создаваемой системе.

- были проведено проектирование системы, а именно была определена структура приложения, с учётом структуры приложения был проведён выбор средств разработки. Было реализовано приложение на основе шаблона MVVM, для его реализации были разработаны структуры представлений, моделей представлений и модели.

- реализовано мобильное приложение которое полностью удовлетворяет поставленным требованиям, а именно приложение имеет возможность составления списков покупок по категориям и хранения их в базе данных, возможность сканирования QR кодов для считывания информации о покупке, возможность сканирования штрих-кодов дисконтных карт для хранения их в базе данных, возможность экспортировать базу данных с отсканированными дисконтными картами, составленными списками покупок и сохранёнными расходами для возможности её импорта на другом мобильном устройстве, а так же приложение ведёт учет потраченных средств за определённые промежутки времени на основе добавленных покупок.

Данная бакалаврская работа написана в соответствии с нормами СТО 4.2–07–2014 Система менеджмента качества [11].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Google Play [Электронный ресурс]: Страница приложения Stocard - Режим доступа: <https://play.google.com/store/apps/details?id=de.stocard.stocard> (дата обращения: 4.03.2020)
2. Google Play [Электронный ресурс]: Страница приложения PINbonus. Режим доступа: <https://play.google.com/store/apps/details?id=com.pinbonus2> (дата обращения: 4.03.2020)
3. Google Play [Электронный ресурс]: Страница приложения Кошелёк. Режим доступа: <https://play.google.com/store/apps/details?id=ru.cardsmobile.mw3> (дата обращения: 4.03.2020)
4. GitHub [Электронный ресурс]: Страница библиотеки Ucrop в репозитории компании Yalantis. Режим доступа: <https://github.com/Yalantis/uCrop> (дата обращения: 15.05.2020)
5. GitHub [Электронный ресурс]: Страница библиотеки Code Scanner в репозитории Юрия Будиева. Режим доступа: <https://github.com/yuriy-budiyev/code-scanner> (дата обращения: 15.05.2020)
6. GitHub [Электронный ресурс]: Страница библиотеки MPAndroidChart в репозитории Филипа Джаоды. Режим доступа: <https://github.com/PhilJay/MPAndroidChart> (дата обращения: 15.05.2020)
7. GitHub [Электронный ресурс]: Страница библиотеки QRGenerator в репозитории Android Mad. Режим доступа: <https://github.com/androidmads/QRGenerator> (дата обращения: 15.05.2020)
8. Android Developers [Электронный ресурс]: Официальная страница для Android разработчиков. Страница с инструкцией по использованию ItemTouchHelper Режим доступа: <https://developer.android.com/reference/kotlin/androidx/recyclerview/widget/ItemTouchHelper> (дата обращения: 15.05.2020)
9. Android Developers [Электронный ресурс]: Официальная страница для Android разработчиков. Страница с инструкцией по использованию

библиотеки Room Режим доступа: <https://developer.android.com/training/data-storage/room> (дата обращения: 15.05.2020)

10. SQLite [Электронный ресурс]: Официальная страница проекта SQLite Режим доступа: <https://www.sqlite.org/index.html> (дата обращения: 15.05.2020)

11. СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 09.01.2014. – Красноярск: СФУ, 2014. – 60 с.

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

Д. А. Швец
подпись инициалы, фамилия

« » 20 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Программная система организации и учёта покупок
тема

Пояснительная записка

Руководитель

Ш
подпись, дата

Доктор, канд. техн. наук
должность, ученая степень

Д. А. Швец
инициалы, фамилия

Выпускник

Ш 26.06.20
подпись, дата

Д. А. Частоступов
инициалы, фамилия

Нормоконтролер

Ш
подпись, дата

Доктор, канд. техн. наук
должность, ученая степень

Д. А. Швец
инициалы, фамилия

Красноярск 2020