Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА Мобильного приложения для автобусной компании

Пояснительная записка к курсовому проекту

МДК.01.03 Разработка мобильных приложений

НАТКиГ.203400.010.000ПЗ

Разработал:

студент группы ПР-22.106

Сакович А.С

**Содержание**

[Введение 3](#_Toc161261558)

[1 Исследовательский раздел 4](#_Toc161261559)

[1.1 Описание предметной области 4](#_Toc161261560)

[1.2 Образ клиента 5](#_Toc161261561)

[1.3 Сценарии 6](#_Toc161261562)

[1.4 Сбор и анализ прототипов 6](#_Toc161261563)

[2 Проектирование приложения 11](#_Toc161261564)

[2.1 UI/UX дизайн приложения 11](#_Toc161261565)

[2.2 Выбор технологии, языка и среды программирования 13](#_Toc161261566)

[3 Разработка мобильного приложения 14](#_Toc161261567)

[3.1 Разработка базы данных 14](#_Toc161261568)

[3.2 Разработка мультимедийного контента 14](#_Toc161261569)

[3.3 Описание используемых плагинов 15](#_Toc161261570)

[3.4 Описание разработанных процедур и функций 16](#_Toc161261571)

[4 Тестирование 19](#_Toc161261572)

[4.1 Протокол тестирования дизайна приложения 19](#_Toc161261573)

[4.2 Протокол тестирования функционала приложения 23](#_Toc161261574)

[Заключение 25](#_Toc161261575)

[Библиография 26](#_Toc161261576)

Введение

Целью курсового проекта является создание мобильного приложения для автобусной компании.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение приложения с точки зрения пользователя для выявления функций приложения;
* разработка и реализация дизайна приложения;
* написание кода приложения;
* тестирование полученного продукта.

Объект исследования – мобильное приложение для автобусной компании.

Предмет исследования – изучение принципов функционирования и инструментов приложения.

Исследовательский раздел

## Описание предметной области

С развитием технологий, люди стали более активно пользоваться интернетом для решения повседневных задач. В том числе и для аренды автобусов. Но даже при использовании сайтов для поиска информации и бронирования, людям часто приходится совершать звонки в автобусные компании для уточнения деталей аренды. Это может быть связано с необходимостью узнать точное количество мест, проверить наличие автобуса на нужную дату, а также обсудить условия аренды и оплаты.

После того как подходящий автобус найден, необходимо лично приехать в офис автобусной компании для оформления документов и регистрации. Это занимает дополнительное время и может быть неудобно для тех, кто живет далеко от места аренды автобуса.

Для автобусных компаний процесс взаимодействия с клиентами также становится все более сложным. Сначала они принимают звонки от заказчиков, затем ищут информацию об автобусах в своей базе данных, проверяют свободные даты и связываются с водителями. Затем они должны обрабатывать заявки на аренду, проверять оплату и документы клиентов.

Все эти процессы можно значительно упростить, если создать единую онлайн-платформу, где люди смогут находить информацию об автобусах, бронировать места и оформлять все необходимые документы без необходимости совершать телефонные звонки или лично посещать офисы автобусных компаний. Такая платформа могла бы значительно сократить время на поиск информации и оформление аренды, упростив взаимодействие между заказчиками и автобусными компаниями.

Таким образом автобусные компании активно внедряют современные технологии для улучшения своего сервиса и обеспечения удобства пассажиров и своих работников. Разработка приложения для автобусной компании предоставляет новые возможности для оптимизации бронирования автобусов, а также обеспечения комфортного взаимодействия с клиентами. И упрощение взаимодействия заказчика с компанией.

## Образ клиента

Данное приложение может стать незаменимым помощником для различных групп пользователей. Во-первых, это могут быть представители туристических агентств. Благодаря приложению, им станет проще организовывать групповые поездки, выбирать наиболее подходящий маршрут и бронировать автобусы. Это позволит сэкономить время и усилия, а также предложить клиентам более качественный сервис.

Во-вторых, приложение может быть полезно для организаторов различных мероприятий. Будь то концерт, спортивный матч или конференция, такое приложение позволит им быстро и удобно найти подходящий автобус для перевозки людей.

В-третьих, приложение может заинтересовать и самих пассажиров. Если они ищут надежный и доступный способ добраться до места назначения, то данное приложение сможет предложить им широкий выбор автобусов и маршрутов, а также обеспечить удобство при бронировании.

Наконец, данное приложение может привлечь и обычных людей, которым необходимо перевезти небольшую группу людей, например, семью или друзей. Они смогут быстро и легко найти подходящий автобус и забронировать его, не выходя из дома.

Таким образом, данное приложение будет полезным для широкого круга пользователей, от туристических агентств до обычных пассажиров, нуждающихся в услугах автобусных перевозок.

## Сценарии

Молодой человек планирует организовать поездку для своей компании друзей и ему необходимо быстро и удобно забронировать автобус целиком. Раньше он звонил в автобусную компанию или искал информацию на различных сайтах, что занимало много времени и было неудобно. Но с помощью мобильного приложения для автобусной компании, он может легко выбрать подходящий автобус, забронировать его на определенную дату и время, а также получить всю необходимую информацию о поездке.

Директор кампании хочет провести корпоратив и выезд для своих сотрудников. Он обычно чтобы забронировать автобус делал это по телефону, долго говорил узнавал какие даты свободны сколько мест в автобусе и многое другое, этот разговор занимал много драгоценного времени. Сейчас с приложением ему станет гораздо легче и удобнее, теперь директор может просто открыть телефон и найти быстро и без лишних разговоров именно тот автобус, который ему нужен и выбрать желаемую дату.

## Сбор и анализ прототипов

Исследуя AppStore и GooglePlay в поиске подобных приложений, было найдено всего 1 подобное приложение, которое могло бы хоть немного совпадать с тем, что реализуется в данном приложении, данное приложение называется «Доставка сотрудников», данное приложение только для какой-то определенной компании у которой есть доступ к данному приложению, а обычному пользователю нельзя просто зайти и взять в аренду автобус. Данное приложение имеет главный экран указанный на рисунке 1 который отображается при первом входе в приложение и на нем отображается информация нужная для пользователя, который зайдет впервые, например там есть 2 кнопки одна из которых для уже существующего пользователя, а другая просто для демонстрации работы приложения.



Рисунок 1 – Главный экран приложения

После перехода по кнопке демонстрации открывается основной экран приложения, указанный на рисунке 2.

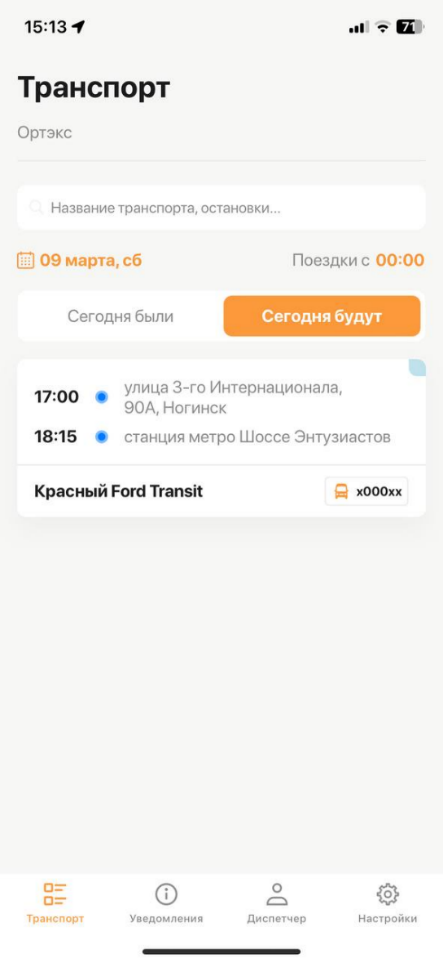


Рисунок 2 – Основной экран приложения

На данном экране можно видеть, что на нем есть поиск, в котором можно найти какую-либо остановку или название транспорта. Также на данном экране отображается информация о предстоящих и уже прошедших поездках. В нижней части экрана можно увидеть кнопки перехода на другие экраны, там имеются такие кнопки как «Транспорт», «Уведомления», «Диспетчер» и «Настройки». Экран «Транспорт» был рассмотрен выше, перейдем по остальным кнопка и ознакомимся с другими экранами.

Далее на следующем экране «Уведомления», указанном на рисунке 3 расположились соответственно уведомления, что должно быть конкретно в этих уведомлениях непонятно, поскольку данный вариант приложения является демонстрацией его функционала, но можно предположить, что здесь появляются уведомления о бронировании автобуса, о каком-то его перемещении и подобное.

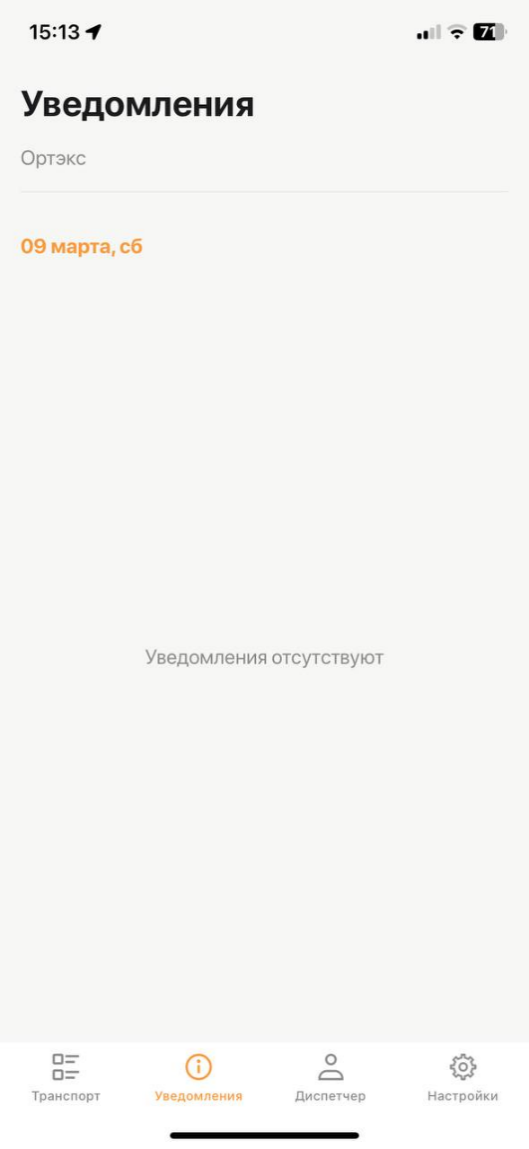


Рисунок 3 – Уведомления

Теперь рассмотрим экран «Диспетчер» указанном на рисунке 4. На данном экране можно заметить, что здесь он служит для того, чтобы связываться с диспетчерской службой данного приложения для решения каких-либо вопросов и помощью с решением тех или иных проблем с арендой автобуса с его подачей и другие интересующие вопросы.

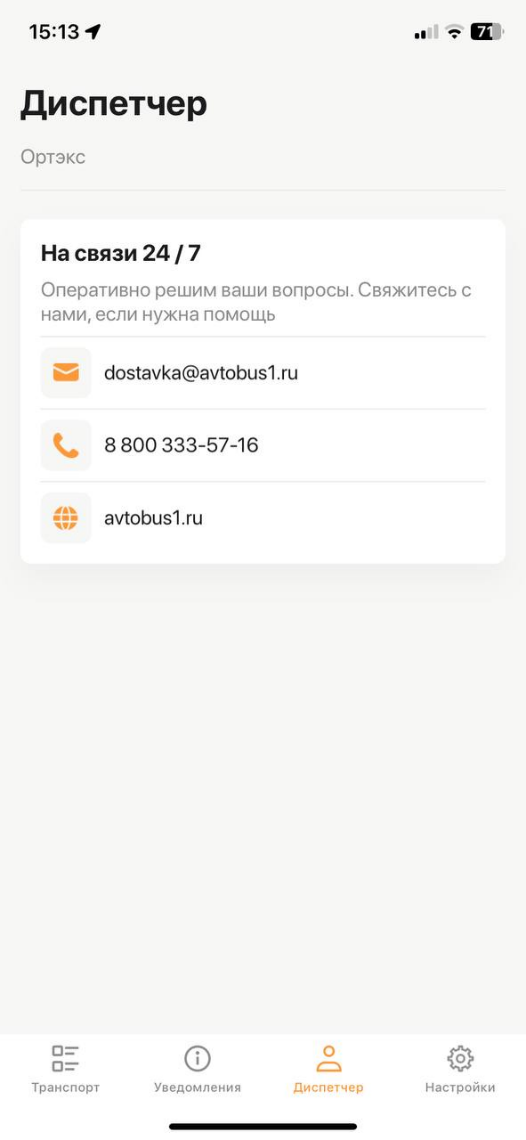


Рисунок 4 – Диспетчер

На последнем экране «Настройки», указанном на рисунке 5 можно увидеть настройки приложения, например здесь можно сменить тему приложения, оставить отзыв о приложении, выйти с приложения, а также увидеть версию данного приложения. В целом обычные настройки, которые стандартны для многих простых приложений. Помимо этого, конечно, с каждого экрана можно перейти на любой другой экран без каких-либо трудностей.

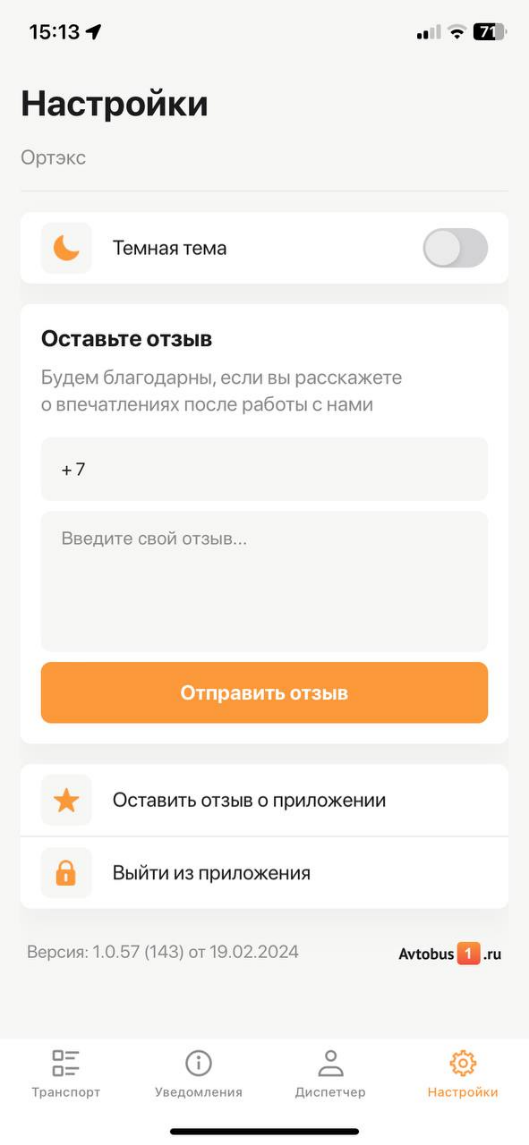


Рисунок 5 – Настройки

Таким образом рассмотрев данное приложение было принято решение реализовать приложение, которое будет простым, которое будет иметь возможность просмотра автобусов не только для одной определенной компании, а для большего круга лиц и для многих компаний, тем самым упрощая и автоматизируя процесс бронирования автобуса как для компаний, так и для заказчиков.

Проектирование приложения

## UI/UX дизайн приложения

Дизайн проекта разработан в программе Figma.

Для проекта были определены основные экраны:

* регистрация;
* авторизация;
* главный экран с автобусами;
* личный кабинет с историей заказов;
* просмотр выбранного автобуса.

Для темы мобильного приложения используется следующая цветовая схема, изображенная на рисунке 6.

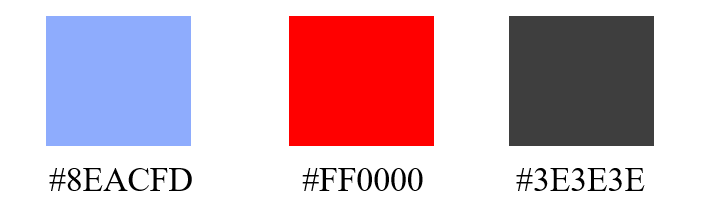


Рисунок 6 – Основная цветовая схема приложения

Данные цвета были выбраны в основном для кнопок, находящихся в приложении. Например, кнопка голубого цвета используется для всех основных кнопок, кнопка красного цвета относится к тем кнопка которые, например являются «Отменой», и соответственно кнопки серого цвета — это второстепенные кнопки, которые являются «задним фоном». Стоит заметить, что в приложении практически не используются обычные цвета для использования фона приложения, обычные цвета заменены на фоновые изображения, которые выглядят лучше и не кажутся просто обычными скучными цветами.

Логотип данного приложения предоставлен на рисунке 7.

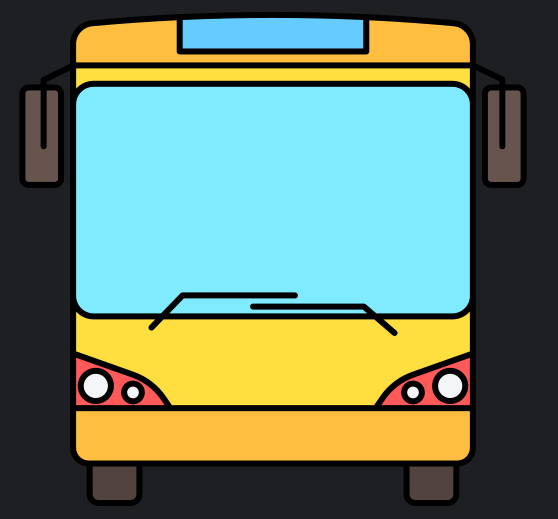


Рисунок 7 – Логотип

На логотипе как можно заметить изображен автобус лицевой частью к пользователю, что будет показывать при первом взгляде на приложение для чего оно нужно и для чего будет использоваться.

Ниже на рисунке 8 представлен дизайн приложения, а также примерное перемещение пользователя по экранам в приложении.

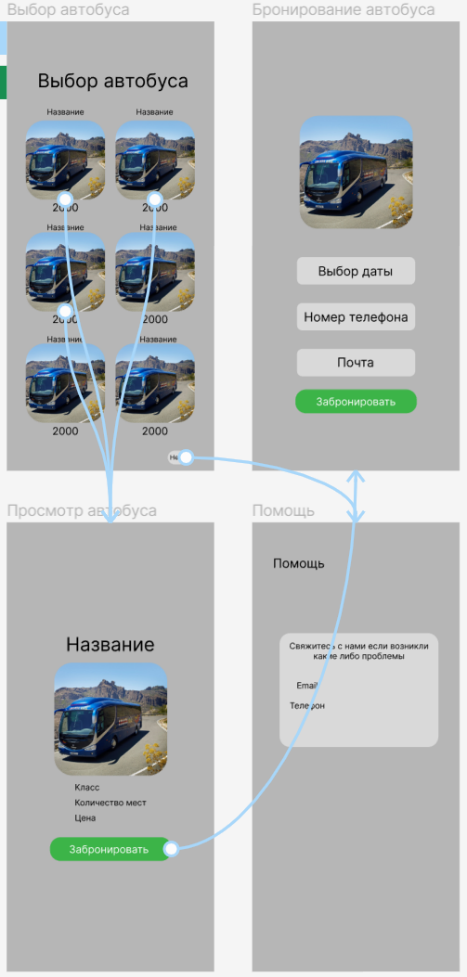


Рисунок 8 – Дизайн и перемещение пользователя в приложении

При первом запуске приложения пользователь попадает на главный экран приложения. После чего пользователю нужно нажать на кнопку «Вход», далее пользователь попадает на основной экран и оттуда у него будет возможность выбрать понравившийся ему автобус, после выбора автобуса пользователь будет перенаправлен в окно бронирования автобуса. При нажатии на кнопку «Помощь» пользователь будет перенаправлен на экран с контактной информацией. Там будет возможность узнать номер телефона и также почтовый адрес поддержки пользователей куда можно будет написать и получить необходимую техническую поддержку.

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio, так как она довольно удобна для создания приложения и является довольно популярной, поэтому в случае возникновения ошибки, легче найти способ её решения. Языком программирования является Java, так как это наиболее распространённый и популярный язык программирования.

Используемый сервис для базы данных и аутентификации – Firebase. Это бесплатная база данных, которая подходит под поставленную задачу. Она довольно быстрая и легка для понимания. Так же в ней представлены все необходимые функции: Authentication (для регистрации и авторизации пользователей), Realtime Database (для синхронизации данных), Storage (для хранения файлов в системе).

Разработка мобильного приложения

## Разработка базы данных

В качестве разрабатываемой базы данных выбрана облачная база данных Firebase, интегрируемая в Android Studio. В ней данные хранятся в формате JSON, то есть, она является NoSQL – базой данных. Пример хранения данных в базе представлен на рисунке 9.

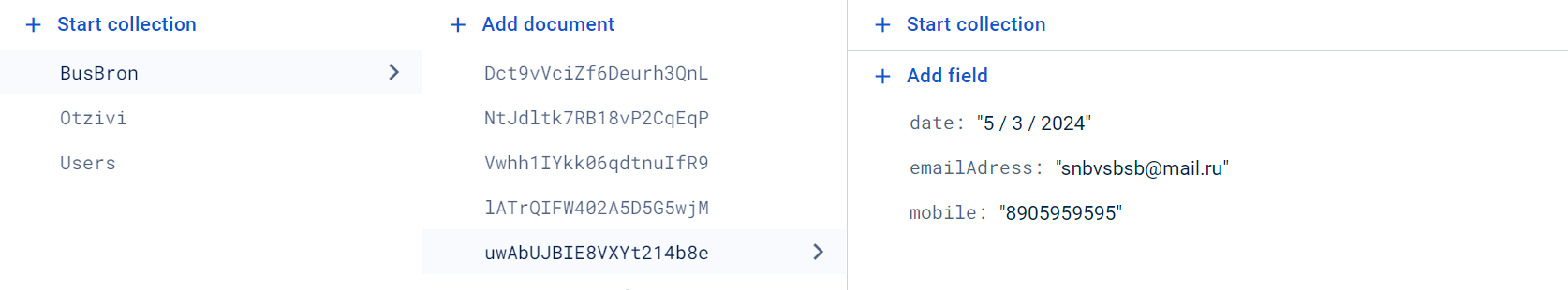


Рисунок 9 – Хранение данных в базе

База данных организована в виде древовидной структуры, где каждое бронирование представлено уникальным идентификатором (ID), который служит вершиной дерева. Каждому ID соответствует набор информации о конкретном бронировании.

В базе данных хранится вся информация о бронировании: дата, почтовый адрес пользователя, который делал бронирование, а также мобильный телефон пользователя.

## Разработка мультимедийного контента

Весь мультимедийный контент разрабатывался с помощью языка разметки XML. Вёрстка выполнялась по дизайну, разработанному ранее в приложении Figma. Однако, по мере разработки, в дизайн были внесены изменения.

Иконки и картинки в приложении были импортированы в виде файлов .png а также .jpg, данный способ импорта был более удобен. Кнопки в проект были импортированы в виде XML-файлов это представлено на рисунке 10. Все ресурсы хранятся в папе «drawble».

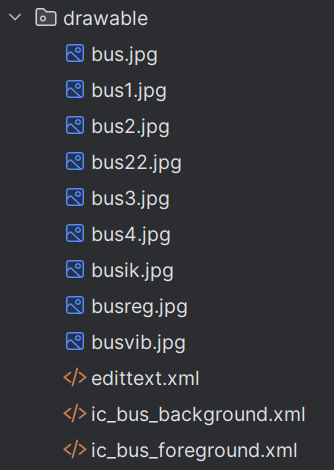


Рисунок 10 – Мультимедийный контент

## Описание используемых плагинов

В проекте используются библиотеки с различными компонентами и функциями. Список всех библиотек, а также их описание представлен в таблице 1.

Таблица 1 – Библиотеки и их описание

|  |  |
| --- | --- |
| Библиотека | Описание |
| androidx.annotation.NonNull | Библиотека, которая предоставляет аннотации для обозначения параметров, методов которые не должны принимать или возвращать значения null. Это помогает предотвратить ошибки NullPointerException в приложении. |
| androidx.fragment.app.FragmentManager | Данный класс управляет фрагментами внутри активности. Позволяет добавлять, удалять и взаимодействовать с фрагментами во время выполнения. |
| android.text.TextUtils | Этот класс предоставляет утилиты для работы с текстом, такие как проверка на пустоту, разбиение текста на части и другие операции. |

Окончание таблицы 1

|  |  |
| --- | --- |
| com.google.android.gms.tasks.OnFailureListener | Данный класс представляет слушателя для обработки успешной обработки события. |
| com.google.firebase.firestore.CollectionReference | Данный класс предоставляет ссылки на коллекции в FireStore. |
| com.google.firebase.firestore.FirebaseFirestore | Этот класс представляет базу данных Firestore в приложении Firebase. Он используется для доступа к коллекциям и документам Firestore для чтения и записи данных. |

Данные библиотеки являются неотъемлемой частью приложения, обеспечивая правильное функционирование и реализацию всех задуманных возможностей. Без данных библиотек приложение не сможет работать в полной мере, так как библиотеки предоставляют необходимые инструменты и функции, которые необходимы для работы.

## Описание разработанных процедур и функций

В приложении реализованы следующие методы предоставленные в таблице 2.

Таблица 2 – Методы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| setOnDateChangeListener() | Считывание выбранной даты с календаря |
| addDataToFirestore() | Отправка данных в базу и их последующее заполнение |
| onNavigationItemSelected() | Переключение между экранами приложения |
| onCreateDialog() | Метод для создания диалогового окна |

При бронировании автобуса данные, которые были заполнены пользователем отправляются и заполняются в базе данных.

При переходе пользователя на основной экран приложения выводятся автобусы, которые доступны для бронирования.

Когда пользователь переходит к бронированию автобуса открывается страница бронирования после чего пользователь подтверждает бронь и эти данные добавляются в базу данных. Ниже представлен код который происходит после нажатия кнопки подтверждения

Button btnPtd = findViewById(R.id.*btnPtd*);  
 btnPtd.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 Date = txDate.getText().toString();  
 Mobile = txMobile.getText().toString();  
 EmailAdress = txEmailAdress.getText().toString();  
  
 //проверка пусты ли поля  
 if (TextUtils.*isEmpty*(Date)) {  
 txDate.setError("Пожалуйста выберите дату поездки");  
 }else if (TextUtils.*isEmpty*(Mobile)){  
 txMobile.setError("Пожалуйста введите номер телефона");  
 }else if (TextUtils.*isEmpty*(EmailAdress)){  
 txEmailAdress.setError("Пожалуйста введите ваш Email");  
 }else {  
 addDataToFirestore(Date, Mobile, EmailAdress);  
  
 //показ диалового окна о успешном бронировании автобуса  
 FragmentManager manager = getSupportFragmentManager();  
 MyDialogFragment myDialogFragment = new MyDialogFragment();  
 myDialogFragment.show(manager, "myDialog");  
  
 }  
  
  
 }  
 });  
}  
  
private void addDataToFirestore(String Date, String Mobile, String EmailAdress) {  
  
 //отправка данных в базу и создании коллекции в базе данных  
  
 CollectionReference comp = BusCompany.collection("BusBron");  
  
 Base base = new Base(Date, Mobile, EmailAdress);  
  
 comp.add(base).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {  
 @Override  
 public void onSuccess(DocumentReference documentReference) {  
 //после успешного добавления данных выводится всплывающее сообщение об успешном завершении  
 Toast.*makeText*(Bronirovanie.this, "Вы успешно забронировали автобус!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(Bronirovanie.this, "Произошла ошибка попробуйте позже" + e, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
}

После отправки данных в базу появляется диалоговое окно, которое сообщает пользователю о том, что автобус успешно забронирован.

Тестирование

## Протокол тестирования дизайна приложения

Тестирование приложения проводится на Android SDK 34 на одной и той же диагонали экрана. Пример проверки основного экрана представлен на рисунке 11.



Рисунок 11 – Тестирование основного экрана

После чего на рисунке 12 представлено тестирование экрана ознакомления с конкретным автобусом, в данном случае это автобус марки «Scania».

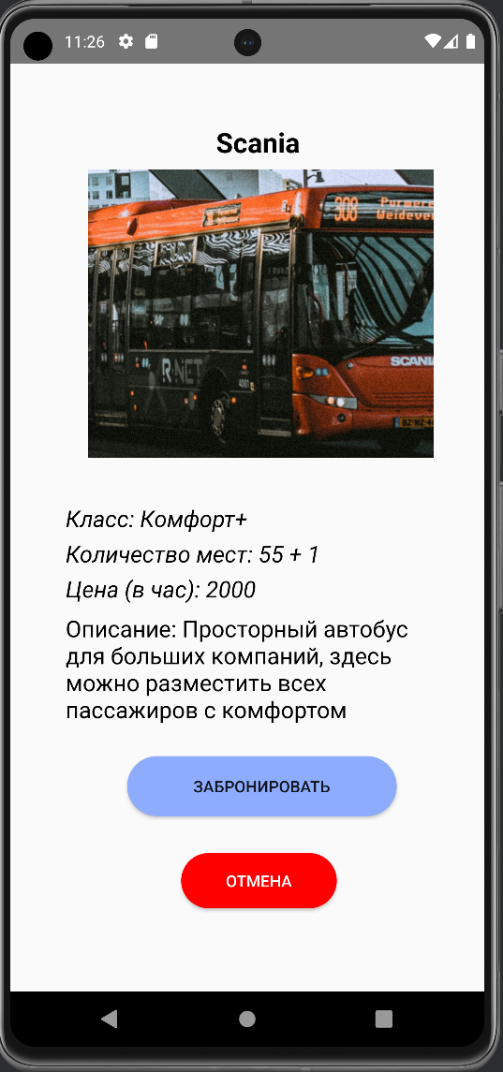


Рисунок 12 – Тестирование экрана с ознакомлением

После чего пользователь нажимает на кнопку с подтверждением бронирования и попадает на экран бронирования, представленного на рисунке 13.

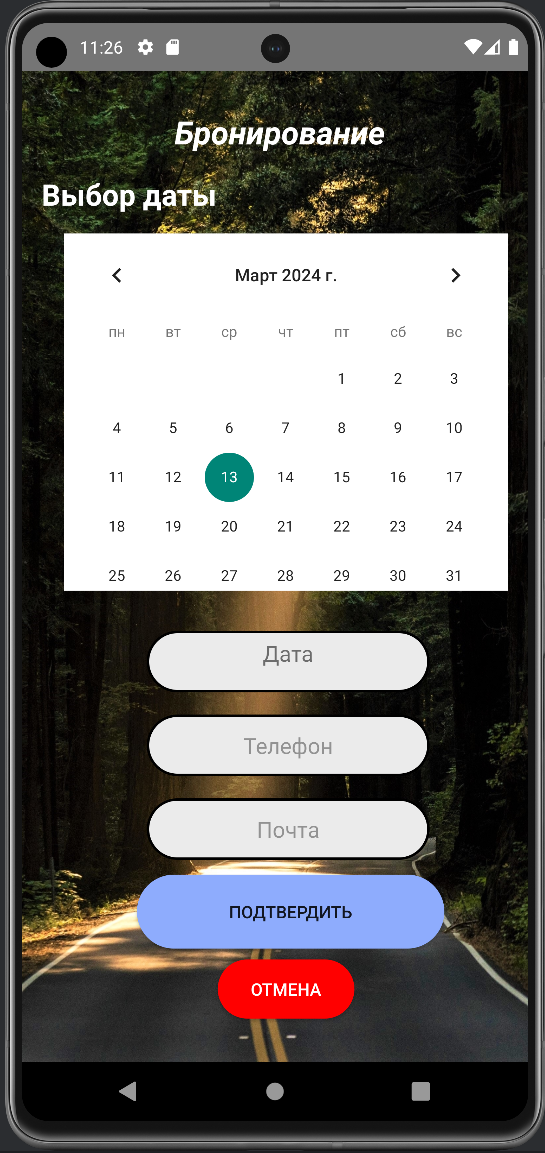


Рисунок 13 – Тестирование экрана бронирования автобуса

Далее на рисунке 14 представлен экран помощи пользователю на котором можно найти телефон поддержки, а также почтовый адрес.

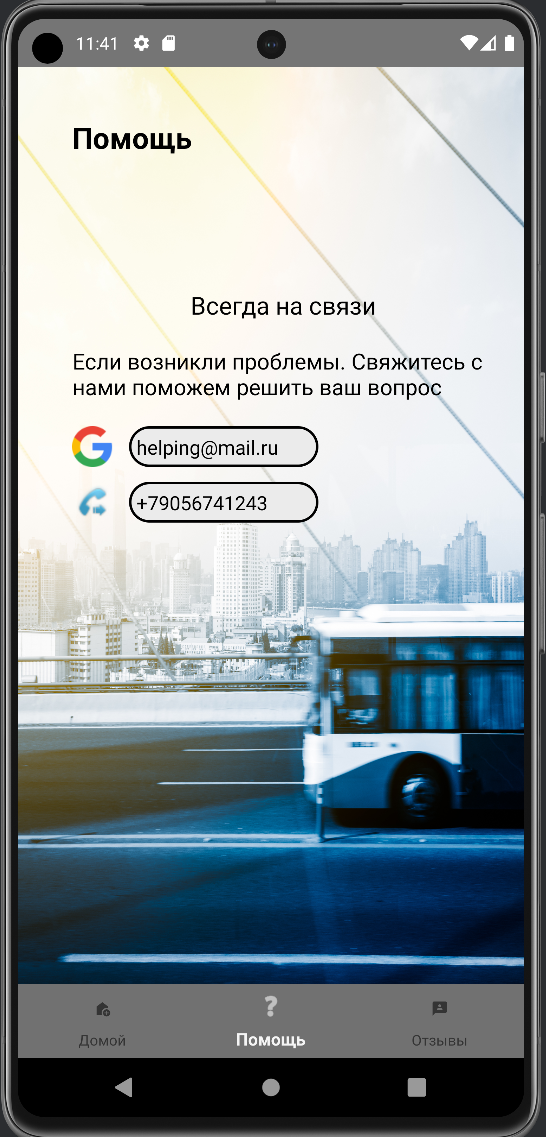


Рисунок 14 – Тестирование экрана помощи

На следующем и последнем экрана приложения, указанного на рисунке 15 представлен раздел с написанием отзыва о приложении и о работе.

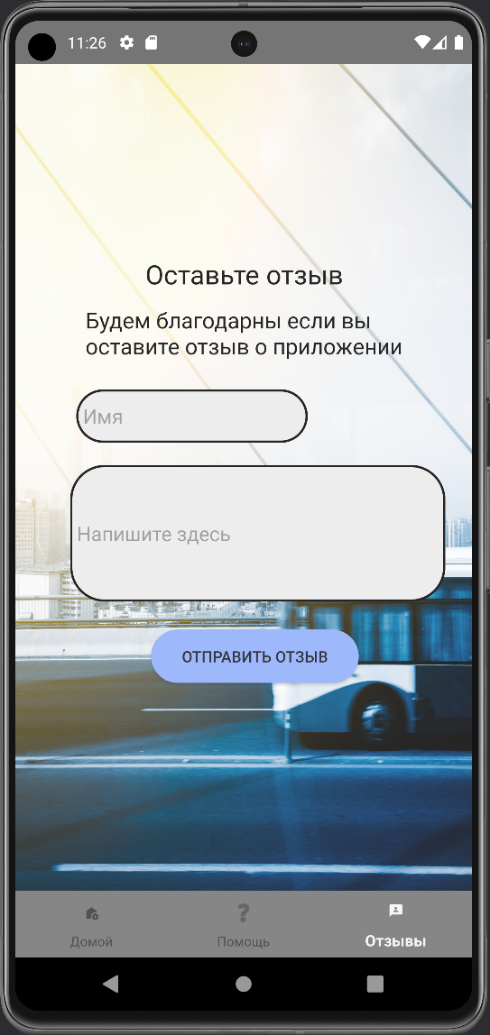


Рисунок 15 – Тестирование экрана с отзывом

Как можно заметить все экраны отображаются корректно и все элементы расположены на своих местах. Текст не растянут и не выходит за пределы экрана.

## Протокол тестирования функционала приложения

Так же необходимо проверить функционал приложения. Для этого, для каждой функции были разработаны TestCase.

В таблице 3 представлено тестирование функции бронирования автобуса.

Таблица 3 – Тестирования функции бронирования автобуса

|  |  |
| --- | --- |
| Название | BusComp |
| Функция | Бронирование автобуса |

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста:  пройден  провал  заблокирован |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| **Предусловие**: |  | |
| Запустить приложение «BusComp» | Приложение открылось корректно |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки главного экрана | Окно для вывода автобусов открылось | пройден |
| Нажать на кнопку выбора автобуса | Переход к следующему экрану с ознакомлением | пройден |
| Нажать на кнопку с надписью «Забронировать» | Переход к экрану бронирования | пройден |
| Заполнить данные, требуемые для бронирования и нажать кнопку подтверждения | В базу данных отправились данные с заполненными данными | пройден |

На следующей таблице 4 представлено тестирование отправки отзыва

Таблица 4 – тестирование функции отправки отзыва

|  |  |  |
| --- | --- | --- |
| **Название**: | BusComp | |
| **Функция**: | Отправка отзыва | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Перейти на экран отправки отзыва | Экран корректно открылся |  |
| **Шаги** **теста (positive)**: |  | |
| Дождаться загрузки экрана с отзывом | Окно для отправки отзыва открылось | пройден |
| Заполнить необходимые данные и нажать кнопку отправки отзыва | В базу данных отправлен отзыв | пройден |

Заключение

В ходе выполнения курсового проекта по разработке мобильного приложения для автобусной компании было уделено особое внимание потребностям и предпочтениям пользователей. Разработанное приложение предоставляет возможность оперативно связываться с поддержкой, отправлять отзывы и предложения, что позволяет пользователям делиться своим опытом использования приложения и получать помощь в случае возникновения каких-либо ошибок или вопросов.

Разработанный интерфейс учитывает удобство и простоту использования, минимизируя количество действий, необходимых для достижения конечной цели пользователя. Экраны приложения не перегружены информацией, предоставляя только самую необходимую и актуальную информацию.

Одним из ключевых этапов в разработке приложения стало тестирование дизайн-макета, которое подтвердило корректность его работы и соответствие всем поставленным задачам. Это позволило гарантировать, что приложение отвечает всем требованиям и успешно выполняет поставленные перед ним задачи.

Таким образом, разработка мобильного приложения для автобусной компании привела к созданию функционального и удобного инструмента, который учитывает потребности и предпочтения пользователей. Данный проект успешно достиг своих целей и является отправной точкой для дальнейшего развития и совершенствования приложения.

Библиография

Нормативно-правовые акты:

1. ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва: Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1. Android Developers [Электронный ресурс]. – Документация Android Studio. – URL: https://developer.android.com/docs/ (дата обращения: 25.11.2022)
2. Java HTML Parser [Электронный ресурс]. – Вытягивание данных с страницы. – URL: https://jsoup.org/ (дата обращения: 10.03.2023)
3. Firebase Documents [Электронный ресурс]. – Документация Firebase. Вытягивание данных с базы данных – URL: https://firebase.google.com/support (дата обращения: 25.02.2023)
4. Figma [Электронный ресурс]. – Прототипирование проектов. – URL: https://www.figma.com/ (дата обращения: 25.11.2022)

**Приложение А**

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

РАЗРАБОТКА мобильного приложения для автобусной компании

Техническое задание

НАТКиГ.203400.010.000ПЗ

Выполнил:

Студент группы ПР-22.106

Сакович А.С.

2024

**Содержание**

Введение 29

1 Назначения разработки 30

2 Требования к мобильному приложению 30

2.1 Требования к функциональным характеристикам 30

2.2 Требования к надёжности 30

2.3 Условия эксплуатации 30

2.4 Требования к составу и параметрам технических средств 30

2.5 Требования к информационной и программной совместимости 31

2.6 Требования к защите информации 31

2.7 Требования к маркировке и упаковке 31

3 Требования к программной документации 31

4 Технико-экономические показатели 32

5 Стадии и этапы разработки 32

6 Порядок контроля и приёмки 33

**Введение**

Настоящее техническое задание распространяется на разработку мобильного приложения «Разработка мобильного приложения для автобусной компании», используемого для бронирования автобусов.

Наименование приложения: «BusComp».

Краткая характеристика области применения: мобильное приложение предоставляет возможность бронирования автобусов.

Основанием для проведения разработки является Протокол №6 от 21 февраля 2022 года.

Наименование темы разработки − «Разработка мобильного приложения для автобусной компании».

Условное обозначение темы разработки – «BusComp».

**1 Назначение разработки**

Основное назначение приложения заключается в:

* обеспечении удобного и понятного интерфейса приложения для пользователя;

Лица, которые могут работать с данной системой:

пользователь приложения – может просматривать автобусы, имеющиеся в наличии, а также бронировать их и отправлять отзывы.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* просмотр автобусов;
* просмотр информации о всех имеющихся автобусах;
* написание отзывов о работе сервиса;

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация стабильного интернет-соединения.

Приложение должно контролировать входную информацию:

* соблюдение типов данных при заполнении полей.

**2.3 Условия эксплуатации**

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

**2.4 Требования к составу и параметрам технических средств**

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android не ниже версии 7.0.

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Java.

**2.6 Требования к защите информации**

Доступ к информации БД предоставляется только администратору базы данных.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* пояснительная записка.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Этапы разработки КП** | **Сроки выполнения** | **Отчётность** |
| 1 | Определение цели и задач, объекта и предмета исследования | 13.03.2023 | Пояснительная записка |
| 2 | Описание предметной области | 08.04.2023 | Пояснительная записка |
| 3 | Выбор технологии, языка и среды программирования | 10.04.2023 | Пояснительная записка |
| 4 | Оформление технического задания | 15.04.2023 | Техническое задание |
| 5 | Проектирование UI/UX дизайна | 17.04.2023 | Спецификации программного обеспечения |
| 6 | Разработка мобильного приложения | 22.04.2023 | Схема структурная системы и спецификации компонентов |
| 7 | Разработка базы данных | 02.05.2023 | Программный  продукт |
| 8 | Отладка и тестирование приложения | 13.05.2023 | Тексты программных компонентов |
| 9 | Оформление документации | 22.05.2023 | Программная  документация |
| 10 | Защита | 27.05.2023 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт;