

Обоснование:

.....
.....
.....
.....
.....
.....

16.05.19.

Дата



Подпись Научного руководителя

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Супрун Родион

Название: Диплом на антиплагиат

Координатор: Гульнара Омарова

Коэффициент подобия 1: 0,6

Коэффициент подобия 2: 0

Тревога: 1

После анализа Отчета подобия констатирую следующее:

- ☐ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- ☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- ☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на _____ дипломный проект
(наименование вида работы)
_____ Супрун Родиона
(Ф.И.О. обучающегося)
_____ 5B070400 – Вычислительная техника и программное обеспечение
(шифр и наименование специальности)

Тема: «Разработка мобильного приложения для поиска и бронирования авиабилетов на платформе Android»

Данная тема является важной и актуальной по ряду причин. В настоящее время международные исследования показывают, что больше половины населения Земли пользуется интернетом с мобильных устройств. В связи с этим, нельзя представить крупную корпорацию без собственного мобильного приложения. Дипломный проект состоит из 4-х разделов, включая введение, заключение и приложений.

Первый раздел работы раскрывает цель разработки мобильного приложения, а также проводится сравнение приложения с основными конкурентами на рынке.

Во втором разделе сделано обоснование выбора инструментов и технологий для реализации приложения для мобильных телефонов.

В третьем разделе была описана проектная документация, включающая в себя описание архитектуры системы и самого приложения, так же приведены диаграмма активности и диаграмма вариантов использования.

Четвертым разделом является “Экспериментальный раздел”. В нем был описан полный функционал мобильного приложения. Использование данного приложения позволит получить доступ к онлайн - услугам системы бронирования ETS на любом телефоне на платформе Android

В целом данный дипломный проект выполнен с учетом всех требований, предъявляемых к дипломному проекту по специальности 5B070400 - «Вычислительная техника и программное обеспечение», студент Супрун Р. рекомендован к защите дипломного проекта.

Научный руководитель

лектор

(должность, уч. степень, звание)



Омарова Г. А.

(подпись)

« 16 »

мая

2019г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекомм уникационных технологий

Кафедра "Компьютерная и программная инженерия"

Супрун Р.П.

Мобильное приложение для поиска и бронирования авиабилетов на платформе
Android

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное
обеспечение

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

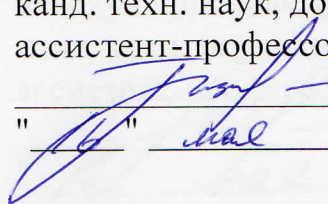
Институт информационных и телекоммуникационных технологий

Кафедра " Компьютерная и программная инженерия "

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой ПИ

канд. техн. наук, доцент,
ассистент-профессор

 Р. Юнусов
" 16 " май 2019 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

На тему: "Мобильное приложение для поиска и бронирования авиабилетов на платформе Android"

по специальности 5B070400 – Вычислительная техника и программное обеспечение

Выполнил

Супрун Р.П.

Научный руководитель

лектор

 Г.А.Омарова
" 16 " май 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
СӘТБАЕВ УНИВЕРСИТЕТІ

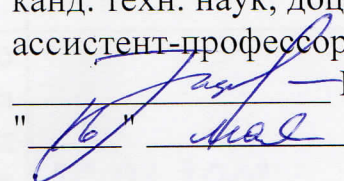
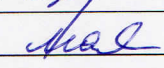
Институт информационных и телекоммуникационных технологий

Кафедра "Компьютерная и программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

УТВЕРЖДАЮ

Заведующий кафедрой ПИ
канд. техн. наук, доцент,
ассистент-профессор

 Р. Юнусов
" 16 "  2019 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Супрун Родиону Павловичу

Тема: Мобильное приложение для поиска и бронирования авиабилетов на платформе Android

Утверждена приказом проректора по академической работе № 1162 - б
от "16" октября 2018 г.

Срок сдачи законченного проекта "17" мая 2019 г.

Исходные данные к дипломному проекту: Техническое задание, описание необходимых функций проекта.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) реализация всех функций системы ETS для бронирования авиабилетов на платформе Android;

б) проектирование и разработка пользовательского интерфейса;

в) разработка, отладка, тестирование программного комплекса.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 15 слайдами презентации.

Рекомендуемая основная литература: из 10 наименований.

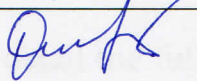
ГРАФИК

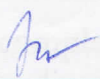
подготовки дипломного проекта

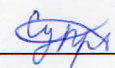
Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области, разработка технического задания	05.11.2018	—
2. Проектирование интерфейса, логики приложения	15.01.2019	—
3. Реализация поиска авиаперелетов по выбранным критериям и вывод их на экран, а так же возможность детального обзора любого выбранного перелета. Реализация бронирования авиаперелета. Реализация вывода истории заказов. Реализация интерфейса пользователя.	26.03.2019	—
4. Тестирование программного обеспечения	02.04.2019	—
5. Написание пояснительной записки к дипломному проекту	06.05.19	—

Подписи

консультантов и нормоконтролера на законченный дипломный проект
с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	Г.А.Омарова Лектор	06.05.19.	
Программное обеспечение	Өмірбекова З.М Ассистент	06.05.19	

Научный руководитель  Г.А.Омарова

Задание принял к исполнению обучающийся  Р.П. Супрун

Дата "10" мая 2019 г.

АННОТАЦИЯ

Дипломный проект является мобильным приложением на платформе Android для поиска и бронирования авиабилетов.

Во введении дипломной работы рассказывается о цели проекта и выполненных задачах.

Первый раздел работы "Исследовательский раздел" раскрывает цель разработки мобильного приложения.

Второй раздел посвящен обзору выбранных инструментов и технологий для реализации данного ПО для мобильных телефонов.

В третьем разделе была создана проектная документация, включающая в себя описание архитектуры системы и самого приложения.

Четвертым разделом является "Экспериментальный раздел". В нем был подробно описан полный функционал мобильного приложения.

Результатом данного дипломного проекта является законченное мобильное приложение на платформе Android, которым будут пользоваться корпоративные клиенты Global air.

ANNOTATION

The graduation project is a mobile application on the Android platform for searching and booking air tickets.

The introduction describes the purpose of the project and the tasks performed.

The first section of the work "Research Section" reveals the purpose of developing a mobile application.

The second section is devoted to the review of selected tools and technologies for the implementation of this software for mobile phones.

In the third section, the project documentation was created, which includes a description of the system architecture and the application itself.

The fourth section is the "Experimental Section". It described in detail the full functionality of the mobile application.

The result of this graduation project is a complete mobile application on the Android platform, which will be used by corporate clients of Global air.

АНДАТПА

Дипломдық жоба - авиабилеттерді іздеу және брондау үшін Android платформасында мобильді қосымша.

Диссертацияда жобаның мақсаты мен орындалатын тапсырмалар

сипатталады.

Жұмыстың бірінші бөлімі «Зерттеу бөлімі» мобильдік қосымшаны әзірлеудің мақсатын анықтайды.

Екінші бөлік ұялы телефондарға арналған осы бағдарламалық жасақтаманы енгізу үшін тандап алынған құралдар мен технологияларды шолуға арналған.

Үшінші бөлімде жүйелік архитектураның сипаттамасы мен қосымшаның өзі қамтылған жобалық құжаттама жасалды.

Төртінші бөлім - «Эксперименттік бөлім». Ол ұялы қосымшаның толық функционалдығын егжей-тегжейлі сипаттады.

Бұл бітіру жобасының нәтижесі - Global Air компаниясының корпоративтік клиенттері пайдаланатын Android платформасында толық мобильді қосымша.

СОДЕРЖАНИЕ

	Введение	8
1	Исследовательский раздел	9
1.1	Цель разработки мобильного приложения	9
1.2	Термины и сокращения	9
1.3	Обзор аналогичных приложений	10
1.3.1	Обзор мобильного приложения Air Astana	10
1.3.2	Обзор мобильного приложения Aviata	14
2	Технологический раздел	17
2.1	Обзор используемых инструментов	17
2.1.1	Андроид ОС	17
2.1.2	Android Studio	17
2.1.3	Язык программирования Java	20
2.1.4	Библиотека Volley	20
2.1.5	Firebase Cloud Messaging	22
3	Проектный раздел	23
3.1	Архитектура системы	23
3.2	Архитектура приложения	24
3.3	Проектирование приложения с использованием UML диаграмм	25
3.3.1	Проектирование приложения с помощью диаграммы вариантов использования	25
3.3.2	Проектирование приложения с помощью диаграммы деятельности	26
3.4	Проектирование интерфейса мобильного приложения	27
4	Экспериментальный раздел	29
4.1	Функции мобильного приложения ETS	29
4.2	Создание мобильного приложения ETS	29
4.2.1	Создание страницы авторизации и регистрации	29
4.2.2	Создание страницы поиска авиаперелетов	31
4.2.3	Создание страницы загрузки авиаперелетов	34
4.2.4	Создание страницы для вывода списка авиаперелетов	34
4.2.5	Создание страницы для фильтрации авиаперелетов	35
4.2.6	Создание страницы “Детали авиаперелета”	36
4.2.7	Создание страницы “Правила обмена и возврата”	37
4.2.8	Создание страницы для ввода данных пассажиров	38
4.2.9	Создание страницы для вывода списка заказов	39
4.2.10	Создание страницы “Мой профиль”	40
	Заключение	43
	Список использованной литературы	44
	Приложение А. Техническое задание	45
	Приложение Б. Текст программы	49

ВВЕДЕНИЕ

Свое введение дипломной работы я хочу начать с рассказа об истории и роли мобильного телефона в нашей жизни. Мобильные телефоны появились в 70-х годах прошлого века. Первый звонок был сделан 3 апреля 1973 года. Он ознаменовал старт новой эры – эры мобильных устройств. Сегодня почти невозможно найти человека, который хотя бы раз в день не держал данный аппарат в руке. Телефон в наше время – это не редкость и не роскошь, это незаменимый помощник. Для того чтобы найти какую-нибудь вещь в интернете, нам не нужен персональный компьютер, достаточно подключить телефон к всемирной сети - Интернет.

Все это плавно подводит к выводу об актуальности дипломной работы на тему “Разработка мобильного приложения для бронирования и поиска авиабилетов”. Актуальность выбранной темы обусловлена тем, что международные исследования показывают, что больше половины населения Земли пользуется интернетом с мобильных устройств. В связи с этим, если раньше нельзя было представить крупную корпорацию без многоуровневого хорошо организованного сайта, то в наше время наличие мобильного приложения обязательно. На это есть несколько причин. Во-первых, приложения реализуют легкий доступ к информации, не нужно запоминать конкретный URL – адрес. Во – вторых, приложение может взаимодействовать с пользователем через notifications (уведомления), наличие уведомлений – одна из главных причин разработки приложения, по моему мнению.

Практическая часть дипломного проекта является решением возникшей необходимости компании Глобал эйр в мобильном аналоге их собственной системы онлайн бронирования услуг, для адаптации к современным условиям ведения бизнеса.

Целью дипломной работы является разработка мобильного приложения для поиска и бронирования авиабилетов, на основе существующей системы бронирования услуг туристической компании GLOBAL AIR – ETS.

Достижение указанной цели осуществлялось посредством решения следующих основных задач:

1. Изучить все возможности системы online – бронирования услуг компании
2. Выбрать и изучить RESTfull библиотеку для взаимодействия мобильного приложения с сервером через POST/GET запросы.
3. Реализовать в мобильном приложении весь возможный функционал данной системы бронирования, начиная от регистрации и заканчивая бронированием авиаперелета на выбранное количество человек.
4. Реализовать удобный и красивый интерфейс пользователя, основанный на преимущественности приложения с сайтом компании.
5. Выполнить все требования заказчика по разработке и сопровождению мобильного приложения

1 Исследовательский раздел

1.1 Цель разработки мобильного приложения

Global air является туристической компанией, которая не использует существующие аналоги программного обеспечения для предоставления тревел услуг, так как в сфере туризма очень много конкурентов со своими плюсами и минусами. И пользоваться таким же ПО, как у конкурентов было бы глупо, компания сразу теряет свою конкурентоспособность, так как не получится ответить на вопросы клиентов “В чем ваша изюминка? Почему именно вашей системой бронирования должен я воспользоваться, если она такая же как у других компаний?”. Именно по этой причине компания стала заниматься разработкой собственной системы бронирования ETS. Данная система предоставляет широкий спектр возможностей для поиска и бронирования как авиаперелетов, так и ЖД билетов, но с одним замечанием – весь функционал доступен только для зарегистрированных корпоративных клиентов компании.

Дипломный проект является мобильной версией онлайн системы бронирования компании. Мобильное приложение реализует весь возможный функционал системы в сфере авиаперелетов: поиск, просмотр, фильтрация и бронирование авиаперелетов, просмотр своих заказов и редактирование личного профиля. Приложение работает в постоянной взаимосвязи с сервером, а значит требуется стабильное Интернет-соединение.

1.2 Термины и сокращения

В данном дипломном проекте использованы некоторые термины и сокращения, которые отображены в таблице 1.

Таблица 1 – Сокращения, термины и их определения

Сокращение или термин	Определение
ОС	Операционная система
БД	База данных
IDE	Интегрированная среда разработки
URL	Унифицированный указатель ресурса
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol
API	Application Programming Interface
CRUD	Создать, прочесть, обновить, удалить
UML	Unified Modeling Language

1.3 Обзор аналогичных приложений

Google Play наводнен на данный момент приложениями для поиска и бронирования авиаперелетов. Приложения различаются не только внешним видом, но и функциональностью, так как каждое мобильное приложение реализует функционал своей десктопной версии.

Рассмотрим функционал самых популярных приложений в Казахстане для поиска и бронирования авиабилетов, а именно Air Astana и Aviata. Приложение Air Astana является разработкой национального авиаперевозчика РК, крупнейшей авиакомпания нашей страны. Aviata – одно из крупнейших авиа-агентств в Казахстане.

Функции обоих приложений схожи, перечислю некоторые из них:

1. Поиск и бронирования авиабилетов
2. Фильтрация авиаперелетов
3. Проверка и отслеживание статуса рейса
4. Уведомления
5. Оплата брони

1.3.1 Обзор мобильного приложения Air Astana

Плюсы приложения Air Astana:

1. Красивое и функционально удобная страница поиска авиаперелета (рисунок 1)
2. Удобное навигационное меню, находящееся прямо под пальцами рук
3. Широкие возможности фильтрации перелетов
4. Возможность использования электронной карты “Номад Клуб”
5. Возможность оплаты перелета

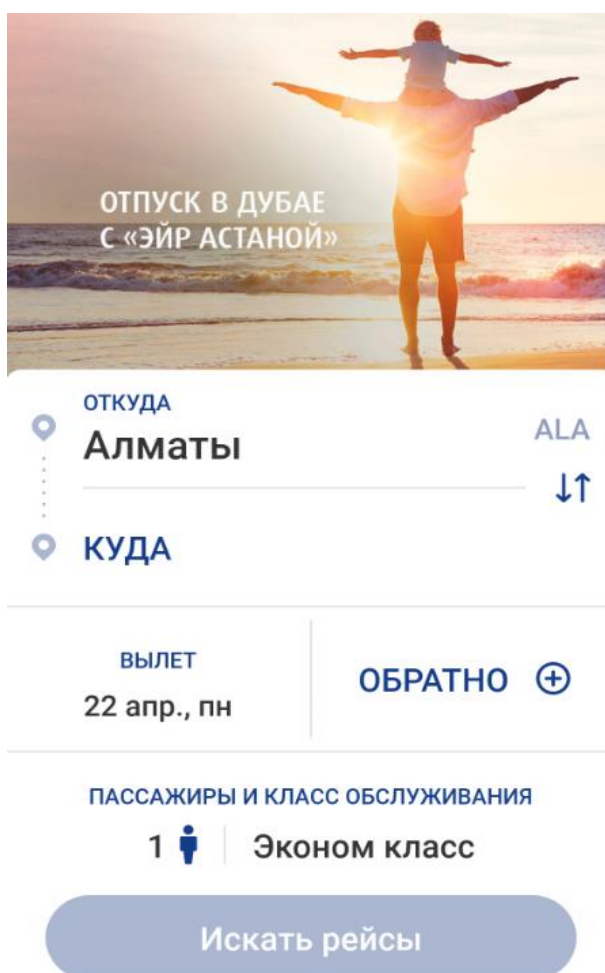


Рисунок 1.1 – Страница поиска авиаперелета в приложении Air Astana

Минусы приложения Air Astana:

1. Ожидание поиска авиаперелетов происходит долго и с зависанием, не совсем понятно приложение работает или зависло. Очень часто приложение очень долго грузится при нажатии на любую кнопку
2. Поиск подходящих перелетов происходит только по рейсам Air Astana, а значит список городов для перелета ограниченно, так как авиакомпания не летает во все города мира. А значит, например, перелет Алматы – Нью-Йорк не найти.
3. Интерфейс вывода авиаперелетов слишком загружен различными элементами в следствии чего, сами перелеты еще надо найти в этом бардаке (рисунок 2 и рисунок 3)
4. После просмотра подробностей перелета нет возможности сразу перейти к бронированию, так как данный функционал доступен только на странице вывода всех авиаперелетов в самом конце списка



1. ВЫБРАТЬ - АВИАРЕЙСЫ

1. Выбрать

2. Бронь

3. Оплатить

Алматы — ➔ Астана

Вс 28 Апр 1 





Цены указаны для всех пассажиров с учетом налогов.

ВЫБЕРИТЕ АВИАРЕЙС

[Перейти к календарю](#)

Алматы в Астана

АПРЕЛЬ - МАЙ 2019

ВС 21 нет доступных	ПН 22 от KZT 54,306	ВТ 23 от KZT 28,906	
---	---	---	---

Рисунок 1.2 – Страница вывода подходящих авиаперелетов в приложении Air Astana

✕ Сортировать по ▾

🔍 Отфильтруйте рейсы ▾

 Самые низкие тарифы

17:45

Международный аэропорт Алматы

↓ Прямой

19:35

Международный аэропорт Астана

Длительность 01:50

JSC Air Astana (KC995)

от KZT **54,306**  ▾

18:50

Международный аэропорт Алматы

↓ Прямой

20:30

Международный аэропорт Астана

Длительность 01:40

JSC Air Astana (KC991)

от KZT **54,306**  ▾

19:45

Международный аэропорт Алматы

↓ Прямой

21:30

Международный аэропорт Астана

Длительность 01:45

JSC Air Astana (KC855)

от KZT **54,306**  ▾

Чтобы продолжить, выберите обратный рейс

ДАЛЕЕ ▸

Рисунок 1.3 – Продолжение страницы вывода подходящих авиаперелетов в приложении Air Astana

1.3.2 Обзор мобильного приложения Aviata

Плюсы приложения Aviata:

1. Удобное и минималистичная стартовая страница поиска авиабилетов
2. Красиво анимированная страница загрузки
3. Удобный вывод списка подходящих авиаперелетов (рисунок 4)
4. Детали выбранного авиаперелета удобно читать и ошибиться в понимании почти невозможно (рисунок 5)
5. Поиск авиаперелета по множеству авиакомпаний
6. Возможность голосового поиска

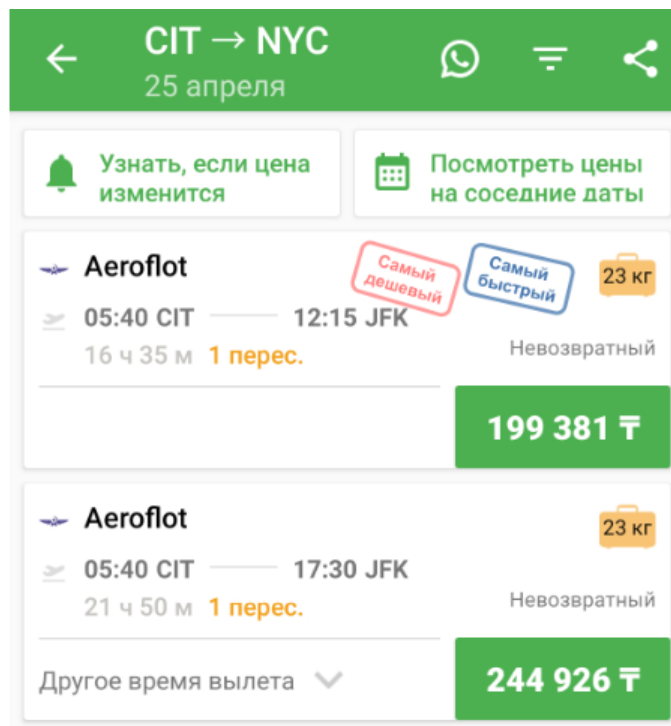


Рисунок 1.4 – Страница вывода подходящих авиаперелетов в приложении Aviata

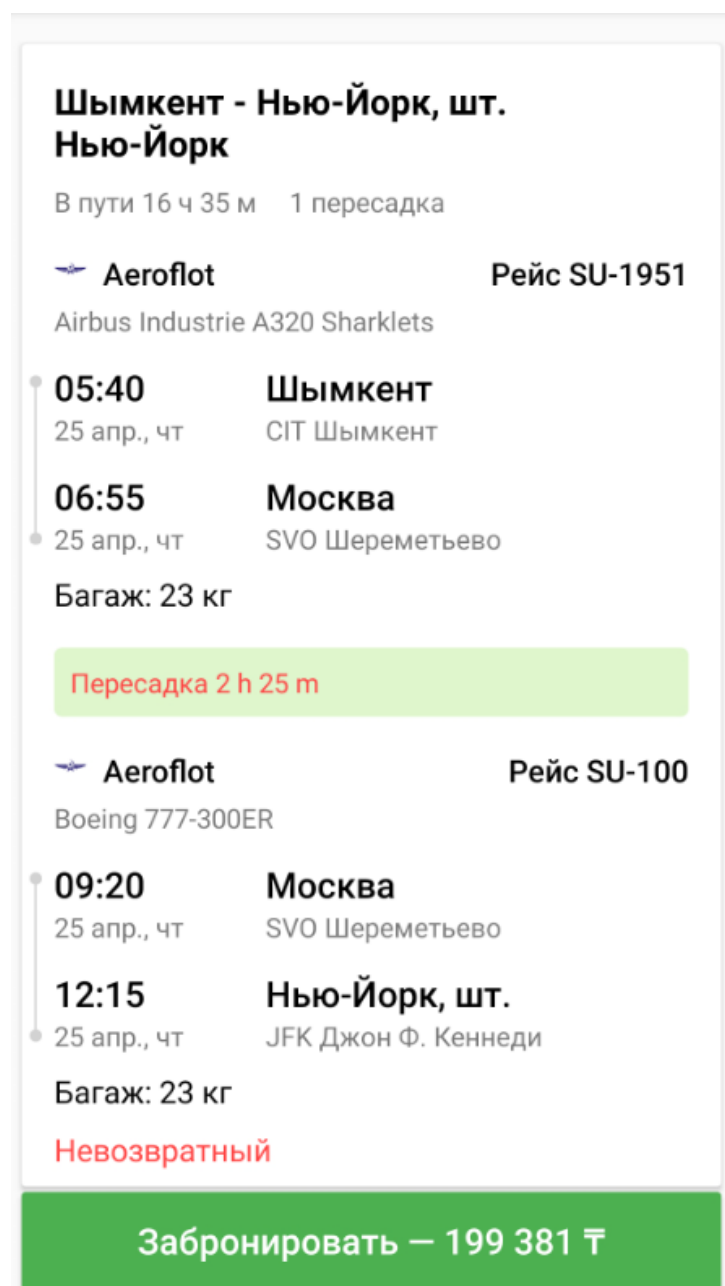


Рисунок 1.5 – Страница деталей выбранного авиаперелетов в приложении Avia.ru

Минусы приложения Avia.ru:

1. Возможности фильтра скудны (рисунок 7)
2. Не удобный ввод дат для документов
3. Личный кабинет в мобильном приложении не работает
4. У пользователей происходит частые ошибки с ценами перелета. В деталях перелета одна цена, при бронировании совсем другая
5. После последнего обновления приложения, отзывы пользователей на приложение стали резко негативными. Очень много появилось новых ошибок, мешающих пользоваться приложением (рисунок 8)

Прямые рейсы

Туда

Вылет

Прилет

УТРО

ДЕНЬ

ВЕЧЕР

НОЧЬ

Авиакомпании

AA

KLM ROYAL DUTCH AIRLINES

AEROFLOT

Рисунок 1.7 – Страница деталей выбранного авиаперелетов в приложении Aviata

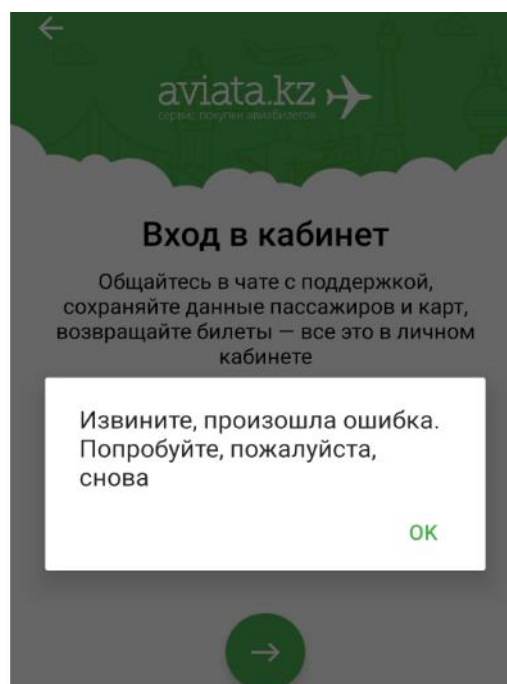


Рисунок 1.8 – Пример частых ошибок приложения

2 Технологический раздел

2.1 Обзор используемых инструментов

2.1.1 Андроид ОС

В самом начале разработки мобильного приложения было решено что, проект будет ориентирован на одну из самых распространенных и популярных операционных систем для мобильных устройств – Андроид. Данная операционная система имеет множество бесспорных преимуществ по сравнению с конкурентами в лице IOS, Simbian, BlackBerry и тд. Самые явные преимущества:

1. Андроид ОС является самой популярной в мире (рисунок 1)
2. Андроид – операционная система с открытым исходным кодом
3. Программное обеспечение для разработки мобильных приложений бесплатно и доступно на всех операционных системах

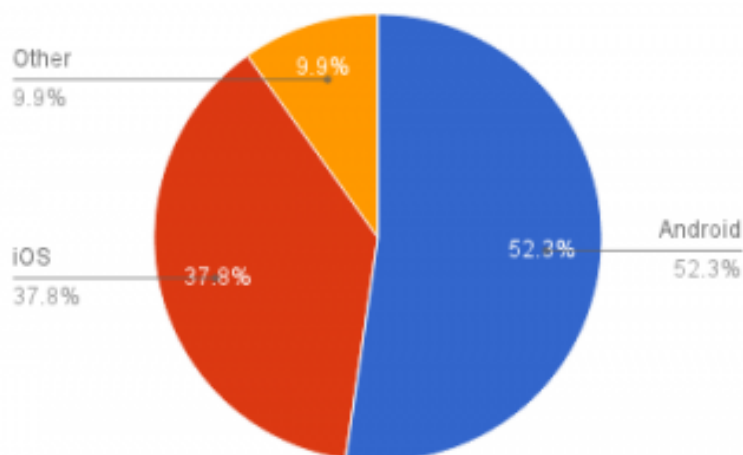


Рисунок 2.1 – Доля устройств на различных ОС для мобильных устройств

2.1.2 Android Studio

Одним из важнейших элементов разработки любого приложения является выбор интегрированной среды разработки – IDE. Для создания приложений на платформе Андроид подходят такие среды разработки как, Eclipse, NetBeans, Android studio. Для работы был выбран последний вариант, который находится в свободном доступе с 2013 года (рисунок 2). Android studio является универсальным IDE от компании Google, так как позволяет создавать и протестировать приложения, которые будут работать на смартфонах, планшетах, портативных ПК, часах, которые работают на основе Андроид ОС. Данная среда

разработки является по настоящему удобной по нескольким причинам:

1. Удобный редактор кода, который помогает в написании кода и предлагает завершение, рефракцию и анализ кода
2. Наличие большой библиотеки готовых шаблонов и компонентов для разработки ПО
3. Возможность предварительной проверки ПО на ошибки
4. Встроен эмулятор Android (рисунок 3)

Эмулятор Android - это виртуальное устройство, которое может представлять любое устройство, работающее на платформе Android. Используется эмулятор Android в качестве целевой платформы для запуска и тестирования мобильных приложений. Эмулятор обладает удобной отличительной особенностью – просмотр показателей производительности при запуске приложения (рисунок 4).

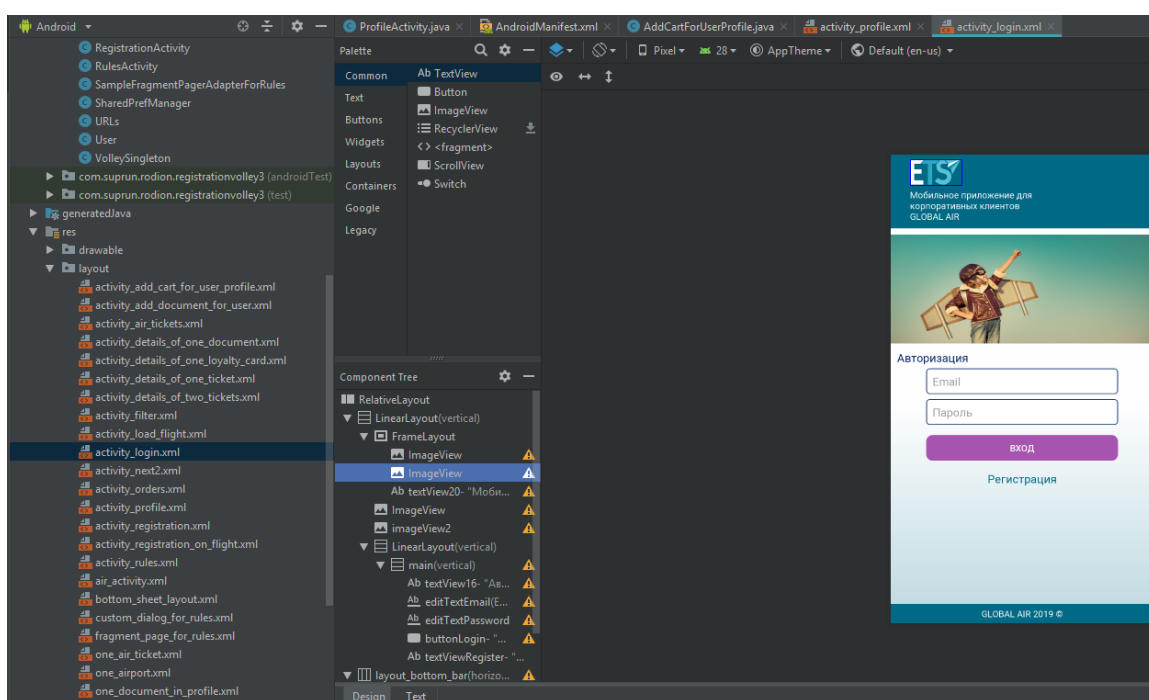


Рисунок 2.2 – Android studio



Рисунок 2.3 – Эмулятор в Android studio

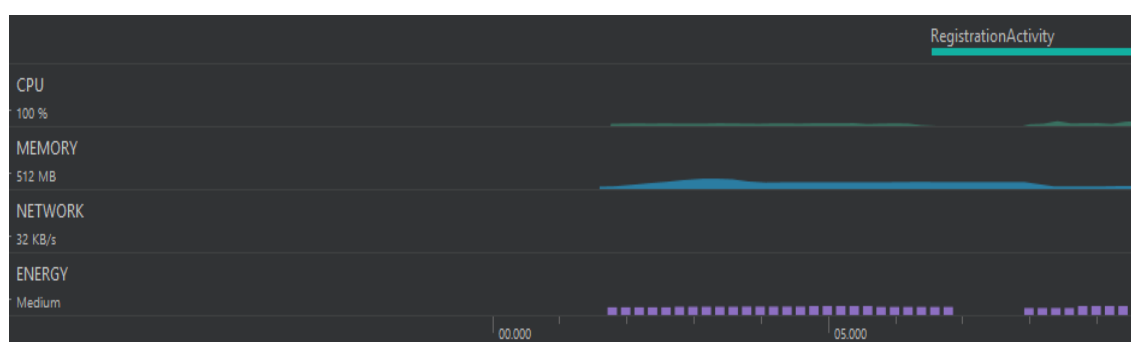


Рисунок 2.4 – Показатели производительности в Android studio

2.1.3 Язык программирования Java

Разработка мобильного приложения происходило на официальном языке программирования, поддерживаемый средой разработки Android Studio - Java. Язык Джава был выбран по многим причинам:

1. Он является одним из самых распространенных и популярных языков программирования в мире
2. Java — строго типизированный объектно-ориентированный язык программирования, обладающий простотой и высокой производительностью
3. Безопасность. Джава, наверное, самый безопасный язык программирования на сегодняшний день

2.1.4 Библиотека Volley

В приложении не используется локальная база данных, так как нет необходимости хранить данных на самом устройстве. Вместо этого приложение берет все данные с БД сервера посредством HTTP протокола передачи данных и реализации API интерфейса на веб-сервере. Данное взаимодействие называется клиент - серверной архитектурой (рисунок 5).

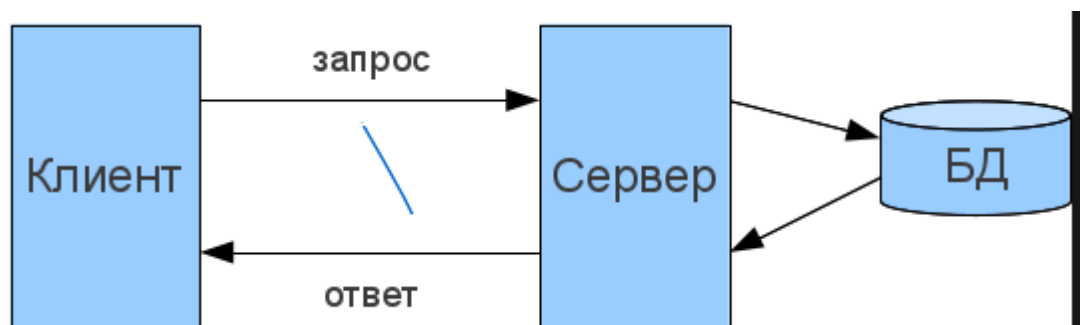


Рисунок 2.5 – Сетевая архитектура "клиент - сервер"

Сервер постоянно ждет запросы от клиента, и готов в любое время ответить на post или get запрос. Для отправки запросов на сервер в Android используются различные библиотеки такие как Retoroft, OkHttp, Volley.

Для работы была выбрана библиотека Volley, так как обладает множеством преимуществ, и самое главное библиотека легка в понимании и использовании. Обычно в Android код сетевого вопроса помещался AsyncTask. Надо было позаботиться об отображении ProgressBar внутри функций OnsourceExecute() и OnPostExecute(). Благодаря Volley больше о такой рутине думать разработчикам

нет необходимости. Volley предлагает множество функций, как указано в его документации:

1. Автоматическое планирование сетевых запросов
2. Несколько одновременных сетевых подключений
3. Volley предоставляет инструменты отладки
4. Поддержка приоритизации запросов
5. Простота настройки, например, для повторных попыток и откатов

Для отправки запросов с помощью библиотеки Volley используется 2 основных класса: Request и Request queue. Класс Request содержит необходимые детали для создания запроса такие как, как выбор типа запроса, URL, данные запроса. В своем приложении я использовал StringRequest, который ответ от сервера преобразует в тип данных – строка (рисунок 6). Затем работаем с ответом, вытаскиваем Json – объекты, массивы и тд.

Класс Request queue используется для отправки запросов, в прямом смысле каждый запрос мы помещаем в очередь.

```
StringRequest stringRequest = new StringRequest(
    Request.Method.GET,
    url,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            // Do something with response string
            mTextView.setText(response);
            requestQueue.stop();
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            // Do something when get error
            Snackbar.make(mCLayout, "Error.", Snackbar.LENGTH_LONG).show();
            requestQueue.stop();
        }
    }
);

// Add StringRequest to the RequestQueue
requestQueue.add(stringRequest);
```

Рисунок 2.6 – Синтаксис запроса с помощью библиотеки Volley

2.1.5 Firebase Cloud Messaging

Для уведомления пользователей о скидках или скором авиаперелете использовалась библиотека Firebase, недавно купленная компанией Google (рисунок 7). Аналогом раньше была библиотека GSM, но компания Google перестала ее поддерживать в апреле этого года. На стороне приложения необходимо реализовать сервисы, которые будут ждать уведомления от сервера. А через консоль Firebase можно отправлять уведомления либо всем пользователям, либо только одному.

The screenshot displays the Firebase console interface for creating a notification. On the left is a dark sidebar with the 'Firebase' logo and a 'Test Lab' button. Below these are two main sections: 'Analytics' and 'Grow'. The 'Analytics' section includes links to Dashboard, Events, Conversions, Audiences, Funnels, User Properties, Latest Release, Retention, StreamView, and DebugView. The 'Grow' section includes links to Predictions, A/B Testing, Cloud Messaging (which is highlighted in blue), In-App Messaging, Remote Config, Dynamic Links, and AdMob. The main content area has a blue header with 'ETS ver 1' and a dropdown arrow, followed by 'Notifications' and a back arrow with the text 'Создание уведомления'. The notification creation form is divided into four steps: 1. 'Уведомление' (Notification), where the title 'Хей' and text 'Привет' are entered; 2. 'Аудитория' (Audience); 3. 'Планирование' (Scheduling), with an option to 'Отправить сейчас' (Send now); and 4. 'События-конверсии (необязательно)' (Events-conversions (optional)). A 'Send test message' button is located between the first and second steps.

Рисунок 2.7 – Консоль для отправления push уведомлений Firebase

3 Проектный раздел

3.1 Архитектура системы

Разрабатываемый дипломный проект является мобильным приложением системы бронирования ETS, а значит постоянно взаимодействует с сервером с помощью POST и GET запросов. Для этой цели на сервере был реализован API интерфейс для взаимодействия мобильного приложения с базой данных (рисунок 1). Для отправки уведомлений пользователям, приложение подключили к Firebase, который является поставщиком облачных услуг. Схема работы системы при отправлении уведомления представлена на рисунке 2.

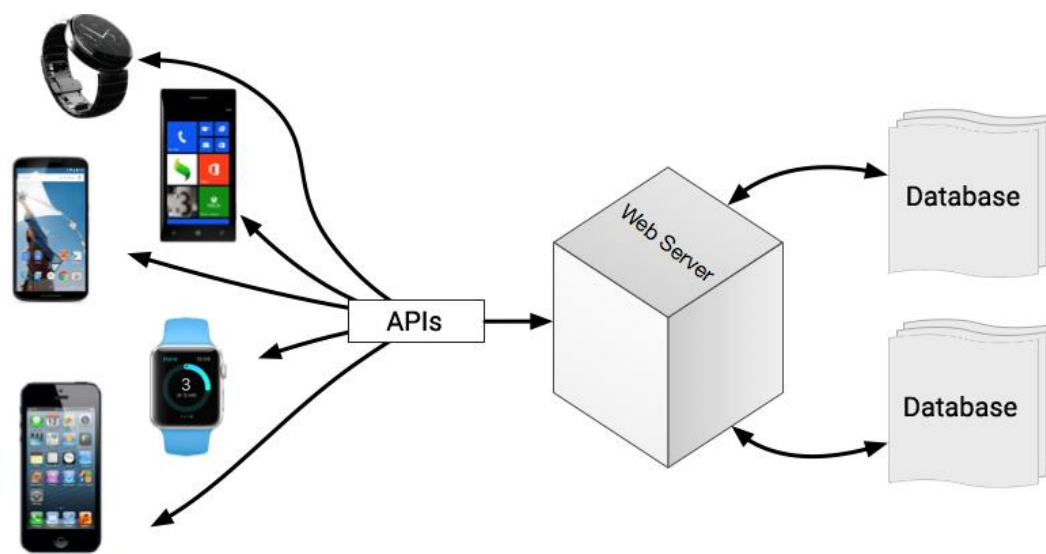


Рисунок 3.1 Архитектура системы

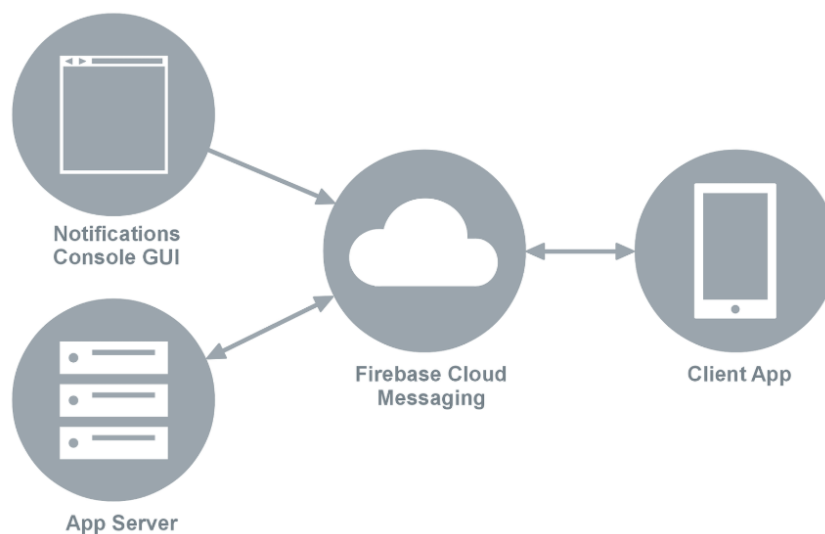


Рисунок 3.2 Архитектура системы для отправления уведомлений

3.2 Архитектура мобильного приложения

Архитектура почти любых Android – приложений состоит из 3 слоев: Domain Layer, Presentation Layer, Data Layer (рисунок 3).

Слой Domain Layer хранит всю бизнес-логику. Используется для различных манипуляций с базой данных – CRUD операций.

Слой Presentation Layer является уровнем представления, уровень, с которым взаимодействует пользователь. Presentation Layer включает в себя пользовательский интерфейс.

Слой Data Layer является источником данных. Для получения данных из этого уровня, слой Presentation обращается к Domain Layer для получения данных.

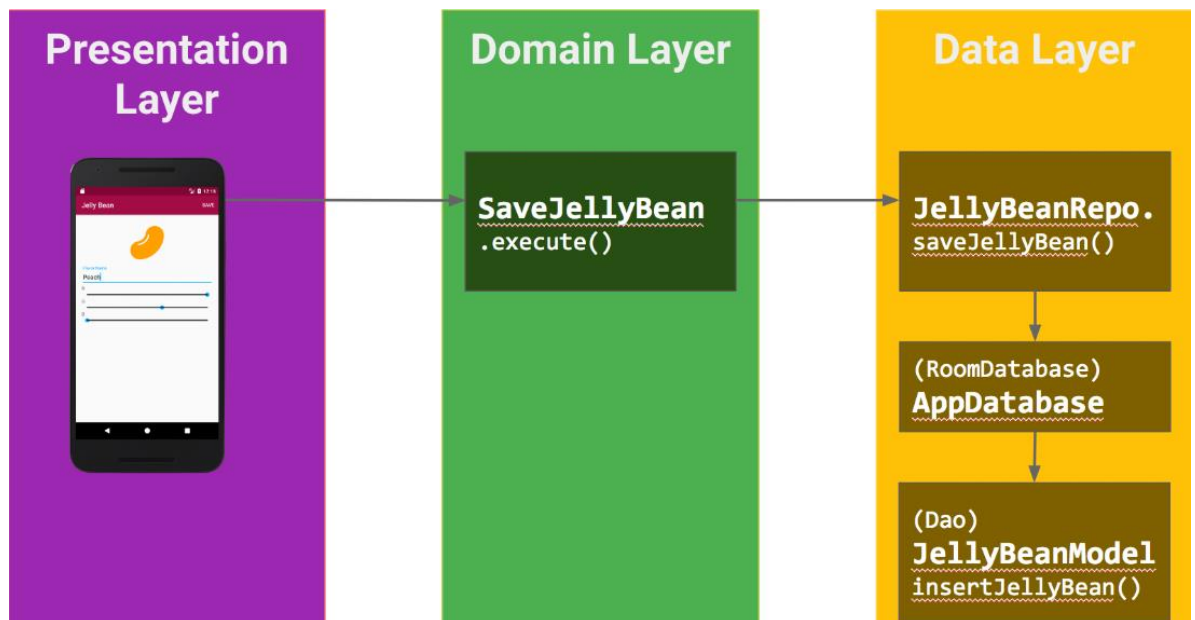


Рисунок 3.2 Архитектура мобильного приложения

3.3 Проектирование мобильного приложения с использованием UML диаграмм

Моделирование является неотъемлемой частью больших программных проектов, а также полезно для средних и небольших проектов. Моделирование играет аналогичную роль в разработке программного обеспечения, которую чертежи и другие планы играют в строительстве небоскреба. Используя модель, лица, ответственные за успех проекта разработки программного обеспечения, могут убедиться, что бизнес-функциональность является полной и правильной, потребности конечного пользователя удовлетворены до внедрения.

3.3.1 Проектирование с помощью диаграммы вариантов использования

Диаграмма вариантов использования - это диаграмма поведения в UML. Диаграмма моделирует функциональность системы с использованием действующих лиц и прецедентов (рисунок 4). Варианты использования - это набор действий, сервисов и функций, которые должна выполнять система. В этом контексте «система» - это то, что разрабатывается или эксплуатируется, например, мобильное приложение. «Действующие лица» - это люди или организации, действующие под определенными ролями в системе.

Так как это мобильное приложение то актер будет только один – пользователь. Обязательным условием для использования системы ETS является обязательная авторизация пользователя. И после этого для использования будут доступны все функции системы бронирования.

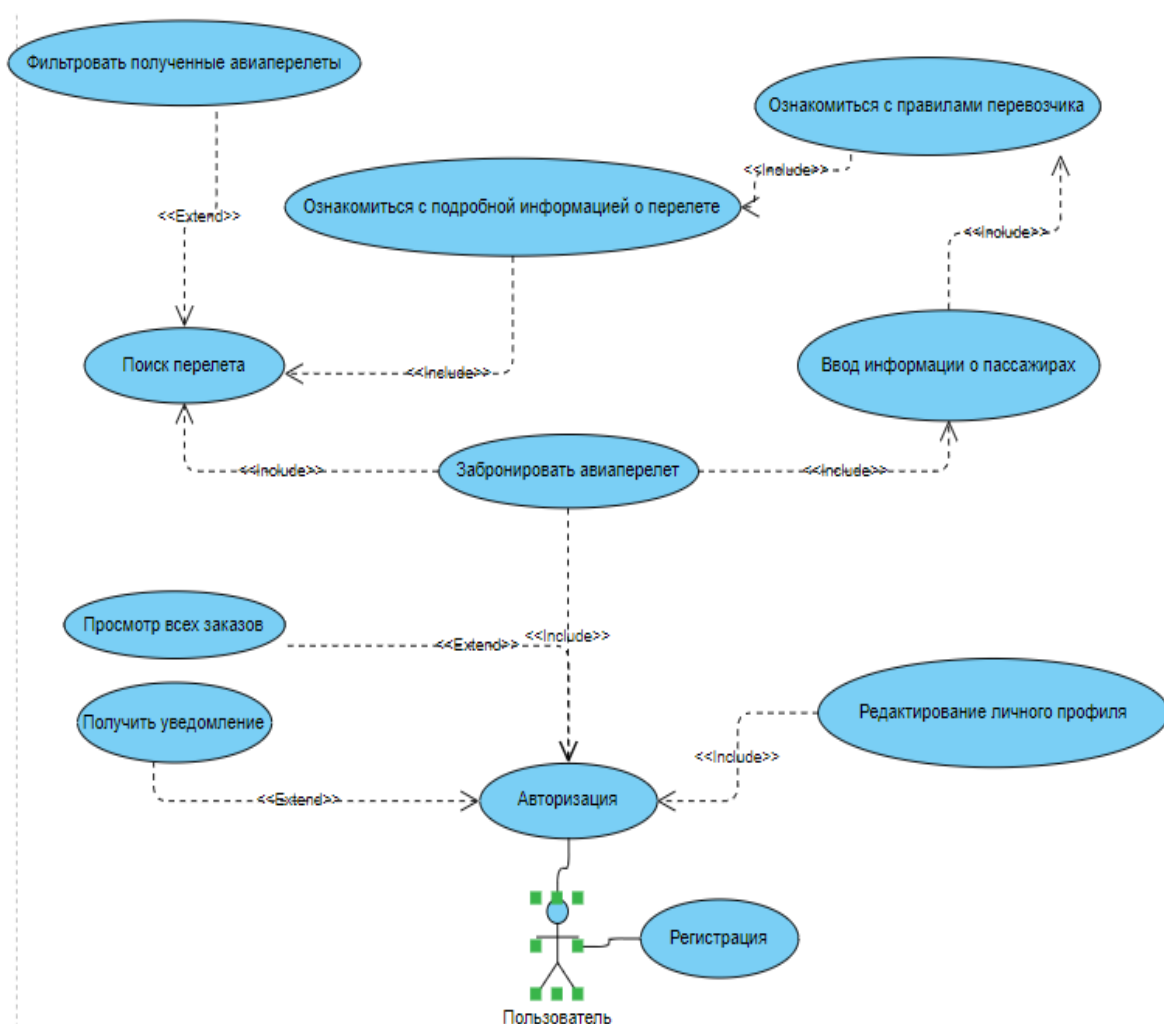


Рисунок 3.4 Диаграмма вариантов использования

3.3.2 Проектирование с помощью диаграммы деятельности

Основная цель диаграммы деятельности аналогична четырем другим диаграммам. Она фиксирует динамическое поведение системы. Диаграмма действий изображает поток управления от начальной до конечной точки, показывая различные пути принятия решений, которые существуют во время выполнения действия (рисунок 5).

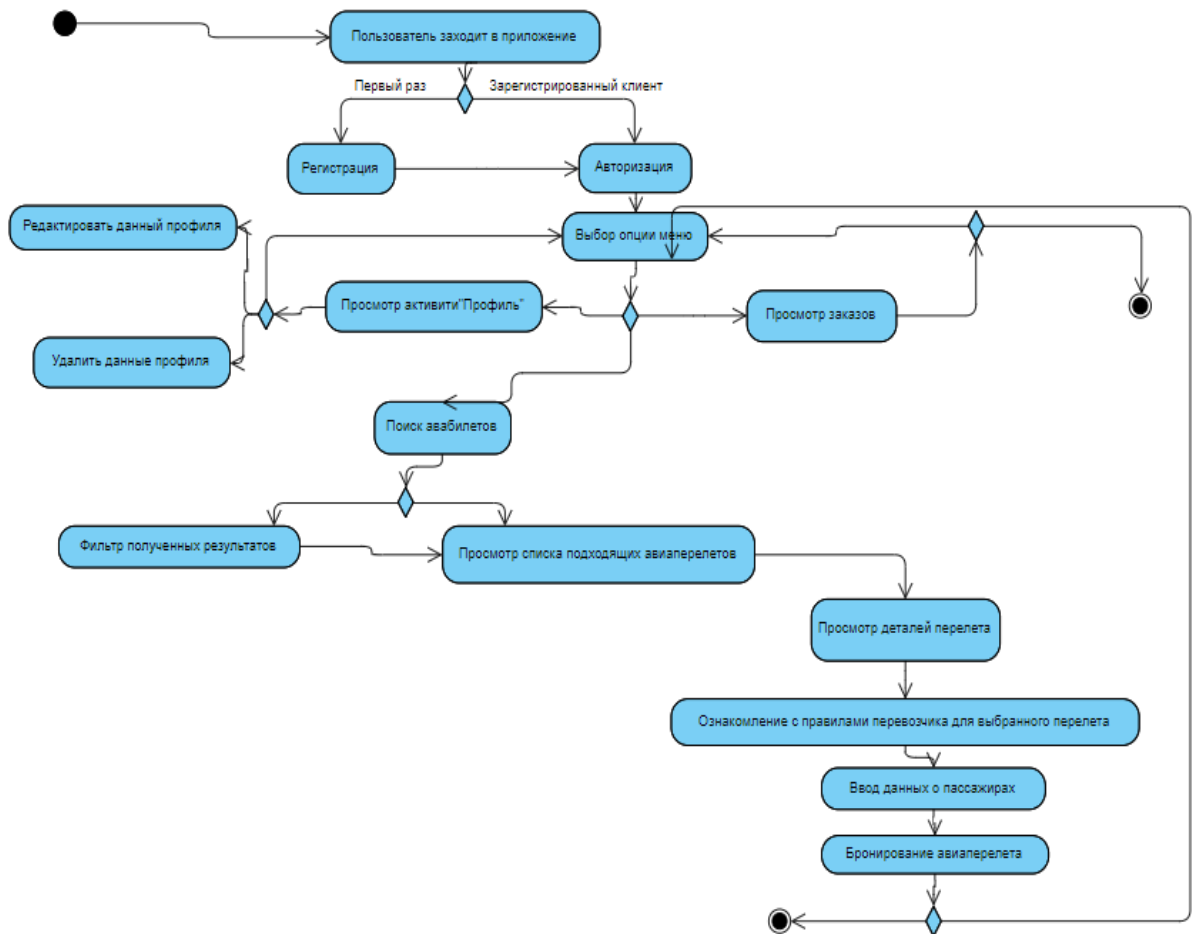


Рисунок 3.5 Диаграмма деятельности

3.4 Проектирование интерфейса мобильного приложения

Интерфейс мобильного приложения – рычаг взаимодействия между пользователем и машинным кодом. Дизайн пользовательского интерфейса - это ремесло, которое включает создание важной части пользовательского опыта; пользователи очень быстро оценивают дизайн по удобству и простоте использования. Пользователи знакомы с элементами интерфейса,

действующими определенным образом, поэтому нужно быть последовательными и предсказуемыми в расположении элементов. Дизайн приложения разрабатывался как в самом Android studio, так и в программе CorelDRAW (рисунок 6, 7, 8).

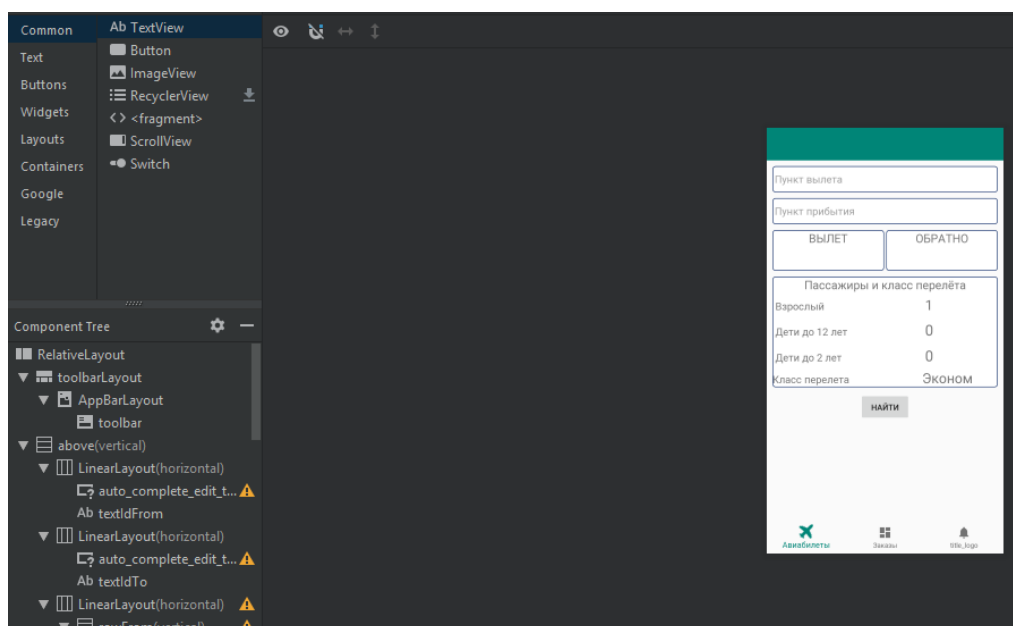


Рисунок 3.6 Расположение элементов пользовательского интерфейса в Android studio

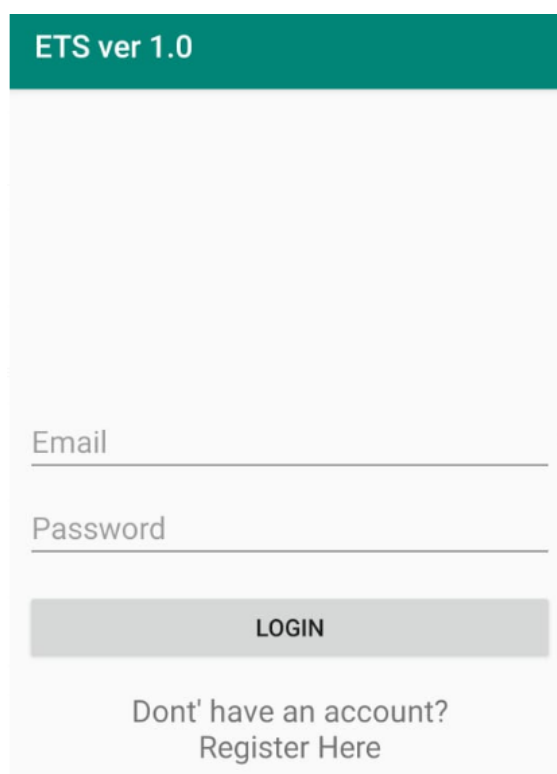


Рисунок 3.7 Первоначальный интерфейс страницы авторизации

YOTA 15:25 85 %

Поиск авиабилетов

Откуда
выберите город

Куда
выберите город

Дата вылета
28 апреля

Обратный рейс
30 апреля

В одну сторону ☒

Пассажир 1

Класс
Эконом ▼

Возраст
Взрослый (старше 14) ▼

Добавить пассажира

Найти билеты

Рисунок 3.8 Созданный прототип страницы приложения в CorelDRAW

4 Экспериментальный раздел

4.1 Функции мобильного приложения ETS

В самом начале создания дипломного проекта компанией Global air был определен функционал мобильного приложения, необходимый для предоставления авиа услуг.

Необходимые функции приложения:

1. Регистрация и авторизация в системе ETS через приложение
2. Поиск и бронирование авиабилетов
3. Большие возможности фильтрации авиаперелетов
4. Проверка и отслеживание статуса заказа
5. Просмотр истории всех заказов пользователя
6. Уведомление пользователей
7. Редактирование личного профиля

О реализации каждой функции и пойдет речь в следующих пунктах.

4.2 Создание мобильного приложения ETS

Любое Android – приложение состоит из нескольких активностей, которые иногда связаны между друг другом. Активити - это один экран пользовательского интерфейса приложения, определенная комбинация XML-файла и Java-файла. По сути, Активити - это контейнер, который содержит как дизайн, так и кодирование. XML-файл представляет нам дизайн нашего экрана. Файл Java содержит код страницы, который отвечает за всё происходящее в активити.

Обычно одна из активностей в приложении обозначается как «основная», предлагаемая пользователю при первом запуске приложения; в моем дипломном проекте такой активностью является авторизация пользователей, так как большинство клиентов уже зарегистрированы в системе ETS.

4.2.1 Создание страницы авторизации и регистрации

Активити авторизации пользователя является стартовой страницей приложения (рисунок 1). Данная страница состоит из двух полей - EditText, необходимые для ввода почтового адреса и пароля соответственно. Кнопка “ВХОД” запускает функцию, которая выполняет множество задач. В начале функции считываются введенные данные, которые сравниваются с требованиям валидации. Например, ни одно из полей не может быть пустым или недостаточно

длинным, а правильность введенного email проверяется с помощью встроенной библиотеки в Android studio. Если введенные данные не прошли проверку, пользователь получит уведомление об ошибках. Если данные прошли валидацию, формируется POST запрос с помощью библиотеки Volley, параметрами которого являются email и пароль пользователя. В случае положительного ответа от системы, пользователь переносится на страницу поиска авиаперелетов, в противном случае получается сообщение о том, что введенные данные не верны. Так же страница содержит две надписи: “Регистрация”, “GLOBAL AIR 2019”. По нажатию на “Регистрация” пользователь перемещается на страницу регистрации в системе (рисунок 2). Сайт компании открывается при нажатии надписи “GLOBAL AIR”.

Активити Регистрация состоит из 3 полей для ввода данных, почтовый адрес, пароль и повторный пароль. Требования валидации такие же, как и на странице авторизации, только добавляется правило, что пароль и повторный пароль должны совпадать. Надписи “Авторизация” и “GLOBAL AIR” выполняют такие же функции, как и на активити авторизации

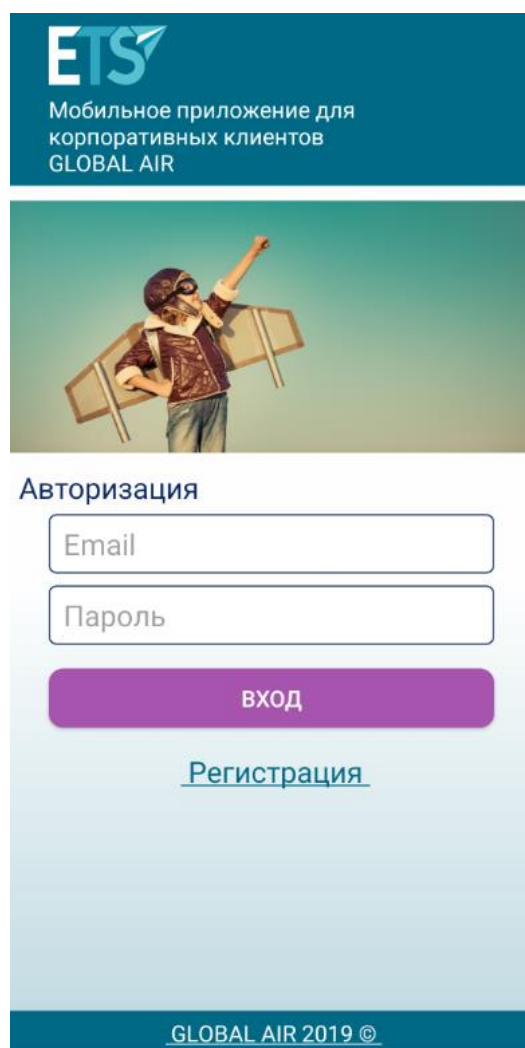


Рисунок 4.1 – Страница авторизации в системе ETS

Регистрация

Email

Пароль

Ввести пароль повторно

РЕГИСТРАЦИЯ

[Авторизация](#)

GLOBAL AIR 2019 ©

Рисунок 4.2 – Страница регистрации в системе ETS

4.2.2 Создание страницы поиска авиаперелетов

Активити поиска авиаперелета состоит из двух полей выбора города – `AppCompatAutoCompleteTextView`, при заполнении пользователем первых букв города, отправляется `Get` запрос на получение подходящих городов, затем пользователь должен выбрать подходящий (рисунок 3 и 4).

Дата вылета и дата обратного рейса реализованы с помощью `DatePickerDialog`. При нажатии на поле ввода появляется календарь для выбора даты вылета. На дату вылета наложены ограничения: дата не может быть выбрана раньше сегодняшнего дня и позже года от этой даты. При выборе даты вылета, на дату обратного рейса накладывается новое ограничение – дата обратного рейса не может быть раньше даты вылета. И если выбрать новую дату вылета, которая будет позже даты обратного рейса, последняя будет удалена и ограничена новой датой, так как не соответствует логике.

Выбор количества пассажиров и их тип происходит через вызова диалога – BottomSheetDialog, имеющий весь необходимый функционал и свои правила, такие как количество пассажиров в сумме не может быть больше 9 (рисунок 5).

Поиск производится путем отправления данных о пунктах вылета и прилета, о датах вылета и возможной даты обратного рейса, а также данные о количестве и типе пассажиров, классе обслуживания. Все эти данные отправляются в Get запросе параметрами по нажатию на кнопку “Найти билеты”, после этого проверяется наличие необходимых данных. Если все данные введены правильно строка запроса передается на страницу загрузки.

Рисунок 4.3 – Страница поиска авиабилетов в системе ETS

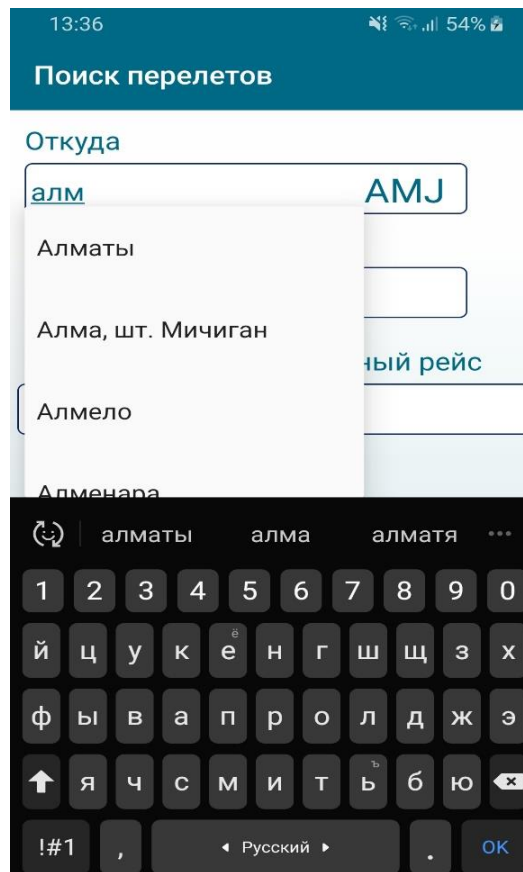


Рисунок 4.4 – Выбор города отправления

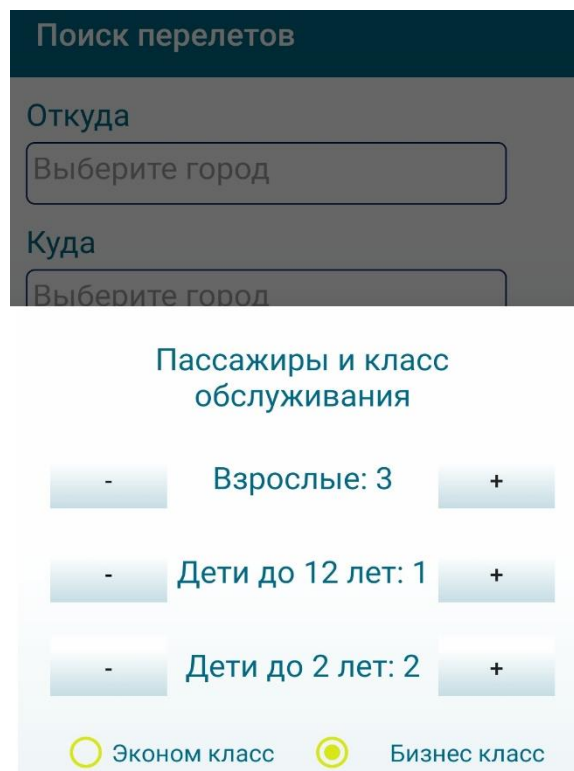


Рисунок 4.5 – Страница поиска авиабилетов в системе ETS

4.2.3 Создание страницы загрузки авиаперелетов

На данной странице происходит запрос с помощью полученной строки данных из прошлой активити. Пока идет ожидание ответа от сервера происходит анимация загрузки, чтобы пользователь не думал, что приложение зависло. После получения ответа в формате строки извлекаются массивы, и объекты данных, такие как массив фильтров, массив городов и тд. Группируются в объекты и передаются на активити вывода списка авиаперелетов (рисунок 6).

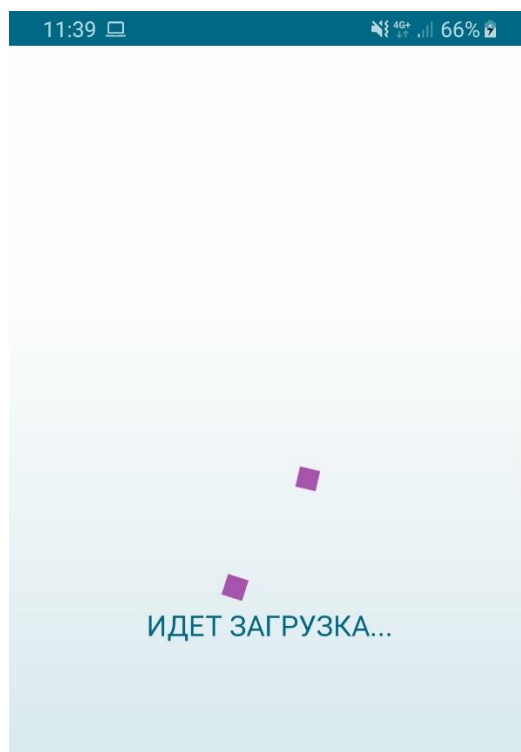


Рисунок 4.6 – Страница поиска авиабилетов в системе ETS

4.2.4 Создание страницы для вывода списка авиаперелетов

На данной активити полученный массив перелетов выводится с помощью ListView и ListAdapter в удобном для пользователя виде (рисунок 7). При нажатии на любой авиаперелете пользователь перенаправляется на страницу деталей авиаперелета.

Надпись “Список авиаперелетов” находится вверху экрана в элементе под названием Toolbar, аналог header в веб программировании. Так же в Toolbar находится стрелка для перехода на страницу поиска авиаперелетов, и кнопка для перехода на страницу фильтра, которой передаем массив возможных фильтров.

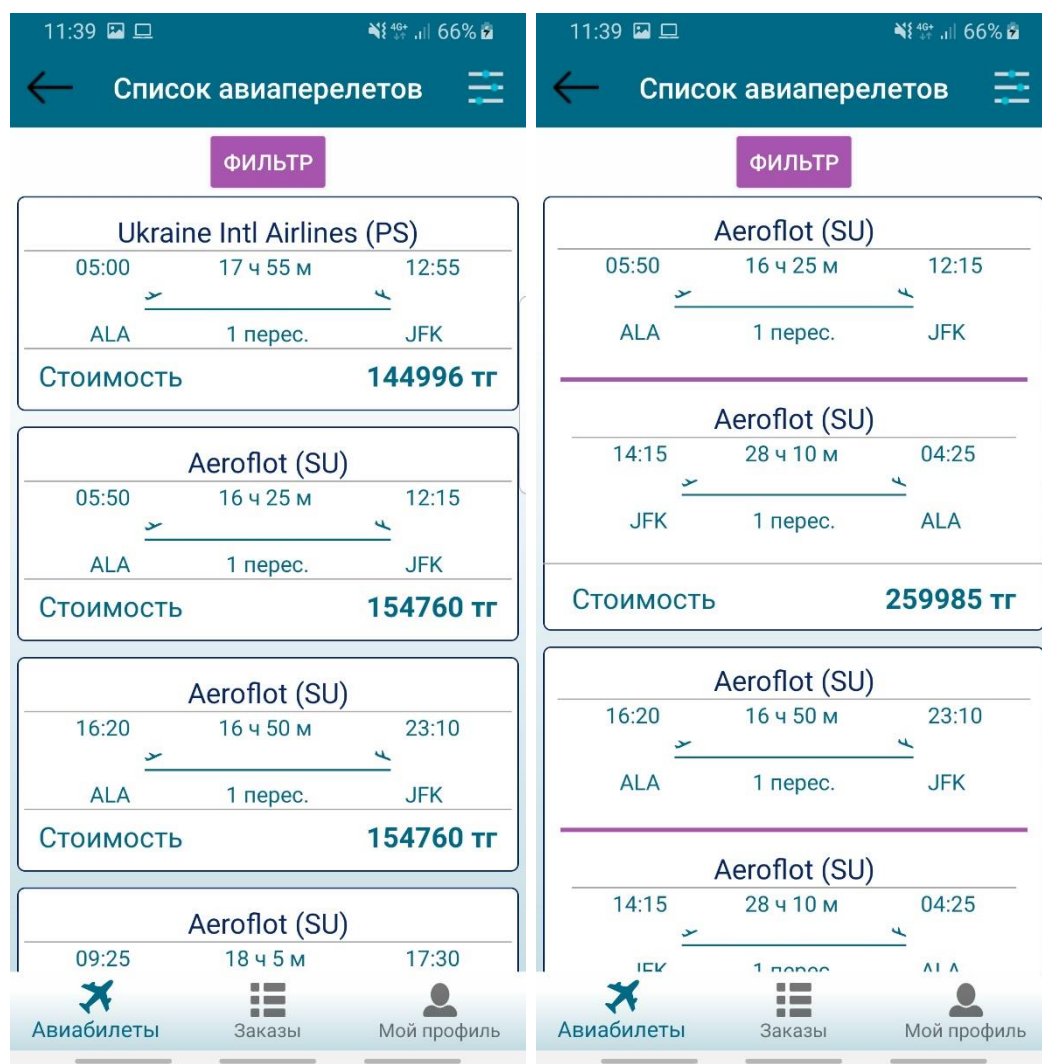


Рисунок 4.7 – Страница “Список авиаперелетов”

4.2.5 Создание страницы для фильтрации авиаперелетов

Данное активити предназначено для фильтрации авиаперелетов. Возможности фильтра широки: фильтрация по цене, фильтрация по времени перелета, фильтрация аэропортов вылета и аэропортов пункта назначения, фильтрация по всем авиакомпаниям совершающие подходящие авиаперелеты, а также фильтрация по количеству пересадок (рисунок 8).

11:42 4G+ 66%

ГОТОВО

Стоимость от 144996 тг до 3089471 тг

144996 3089471

Min Max

Время в пути от 16 ч до 45 ч

16 44

Min Max

Аэропорты вылета

Almaty ☒

Аэропорты прибытия

New York, John F Kennedy Intl ☒

New York, Newark Liberty Intl ☒

New York, Laguardia ☒

Пересадки

1 перес. ☒

2 перес. ☒

Авиакомпании

Ukraine Intl Airlines (PS) ☒

Aeroflot (SU) ☒

Uzbekistan Airways (HY) ☒

Emirates (EK) ☒

KLM Royal Dutch Airlines (KL) ☒

Singapore Airlines (SQ) ☒

Turkish Airlines (TK) ☒

United Airlines (UA) ☒

Lufthansa (LH) ☒

Asiana Airlines (OZ) ☒

Austrian (OS) ☒

Рисунок 4.7 – Фильтр авиаперелетов Алматы – Нью-Йорк

4.2.6 Создание активити “Детали авиаперелета”

При запросе авиаперелетов приходит полная информация по каждому авиаперелету, которые я помещаю в массив. Благодаря этому на сервер нет необходимости отправлять запрос для деталей перелета, данные берутся из массива. Функциональными являются 2 кнопки: кнопка перехода назад в ToolBar и кнопка забронировать с ценой авиаперелета.



Рисунок 4.8 – Детали перелета Алматы - Астана

4.2.7 Создание активности “Правила обмена и возврата”

На данной странице происходит запрос на сервер для получения правил перевозчика на выбранный перелет, параметром запроса является id авиаперелета (рисунок 9).

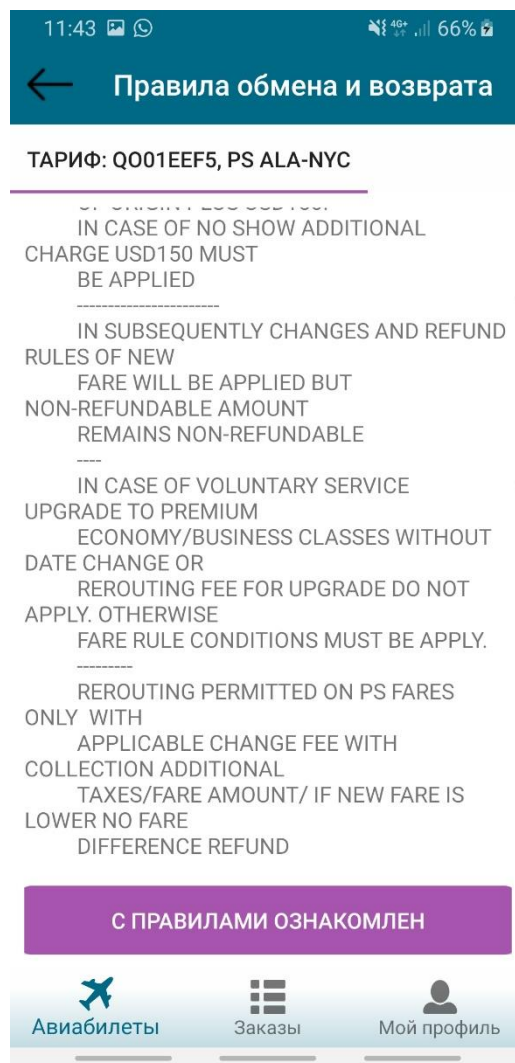


Рисунок 4.9 – Правила обмена и возврата авиаперелета Алматы – Нью-Йорк

4.2.8 Создание активности для ввода данных пассажиров

На данной странице происходит заполнение всей информации о пассажирах: ФИО, дата рождения, гражданство, ИИН, тип документа, номер документа, срок действия документа (рисунок 10). Для вывода всех полей для ввода информации любого количества пассажиров был написан свой адаптер, который работает с полями класса пассажир. Гражданство выбирается из выпадающего списка возможных стран, запрос массива стран произошел заранее на странице деталей перелета и передается от одной активности другой. Дата рождения выбирается с помощью `DataPickerDialog`. В зависимости от того пассажир - ребенок, младенец или взрослый, на выбор даты рождения наложены ограничения. Ребенок не может быть старше 12 лет, а взрослый не может быть младше 12 лет и тд.

Рисунок 4.10 – Ввод данных пассажиров

4.2.9 Создание активити для вывода списка заказов

Данное активити предназначено для вывода всех заказов текущего пользователя. В методе onCreate() вызывается функция для GET запроса на сервер. Входными параметров у запроса нет, необходимо только чтоб пользователь был авторизован.

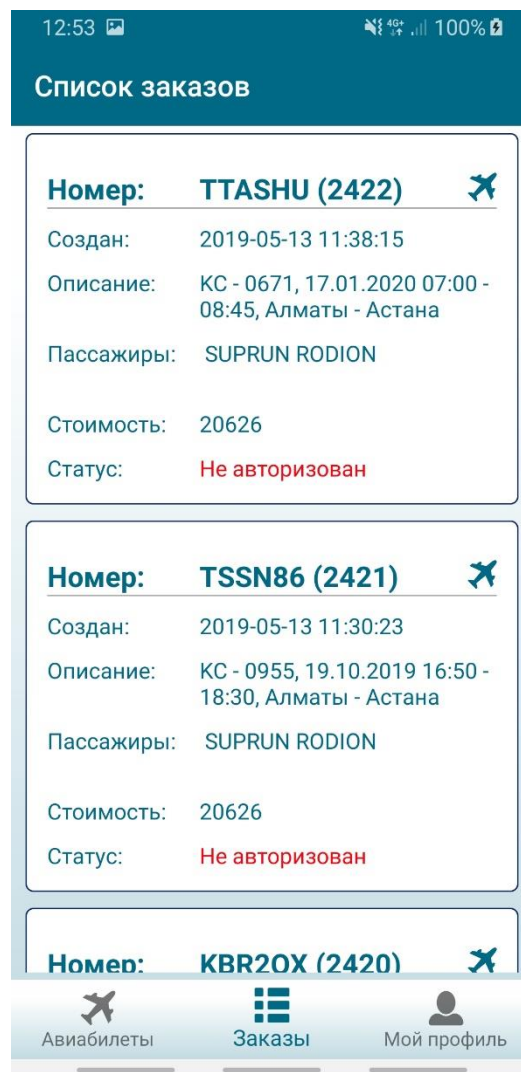


Рисунок 4.11 – Вывод списка заказов

4.2.10 Создание активности “Мой профиль”

Благодаря данной активности пользователь может редактировать, добавлять и удалять различные данные своего профиля, такие как пол, возраст, ФИО на английском и русском языке, название компании, номера телефонов и тд (рисунок 12).

Так же на данной странице выводится список документов: паспорта и удостоверения личности. При нажатии на документ пользователь переходит на страницу редактирования выбранного документа. Документ может быть, как отредактирован, так и полностью удален (рисунок 12 и 14).

Отличительной особенностью системы ETS является так же возможность добавления карт лояльностей авиаперевозчиков (рисунок 12 и 13).

14:57 4G+ 76%

Фамилия
SUPRUN

Имя
ММК!!GFGGGGGGGGGGGGGGGGGGGG

Гражданство
Португалия

Вид документа
Удостоверение личности

Серия и номер документа
1515

Дата выдачи
15-04-2019

Годен до
27-04-2019

УДАЛИТЬ ДОКУМЕНТ СОХРАНИТЬ ИЗМЕНЕНИЯ

Рисунок 4.14 – Редактирование документа пользователя

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы было создано мобильное приложение на платформе Android, на основе системы бронирования ETS. Программный продукт ориентирован на корпоративных клиентов компании Global air, системой не могут пользоваться физические лица, и клиенты не прошедшие первоначальную регистрацию в системе. Так же был проведен анализ приложений, имеющих в Google play, для создания приложения, не имеющее недостатков конкурентов в данной области.

Функции созданного приложения:

1. Регистрация и авторизация пользователя
2. Поиск авиаперелетов
3. Фильтрация авиаперелетов
4. Детальный обзор выбранного авиаперелета
5. Ознакомление пользователя с правилами перевозчика
6. Бронирование авиаперелета на выбранное количество пассажиров
7. Вывод списка заказов пользователя
8. Редактирование личного профиля

Созданный дипломный проект является законченным программным продуктом в области бронирования авиабилетов, который будет в последствии развиваться и улучшаться. Приложение было протестировано не только на стандартных эмуляторах, встроенных в Android studio, но и на реальных устройствах на платформе Android. Следующим шагом в развитии будет добавление функционала поиска и бронирования железнодорожных билетов. Приложение не является кроссплатформенным, ориентированно только на платформу Android.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. П. Дейтел, Х. Дейтел, Э. Дейтел. Андроид для программистов. – СПб.: Питер, 2013. – 608 с.
2. Дон Фелкер. Android: разработка приложений для чайников. Пер. с англ. — М.: ООО “И.Д. Вильямс”, 2012. – 336 с.
3. Коматинени Сатия, Маклин Дэйв. Android 4 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов. Пер. с англ. — М.: ООО “И.Д. Вильямс”, 2012. — 880 с.
4. Медникс З., Дорнин Л., Мик Б. Программирование под Android. 2-е изд. — СПб.: Питер, 2013. — 560 с.
5. Освой программирование играючи // Электронная версия на сайте <http://developer.alexanderklimov.ru/android/>.
6. Start Android - учебник по Android для начинающих и продвинутых // Электронная версия на сайте <https://startandroid.ru/ru/>.
7. METANIT - сайт о программировании. Программирование под ОС Андроид // Электронная версия на сайте <https://metanit.com/java/android/>.

Приложение А

(обязательное)

Техническое задание

А.1.1 Техническое задание на разработку мобильного приложения для поиска и бронирования авиабилетов на платформе Android

Настоящее техническое задание распространяется на разработку мобильного приложения на платформе Android, предназначенного для поиска и бронирования авиабилетов. Использовать данное приложение будут корпоративные клиенты туристической компании Global air. Мобильное приложение позволит выйти компании со своей системой бронирования ETS на мобильный рынок, что в свою очередь увеличит распространенность системы.

А.1.1.1 Назначение

Мобильное приложение предназначено для online поиска и бронирования авиабилетов для корпоративных клиентов компании Global air.

А.1.1.2 Требования к внешнему виду приложения

Приложение должно обладать современным дизайном, а также продуманным и удобным пользовательским интерфейсом. Так же должна соблюдаться преемственность приложения с сайтом компании.

А.1.1.3 Требования к функциональным характеристикам

Приложение должно обладать следующим функционалом:

- работа приложения должна быть завязана на взаимосвязи с сервером системы;
- регистрация и авторизация пользователей в системе ETS;
- ввод первоначальных данных пользователем для поиска перелетов по всем правилам Amdeus
- отправка и расшифровка запросов на сервер для поиска перелетов;
 - отправка и расшифровка запросов на сервер для реализации

Продолжение приложения А

- дополнительной фильтрации, используя все возможности системы;
- возможность детального обзора информации о выбранном пользователем перелете.
- Возможность ознакомления пользователя с правилами перевозчика на выбранном перелете
- Отправка проверенных данных о пассажирах для бронирования авиаперелета
- Возможность обзора списка заказов пользователя
- Возможность редактирования данных профиля. Добавление, удаления, редактирование документов, содержащих паспортные данные пользователя. Редактирование, удаление и добавление новых карт лояльностей. Возможность добавление и редактирования пользователем его особых требований, таких как выбор питания или выбор места.

Вводимые данные:

- На странице регистрации и авторизации вводимыми данными являются email и пароль пользователя;
- На странице поиска авиаперелетов, вводимые параметры как выбор городов отправления и прибытия должны формироваться с помощью AutoCompleteText, выбор даты отправления с помощью диалога календаря, выбор типа пассажиров и их количества с помощью всплывающего меню;
- На странице фильтрации минимальные и максимальные значения стоимости перелета и его длительности являются вводимыми параметрами и должны быть выбраны ползунками. Аэропорты вылета, прибытия, количество пересадок и авиакомпании должны выбираться прожатием галочек;
- На странице заполнения данных пассажиров вводимыми параметрами является вся паспортная информация о всех пассажирах.
- На странице “Профиль” вводимыми параметрами являются общие данные о пользователе такие как пол, дата рождения, ФИО на латинице, ФИ на русском языке. При добавлении документов и карт лояльностей вводимыми данными являются все данные о документе.

Выходные данные:

- На страницах регистрации и авторизации на сервер отправляются email и пароль пользователя. В случае неправильных данных пользователь уведомляется об ошибке и ее сообщении. В случае успеха переходит на следующую страницу;
- На странице поиска авиаперелетов при поиске городов от сервера приходит список городов, который выводится выпадающим списком. При нажатии на поиск проверяется правильность всех вводимых данных. В случае правильного ввода данных от сервера придет ответ,

Продолжение приложения А

содержащий все перелеты подходящие пользователю. В случае какой либо ошибки пользователь уведомляется об этом;

- На странице “Фильтр” при нажатии на кнопку фильтрации, сервер возвращает массив перелетов подходящих под параметры пользователя. В случае отсутствия подходящих перелетов пользователь переносится на страницу назад и уведомляется об отсутствии результатов фильтрации;
- На странице “Детали перелета” на экране приложения показывается вся доступная информация по перелету;
- После нажатия на кнопку “Забронировать перелет” от сервера приходит ответ с правилами перелета данного перевозчика, который выводится на экран.
- На следующей странице выходными параметрами являются всплывающие сообщения об успешном бронировании перелета, или о том, что не заполнено какое-либо поле. В случае успеха пользователь переносится на стартовую страницу поиска.
- На активти под названием “Мой профиль” выходными параметрами является реакция системы на поля, которые заполнил пользователь. В случае неправильного ввода выводится сообщение с указанием какое поле не заполнено или заполнено с ошибками. В случае успешного ввода всей информации пользователь отправляет данные на сервер, и получает сообщение об успешном добавлении данных.

А.1.1.4 Поддержка приложения

Предусмотреть поддержку приложения, связанная как с исправлением текущих ошибок, с развитием и обновлением приложения, связанными с функциональностью и выходом новых версий мобильных платформ.

А.1.1.5 Требования к надежности

Предусмотреть контроль вводимой информации пользователем, начиная от ввода почтового адреса, заканчивая информацией пассажиров для перелета. Предусмотреть блокировку некорректных действий пользователя при работе с мобильным приложением, отказы приложения в данном случае недопустимы. Обеспечить полное и правильное взаимодействие с системой бронирования. Предусмотреть устойчивость и работоспособность приложения при сбоях и ошибках системы.

Продолжение приложения А

А.1.1.6 Требования к составу и параметрам технических средств

Приложение должно работать на мобильных устройствах на операционной системой Android. Минимальная конфигурация: операционная система – Ice Cream Sandwich, на данной версии ОС приложение запустится на 100% всех устройств; объем ОЗУ и другие характеристики телефона не важны.

А.1.1.7 Требования к программной документации

Разрабатываемые активити приложения должны быть задокументированы с помощью комментариев разработчика. Больше требований к программной документации у заказчика нет.

Приложение Б (обязательное)

Текст программы

```
//Активти регистрации пользователя
public class RegistrationActivity extends AppCompatActivity {
    private String TAG = RegistrationActivity.class.getSimpleName();
    EditText editTextRepassword, editTextEmail, editTextPassword;
    RadioGroup radioGroupGender;
    TextView ourCompany;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
        ourCompany = (TextView) findViewById(R.id.ourCompany);
        //Нажатие на ссылку
        ourCompany.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent browserIntent = new
                    Intent(Intent.ACTION_VIEW, Uri.parse("https://ets.globalair.kz/"));
                startActivity(browserIntent);
            }
        });
        editTextRepassword = (EditText) findViewById(R.id.editTextRepassword);
        editTextEmail = (EditText) findViewById(R.id.editTextEmail);
        editTextPassword = (EditText) findViewById(R.id.editTextPassword);
        //Нажатие на кнопку регистрации
        findViewById(R.id.buttonRegister).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                registerUser();
            }
        });

        //Нажатие на кнопку авторизации
        findViewById(R.id.textViewLogin).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });
    }
}
```

Продолжение приложения Б

```
startActivity(new Intent(RegistrationActivity.this, LoginActivity.class));
} }); }
```

```
private void registerUser() {
    //Проверка данных считанных с полей
    final String email = editTextEmail.getText().toString().trim();
    final String password = editTextPassword.getText().toString().trim();
    final String repassword = editTextRepassword.getText().toString().trim();
    if (TextUtils.isEmpty(email)) {
        editTextEmail.setError(Html.fromHtml("<background-color='red'>Please
enter your email</background-color>"));
        editTextEmail.requestFocus();
        return;
    }
    if (!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        editTextEmail.setError("Enter a valid email");
        editTextEmail.requestFocus();
        return;
    }
    if (TextUtils.isEmpty(password)) {
        editTextPassword.setError("Enter a password");
        editTextPassword.requestFocus();
        return;
    }
    if (password.length() < 6) {
        editTextPassword.setError("Password should be atleast 6 character long");
        editTextPassword.requestFocus();
        return;
    }
    if (TextUtils.isEmpty(repassword)) {
        editTextRepassword.setError("Enter a repassword");
        editTextRepassword.requestFocus();
        return;
    }
    if (repassword.length() < 6) {
        editTextRepassword.setError("Repassword should be atleast 6 character
long");
        editTextRepassword.requestFocus();
        return;
    }
    if (!(repassword.equals(password))) {
        editTextRepassword.setError("Passwords do not match");
    }
}
```

Продолжение приложения Б

```
        editTextRepasword.requestFocus();
        return;
    }
    //Создание запроса
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
        URLs.URL_REGISTER,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.e(TAG, "Response from url: " + response);
                try {
                    JSONObject obj = new JSONObject(response);
                    if (!obj.getBoolean("success"))
                    {
                        Toast toast = Toast.makeText(getApplicationContext(),
                            obj.getString("message"), Toast.LENGTH_LONG);

                            View view1 = toast.getView();
                            toast.getView().setPadding(20, 20, 20, 20);
                            view1.setBackgroundResource(R.color.RED);

                            toast.show();
                        }
                    if (obj.getBoolean("success")) {
                        finish();
                        startActivity(new Intent(getApplicationContext(),
                            LoginActivity.class));
                    } else {
                        Toast.makeText(getApplicationContext(),
                            obj.getString("message"), Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(getApplicationContext(), error.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        }) {
        @Override
```

Продолжение приложения Б

```
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("email", email);
    params.put("password", password);
    return params;
}
@Override
public Map<String, String> getHeaders() throws AuthFailureError {
    Map<String, String> params = new HashMap<String, String>();
    params.put("x-requested-with", "XMLHttpRequest");
    // String creds = String.format("%s:%s", "email", "password");
    //String auth = "Basic " + Base64.encodeToString(creds.getBytes(),
Base64.DEFAULT);
    params.put("Content-Type", "application/x-www-form-urlencoded");
    //params.put("Authorization", auth);
    return params;
}
};
VolleySingleton.getInstance(this).addToRequestQueue(stringRequest); }
```