

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения высшего  
образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

## КУРСОВАЯ РАБОТА

По Разработка приложений для мобильных операционных систем

Тема Пассажироперевозки

Руководитель

Колпаков А.А.

(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент ПИН - 121

(группа)

Карасев Э.Ф.

(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2024



В данной курсовой работе рассматривается разработка и анализ информационной системы для транспортной компании. В условиях растущего интереса к услугам транспортировки, эффективное управление информацией становится ключевым фактором для оптимизации бизнес-процессов и улучшения обслуживания клиентов. Целью работы является создание мобильного приложения на языке Kotlin с использованием среды разработки Android Studio, которое обеспечит учет маршрутов, отслеживание продаж билетов, управление данными пользователей и генерацию отчетов. В рамках работы решаются задачи по проектированию архитектуры системы, разработке базы данных, созданию пользовательского интерфейса и реализации функциональных возможностей приложения.

This coursework examines the development and analysis of an information system for a transport company. With growing interest in transportation services, effective information management is becoming a key factor in optimizing business processes and improving customer service. The goal of the work is to create a mobile application in Kotlin using the Android Studio development environment, which will provide route accounting, ticket sales tracking, user data management and report generation. The work solves the tasks of designing the system architecture, developing a database, creating a user interface and implementing the functionality of the application.

## Содержание

Введение.....	6
1. Анализ технического задания.....	8
1.1 Описание предметной области .....	8
2. Разработка моделей данных.....	10
2.1 Концептуальная модель базы данных .....	10
2.2 Логическая модель базы данных.....	12
2.3 Физическая модель базы данных .....	13
3. Проектирование работы системы.....	15
3.1 Архитектура системы .....	15
3.2 Основные функциональные требования .....	17
4. Разработка и реализация АИС .....	19
5. Тестирование АИС.....	22
6. Руководство программиста .....	23
7. Руководство пользователя.....	27
Заключение .....	29
Список литературы: .....	31
Приложение 1. ....	32
Приложение 2 .....	34
Приложение 3 .....	35

					МИВлГУ 09.03.04 - 0.010			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Шеварев Д.А.			Пассажироперевозки	Лит.	Лист	Листов
Провер.		Карасев Э.Ф.					4	41
Реценз.						МИ ВлГУ ПИН-121		
Н. Контр.								
Утверд.								

## Введение

В условиях современного рынка транспортных услуг, где мобильность становится неотъемлемой частью повседневной жизни, растет интерес к услугам транспортировки. Потребители ищут удобные и доступные способы передвижения, что делает транспортные компании особенно актуальными. Эффективное управление информацией в таких компаниях становится необходимостью, так как это позволяет не только оптимизировать бизнес-процессы, но и улучшить уровень обслуживания клиентов.

Информационные системы (ИС) играют ключевую роль в управлении транспортными компаниями. Они обеспечивают учет маршрутов, отслеживание продаж билетов, управление данными пользователей и генерацию отчетов. В результате использование ИС позволяет более эффективно реагировать на потребности клиентов и адаптироваться к изменениям рынка.

Кроме того, в современном мире, где не у каждого есть доступ к персональному компьютеру, но практически у каждого имеется мобильное устройство, становится актуальным создание приложений, которые могут работать на смартфонах и планшетах. Это позволяет значительно расширить доступность информационной системы для сотрудников компании и клиентов, обеспечивая удобство и мобильность в управлении бизнес-процессами. Мобильные приложения могут предоставлять возможность оперативного доступа к данным о маршрутах, продажах и клиентах, что особенно важно в условиях динамично меняющегося рынка.

Целью данной курсовой работы является разработка и анализ информационной системы для транспортной компании. В рамках работы планируется решить следующие задачи:

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

- Исследование бизнес-процессов транспортных компаний.
- Проектирование базы данных, отражающей специфику данного рода предприятий.
- Разработка функционала системы для работы с пользователем.

					МИВлГУ 09.03.04 – 0.010	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

## 1. Анализ технического задания.

В данной курсовой работе необходимо разработать информационную систему для транспортной компании, занимающейся продажей билетов на маршруты. Реализуемая программа должна быть разработана с помощью Android Studio. Android Studio является одной из наиболее популярных и мощных платформ для разработки мобильных приложений. Она предоставляет разработчикам широкий набор инструментов и возможностей для создания масштабируемых, высокопроизводительных и безопасных приложений.

### 1.1 Описание предметной области

Информационная система «Транспортная компания» предназначена для автоматизации процессов, связанных с продажей билетов на маршруты. В этой системе пользователи взаимодействуют с программой через различные функциональные модули, каждый из которых обеспечивает определенные функции и задачи.

#### Клиенты:

- Клиенты могут зарегистрироваться в системе, создавать учетные записи, связывая их с покупками.
- Зарегистрированные пользователи могут просматривать доступные маршруты, а также могут приобрести билет. Могут просматривать историю своих покупок.

#### Администраторы:

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



- Администраторы могут зарегистрироваться в системе, просматривать историю продаж, добавлять и удалять маршруты, а также редактировать роли пользователя.
- База данных (БД) будет основой информационной системы и обеспечит хранение, обработку и доступ к информации.

## 1.2 Формирование требований к системе

- Компания занимается продажей билетов на маршруты.
- База данных должна содержать следующие элементы: номер маршрута, пункты отправления и назначения, время отправления и прибытия, цена билета.
- Должна быть возможность регистрации и авторизации пользователей в системе. Которые будут покупать и соответственно продавать имеющиеся в наличии билеты. Администраторы будут следить за продажами билетов, а также за модерацией существующих маршрутов.

## 1.3 Обоснование выбора средств разработки

Для разработки приложения курсового проекта была выбрана среда разработки Android Studio. Для разработки серверной части был выделен язык программирования: Kotlin + Java.

К достоинствам Android Studio относятся:

- Среда разработки поддерживает работу с несколькими языками программирования, к которым относятся самые популярные – C/C++, Java, Kotlin.
- Редактор кода, с которым удобно работать.

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

- Позволяет разрабатывать приложения не только для смартфонов/планшетов, а и для портативных ПК, приставок для телевизоров Android TV, устройств Android Wear, новомодных мобильных устройств с необычным соотношением сторон экрана.
- Тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе.
- Рефакторинг уже готового кода.
- Достаточно большая библиотека с готовыми шаблонами и компонентами для разработки ПО.
- Разработка приложения для Android N – самой последней версии операционной системы.
- Предварительная проверка уже созданного приложения на предмет ошибок в нем.
- Большой набор средств инструментов для тестирования каждого элемента приложения, игры.
- Для неопытных/начинающих разработчиков специально создано руководство по использованию Android Studio, размещенное на официальном сайте утилиты.

Android Studio — это интегрированная среда разработки (IDE), предназначенная для создания приложений на платформе Android. Она предоставляет разработчикам мощные инструменты для написания, отладки и тестирования приложений, включая редакторы кода, визуальные редакторы интерфейса и встроенные средства для управления зависимостями.

## 2. Разработка моделей данных

### 2.1 Концептуальная модель базы данных

					МИВлГУ 09.03.04 – 0.010	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

Первый этап создания автоматизированной информационной системы — разработка концептуальной модели. При создании концептуальной модели базы данных мы демонстрируем область предмета, для которой предназначена система. Однако концептуальная модель не является идеальной для разработки, поэтому необходимо нормализовать все таблицы для создания реляционной базы данных. Следующим этапом является построение логической модели, которая аналогична концептуальной.

Модель учитывает все аспекты исследуемой предметной области, что позволяет получить точный результат. Далее необходимо создать физическую модель данных, которая определит, как данные будут храниться и доступны на физическом уровне: названия таблиц и столбцов, типы данных, первичные и внешние ключи и т.д.

Первым шагом в разработке концептуальной модели данных является выделение сущностей, а затем создание самой модели. Эта модель нужна для дальнейшего разрабатывания базы данных и её преобразования, например, в реляционную базу данных. Результат представлен на рисунке 1 в приложении 1.

## 2.2 Логическая модель базы данных

Графическое представление логической структуры исследуемой предметной области — цель построения логической модели. Благодаря этому моделируются сущности предметной области и их взаимоотношения.

Сущности представляют объекты, на которые направлена деятельность в данной предметной области, а также субъекты, осуществляющие эту деятельность. Свойства объектов и субъектов описываются при помощи атрибутов. Взаимоотношения между объектами и субъектами показываются при помощи связей. Правила и ограничения взаимоотношений определяются свойствами-связями. Обычно связи отражают зависимости между сущностями и влияние одной сущности на другую.

На рисунке 2 в приложении 1 представлена логическая модель для информационной системы транспортной компании. Сущность «Маршруты» определяет всю информацию о маршрутах, которые имеются в наличии или были проданы (удалены из БД). В сущности «Продажи» содержатся данные, связанные с продажей билетов, она связывает несколько сущностей: «Клиент» и «Маршруты».

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

## 2.3 Физическая модель базы данных

Физическая модель данных, представленная на рисунке 3 в приложении 1, составлена на основе логической модели, где уже отражены и учтены все основные аспекты исследуемой предметной области. После проектирования этой модели будет осуществляться непосредственно создание базы данных и её таблиц.

Опираясь на данные разработанной диаграммы сущность-связь, выделим шесть таблиц.

### Таблица «Маршруты»

В этой таблице будет храниться информация о маршрутах.

ID: {строка цифр} (Id)

Номер маршрута: {строка символов} (TripNumber)

Пункт отправления: {строка символов} (DepartureLocation)

Пункт назначения: {строка символов} (DestinationLocation)

Время отправления: {строка символов} (DepartureDatetime)

Время прибытия: {строка символов} (ArrivalDatetime)

Цена: {строка цифр} (Price)

Фото: {строка символов} (Image)

### Таблица «Билеты»

В этой таблице будет храниться информация о существующих транзакциях.

ID продажи: {строка цифр} (Id)

Маршрут: {FK} (Trip)

Покупатель: {FK} (User)

Цена: {строка цифр} (Price)

Дата продажи: {дата и время} (SaleDate)

### Таблица «Пользователь»

В этой таблице будет храниться информация о существующих пользователях и их ролях.

ID пользователя: {строка цифр} (Id)

Email: {строка символов} (Email)

Логин: {строка символов} (Login)

Пароль: {строка символов} (Password)

Роль: {строка символов} (Role)

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

### 3. Проектирование работы системы

#### 3.1 Архитектура системы

Архитектура системы включает в себя основные компоненты и их взаимодействие. В моем курсовом проекте присутствуют следующие компоненты:

Страницы приложения:

- LoginActivity: Форма авторизации.
- SignupActivity: Форма регистрации.
- TripsActivity: Главная страница с маршрутами.
- TripActivity: Страница маршрута.
- ProfileActivity: Профиль пользователя.
- AddTripActivity: Форма добавления нового маршрута.
- EditTripActivity: Форма редактирования маршрута.
- PurchaseHistoryActivity: История покупок.
- AdminActivity: Панель администратора.
- AdminUserActivity: Управление пользователями.
- AdminTripActivity: Управление маршрутами.
- AdminSaleActivity: Управление продажами.

Серверная часть:

- DBHelper: Класс для работы с базой данных.
- Таблицы: users, trips, sales.

Авторизация и регистрация:

- Пользователь вводит данные на формах LoginActivity и SignupActivity.
- Данные отправляются в DBHelper для проверки или добавления в базу данных.

#### Просмотр маршрутов:

- Пользователь просматривает маршруты с помощью TripsActivity. Данные о маршрутах загружаются из базы данных через DBHelper.

#### Покупка билета:

- Пользователь выбирает маршрут в TripActivity и нажимает кнопку "Купить".
- Данные о покупке отправляются в DBHelper для добавления в таблицу sales.

#### Профиль пользователя:

- Пользователь просматривает свои данные и маршруты на ProfileActivity.
- Данные загружаются из базы данных через DBHelper.

#### Панель администратора:

- Администратор управляет пользователями, маршрутами и продажами через AdminActivity, AdminUserActivity, AdminTripActivity, AdminSaleActivity.
- Данные обновляются в базе данных через DBHelper.



### 3.2 Основные функциональные требования

Регистрация нового пользователя:

- Пользователь вводит данные на форме SignupActivity. Данные отправляются в DBHelper для добавления в базу данных. Пользователь получает сообщение об успешной регистрации.

Авторизация пользователя:

- Пользователь вводит данные на форме LoginActivity. Данные отправляются в DBHelper для проверки. Пользователь получает доступ к системе или сообщение об ошибке.

Просмотр маршрутов:

- Пользователь просматривает маршруты на TripsActivity. Данные о маршрутах загружаются из базы данных через DBHelper.

### 3.3 Сценарии использования

Регистрация нового пользователя:

Пользователь вводит данные на форме SignupActivity. Данные отправляются в DBHelper для добавления в базу данных. Пользователь получает сообщение об успешной регистрации.

Авторизация пользователя:

Пользователь вводит данные на форме LoginActivity. Данные отправляются в DBHelper для проверки. Пользователь получает доступ к системе или сообщение об ошибке.

Просмотр маршрутов:

					МИВЛГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

Пользователь просматривает маршруты на TripsActivity. Данные о маршрутах загружаются из базы данных через DBHelper.

#### Покупка билета:

Пользователь выбирает маршрут на TripActivity и нажимает кнопку "Купить". Данные о покупке отправляются в DBHelper для добавления в таблицу sales.

#### Управление маршрутами администратором:

Администратор редактирует или удаляет маршруты через AdminTripActivity. Данные обновляются в базе данных через DBHelper.

#### Просмотр истории покупок:

Пользователь просматривает историю своих покупок на PurchaseHistoryActivity. Данные загружаются из базы данных через DBHelper.

Это описание помогает понять, как система будет работать, какие компоненты будут взаимодействовать друг с другом и какие требования должны быть учтены при разработке.

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

#### 4. Разработка и реализация АИС

Разработка и внедрение автоматизированной информационной системы (АИС) для транспортной компании является значимой задачей, требующей решения. Реализация проекта начинается с экрана авторизации и соответственно регистрации пользователя. Экран авторизации будет содержать поля для ввода email и пароля и кнопки «Авторизация» (Приложение 3 рисунок 4). В случае, если аккаунта у пользователя нет, он может зарегистрироваться, перейдя по ссылке ниже кнопки «Авторизация» (Приложение 3 рисунок 5).

Для разработки данных форм была использована таблица users:

```
CREATE TABLE users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    email TEXT,  
    login TEXT,  
    password TEXT,  
    role TEXT  
)
```

Данная таблица необходима для реализации ролей в информационной системе.

Для работы с базой данных был создан класс DBHelper, в котором реализован основной функционал: создание базы данных, отправка запросов, получение данных. Были реализованы методы addUser, getUser, getUserById, getUserIdByEmail, isEmailUnique, getAllUsers, updateUserRole, addTrip, getTrips, getTripsByUserId, updateTripStatus, addSale, getSales, getSalesByUserId, updateTrip, getTripById, getAllTrips, deleteTrip для работы с базой данных.

## Формы авторизации и регистрации:

Форма авторизации (AuthActivity) содержит поля для ввода email и пароля, а также кнопку "Войти". При нажатии на кнопку "Авторизация" вызывается метод getUser из DBHelper, который проверяет наличие пользователя в базе данных. Если пользователь существует, он авторизуется и переходит на главную страницу (TripsActivity). Если пользователь не найден, выводится сообщение об ошибке.

Форма регистрации (SignupActivity) содержит поля для ввода логина, email, телефона, пароля и чекбокс для выбора роли администратора. При нажатии на кнопку «Зарегистрироваться» добавляет нового пользователя в базу данных. Если email уже зарегистрирован, выводится сообщение об ошибке.

## Главная страница и просмотр маршрутов:

После успешной авторизации пользователь попадает на главную страницу (TripsActivity), где отображаются все доступные маршруты (Приложение 3 рисунок 6). Маршруты отображаются в RecyclerView с использованием адаптера TripsAdapter. Пользователь может просматривать маршруты, а также использовать поиск для фильтрации маршрутов по пункту отправления, пункту назначения или времени.

При нажатии на маршрут открывается страница маршрута (TripActivity) (Приложение 3 рисунок 7), где отображаются подробные сведения о маршруте. Если пользователь не является администратором, он может нажать кнопку "Купить", чтобы приобрести билет. При нажатии на кнопку "Купить" добавляет новую запись в таблицу tickets и в историю покупок пользователя.

## Профиль пользователя:

					МИВлГУ 09.03.04 – 0.010	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

Пользователь может перейти на страницу профиля (ProfileActivity), где отображаются его личные данные, список его маршрутов и история покупок (Приложение 3 рисунок 8,9). Если пользователь является администратором, он может перейти на панель администратора (AdminActivity).

Панель администратора:

Администратор имеет доступ к панели управления (AdminActivity) (Приложение 3 рисунок 10), где он может управлять пользователями, маршрутами и продажами. Администратор может просматривать всех пользователей (Приложение 3 рисунок 11), все маршруты (Приложение 3 рисунок 12) и все продажи (Приложение 3 рисунок 13).

Администратор может добавлять и удалять маршруты. Для добавления нового маршрута используется метод addTrip из DBHelper, для удаления маршрута — метод deleteTrip.

История покупок:

Пользователь может просматривать историю своих покупок на странице истории покупок (PurchaseHistoryActivity) (Приложение 3 страница 14). История покупок отображается в RecyclerView с использованием адаптера PurchaseHistoryAdapter. Для получения истории покупок используется метод getSalesByUserId из DBHelper.

Рекомендуемые системные требования:

Операционная система Android: 8.0.0;

Процессор с частотой 1 ГГц;

4 ГБ оперативной памяти;

Не менее 50 МБ места на жёстком диске;

## 5. Тестирование АИС

Основная цель проведения тестирования состоит в проверке соответствия реализации системы требуемой функциональности. Во время тестирования каждый тест регистрируется и его результаты сравниваются с ожидаемым. Если ожидаемый результат не совпадает с фактическим, это отмечается в протоколе тестирования. В таблице 1 представлена методика проведения тестирования разработанного программного продукта.

Выполненное действие	Полученный результат
Запуск исполняемого файла	Отображение формы авторизации.
Переход по ссылке	Отображение формы регистрации.
Заполнение данных	Вывод сообщения “пользователь добавлен”
Заполнение существующими данными	Вывод сообщения "пользователь уже существует!"
Авторизация под именем «admin»	Отображение интерфейса администратора.
Удаление маршрута	Удаление данных из списка.
Изменение роли пользователя	После авторизации пользователь может зайти на страницу администратора
Авторизация пользователя	Отображение интерфейса пользователя. Возможность покупки товара
Нажатие кнопки «купить»	В профиле пользователя добавляется история покупок билетов, а также обновляется счетчик занятых мест по маршруту. Отображение данных у администратора

Таблица 1 – методика тестирования разработанной программы

Результаты, полученные в ходе тестирования разработанного программного продукта, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания.

## 6. Руководство программиста

Приложение включает в себя работу с базой данных, аутентификацию и регистрацию пользователей, а также функциональность для продажи клиентами билетов и соответственно их покупки.

Основные компоненты приложения:

AuthActivity: Форма авторизации.

SignupActivity: Форма регистрации.

TripsActivity: Главная страница с маршрутами.

TripActivity: Страница маршрута.

ProfileActivity: Профиль пользователя.

AddTripActivity: Форма добавления нового маршрута.

AdminActivity: Панель администратора.

AdminUserActivity: Управление пользователями.

AdminTripActivity: Управление маршрутами.

TripsAdapter: Класс для работы с выводом данных о маршруте.

Ticket: Модель билетов.

User: Модель пользователя.

Trip: Модель маршрута.

DBHelper: Класс для работы с базой данных SQLite.

Основные методы и функции:

addUser(user: User): Добавляет нового пользователя в базу данных.

getUser(email: String, pass: String): Проверяет наличие пользователя с указанными email и паролем.

getUserById(userId: Int): Возвращает пользователя по его ID.

getUserIdByEmail(email: String): Возвращает ID пользователя по его email.

getUserIdByProductId(productId: Int): Возвращает ID пользователя по ID продукта.

isEmailUnique(email: String): Проверяет уникальность email.

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

getAllUsers(): Возвращает список всех пользователей.

updateUserRole(userId: Int, isAdmin: Boolean): Обновляет роль пользователя (администратор или нет).

addTrip(trip: Trip, userId: Int): Добавляет новый маршрут в базу данных.

getTrips(): Возвращает список всех маршрутов.

getTripsByUserId(userId: Int): Возвращает список маршрутов, проданных конкретным пользователем.

updateTripStatus(tripId: Int, status: Boolean): Обновляет статус маршрута (продан или нет).

addSale(sale: Sale): Добавляет новую продажу в базу данных.

getSales(): Возвращает список всех продаж.

getSalesByUserId(userId: Int): Возвращает список продаж, совершенных конкретным пользователем.

updateTrip(trip: Trip): Обновляет информацию о маршруте.

getTripById(tripId: Int): Возвращает маршрут по его ID.

getAllTrips(): Возвращает список всех маршрутов.

deleteTrip(tripId: Int): Удаляет маршрут из базы данных.

### LoginActivity

onCreate(savedInstanceState: Bundle?): Инициализирует активность авторизации, устанавливает обработчики событий для кнопок и полей ввода.

### SignupActivity

onCreate(savedInstanceState: Bundle?): Инициализирует активность регистрации, устанавливает обработчики событий для кнопок и полей ввода.

### TripsActivity

onCreate(savedInstanceState: Bundle?): Инициализирует главную активность, устанавливает обработчики событий для кнопок и полей ввода, загружает список маршрутов.



filterTrips(query: String?): Фильтрует список маршрутов по заданному запросу.

### TripActivity

onCreate(savedInstanceState: Bundle?): Инициализирует активность отображения отдельно взятого маршрута, устанавливает обработчики событий для кнопок.

### ProfileActivity

onCreate(savedInstanceState: Bundle?): Инициализирует активность профиля пользователя, устанавливает обработчики событий для кнопок и полей ввода, загружает информацию о пользователе и его маршрутах.

### AdminTripActivity

onCreate(savedInstanceState: Bundle?): Инициализирует активность администратора для управления маршрутами, загружает список маршрутов.  
onResume(): Обновляет список маршрутов при возобновлении активности.

### AddTripActivity

onCreate(savedInstanceState: Bundle?): Инициализирует активность добавления нового маршрута, устанавливает обработчики событий для кнопок и полей ввода.  
addTrip(): Добавляет новый маршрут в базу данных.

### TripsAdapter

onCreateViewHolder(parent: ViewGroup, viewType: Int): Создает новый ViewHolder для отображения элемента списка маршрутов.  
onBindViewHolder(holder: MyViewHolder, position: Int): Привязывает данные маршрута к ViewHolder.

updateTrips(newTrips: List<Trip>): Обновляет список маршрутов и уведомляет адаптер о изменениях.

## 7. Руководство пользователя

### Руководство пользователя для мобильного приложения

Это руководство поможет вам освоиться с основными функциями и страницами Android-приложения. В нем описаны ключевые действия, которые доступны пользователю.

При запуске программы пользователь попадает на страницу регистрации, заполняет данные, далее нажимает на кнопку «Регистрация». После этого клиент получит сообщение об успешной регистрации, и перейдя по ссылке ниже попадает на страницу авторизации. Далее необходимо ввести данные, указанные при регистрации, и нажать на кнопку «Авторизация». После нажатия на кнопку пользователь увидит сообщение об успешной авторизации и попадет на страницу с маршрутами. Пользователь может найти какой-либо маршрут, введя в поисковую строку название или модель интересующего маршрута. Далее пользователь может выбрать маршрут, и после нажатия на кнопку «Просмотреть» пользователь попадет на страницу «Информация о маршруте», где указана основная информация о выбранном маршруте. Далее пользователь может нажать на кнопку «Купить», после чего вылетит подтверждение о покупке. После покупки пользователь может перейти на страницу своего профиля, нажав на кнопку в правом верхнем углу. В профиле пользователь может увидеть страницу, где указан список купленных им билетов.

Регистрация и авторизация администратора практически не отличается от пользователя, за исключением того, что при регистрации администратора необходимо попросить у главного администратора выдать ему роль. После успешной авторизации администратор может перейти в профиль, где присутствует основная информация с выбором разделов. Выбрав нужный раздел, мы можем просмотреть какую-либо информацию. При переходе в раздел «Пользователи» мы сможем увидеть список зарегистрированных пользователей и поменять им роль, нажав на определенную кнопку

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

«Пользователь» или «Администратор». При переходе в раздел «Продажи», мы сможем просмотреть подробную информацию о проданных компанией билетах. При переходе в раздел «Маршруты» мы увидим список всех маршрутов.

					МИВлГУ 09.03.04 – 0.010	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

## Заключение

В заключение хотелось бы отметить, что данная курсовая работа, разработанная на Android Studio, была предназначена для организации эффективного управления бизнес-процессами в данной сфере.

В ходе выполнения работы был осуществлен анализ требований к системе, проведено проектирование базы данных на SQLite, разработаны необходимые функциональные модули и интерфейсы на Android Studio, а также проведено тестирование и отладка системы.

Информационная система транспортной компании, основанная на сочетании SQLite и Android Studio, обладает рядом преимуществ. Прежде всего, данная связка технологий обеспечивает надежное хранение данных в базе данных, эффективное извлечение и обработку информации, а также обеспечивает удобный пользовательский интерфейс для взаимодействия с системой.

Разработанная информационная система позволяет упростить и автоматизировать процессы продажи билетов на маршруты.

В результате выполнения данной работы была достигнута поставленная цель — разработка информационной системы, способной повысить эффективность работы транспортной компании. Однако, для ее дальнейшего успешного использования и развития, рекомендуется провести дополнительное тестирование системы на более широком объеме данных, а также реализовать функции взаимодействия с другими информационными системами.

В целом, данная работа демонстрирует возможности и перспективы использования Android Studio для разработки информационных систем, а

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

также подчеркивает значимость автоматизации бизнес-процессов для повышения эффективности и конкурентоспособности организации.

Список литературы:

1. Введение в разработку приложений для смартфонов на ОС Android / А.Семакова Национальный открытый Университет «ИНТУИТ», 2016
2. Колисниченко Д.Н. Программирование для Android 5. Самоучитель. — СПб.: БХВ-Петербург, 2015. — 303 с.
3. 2016. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е изд. — СПб.: Питер, Гриффитс Дэвид, Гриффитс Дон Head First. Программирование для Android. 2-е изд. — СПб.: Питер, 2018. — 912 с.

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		31

## Приложение 1.

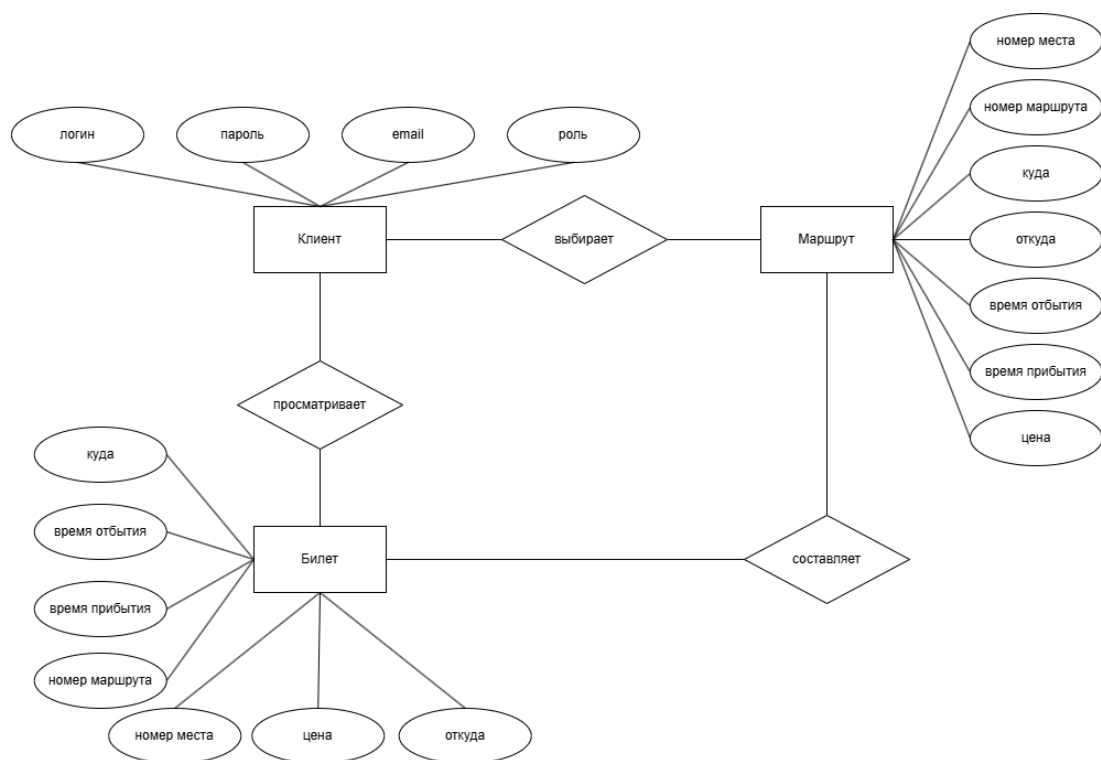


Рисунок 1 - Концептуальная модель данных

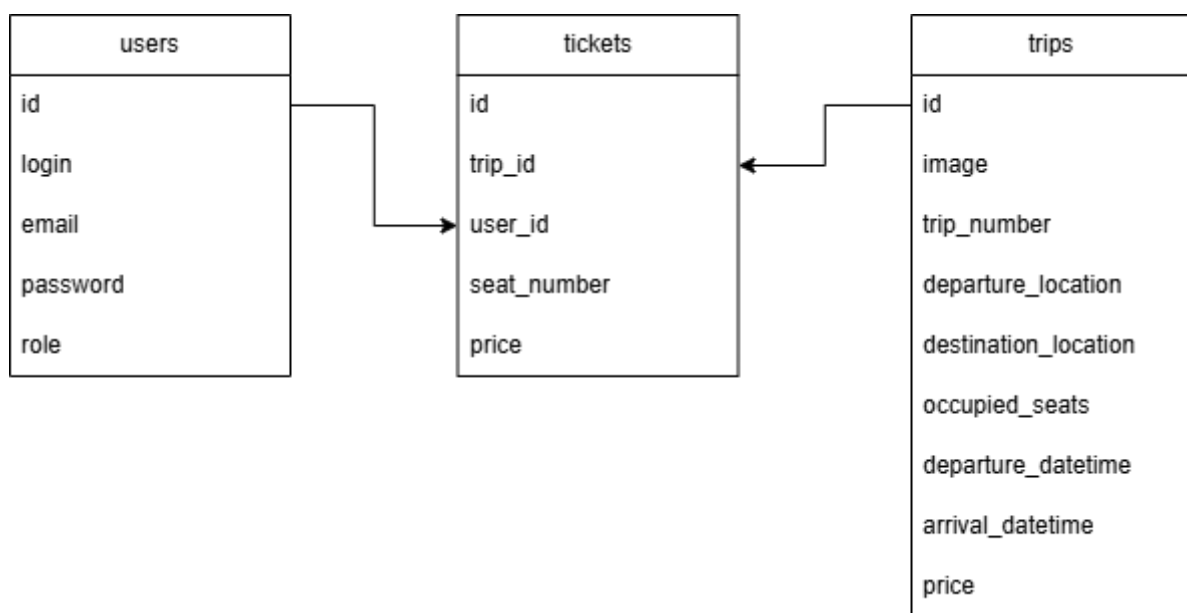


Рисунок 2 – Логическая модель данных



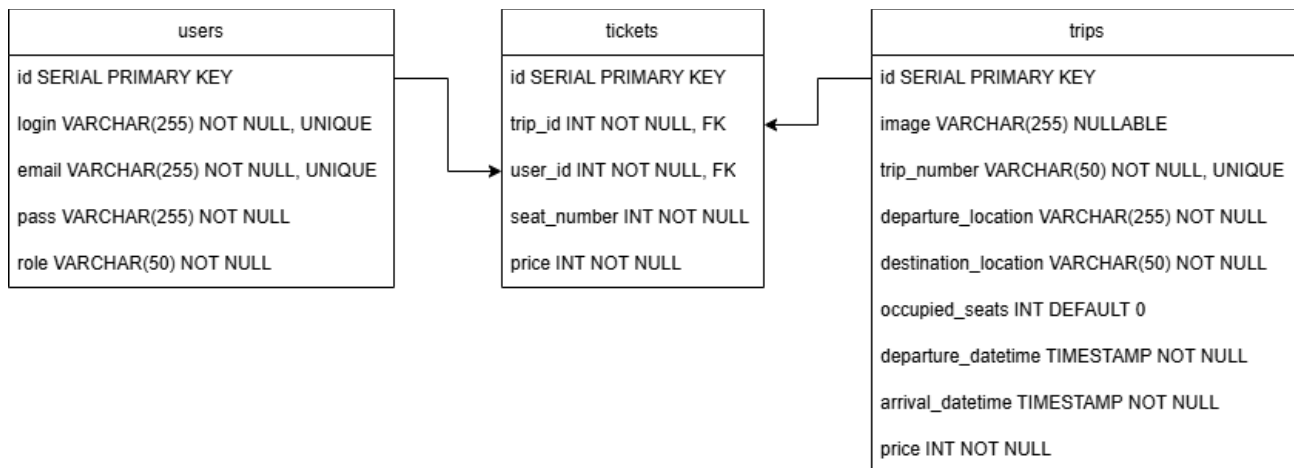


Рисунок 3 – Физическая модель данных

## Приложение 2

С подробным кодом программы вы можете ознакомиться по данной ссылке:  
<https://github.com/E-Yokou/BusCompanyApp.git>

					МИВлГУ 09.03.04 – 0.010	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

### Приложение 3

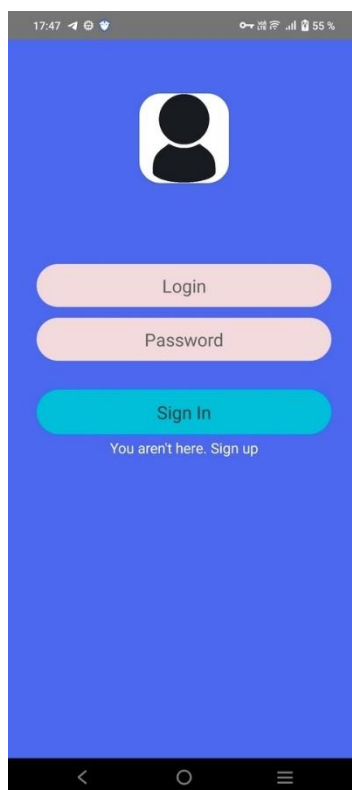


Рисунок 4 – Экран авторизации

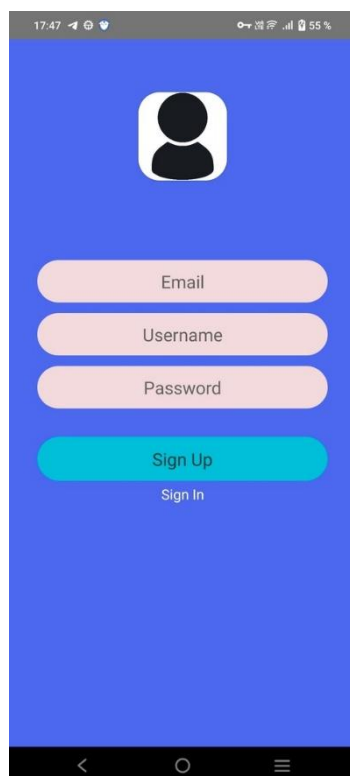


Рисунок 5 – Экран регистрации

					МИВлГУ 09.03.04 – 0.010	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

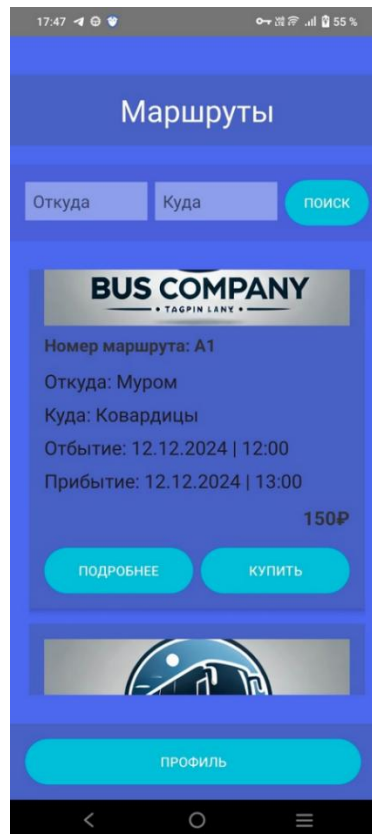


Рисунок 6 – Главный экран со списком маршрутов

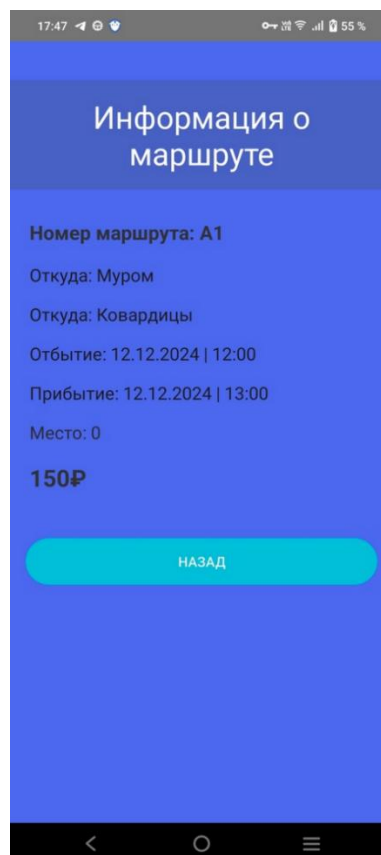


Рисунок 7 – Экран просмотра товара

					МИВлГУ 09.03.04 – 0.010	Лист
						36
Изм.	Лист	№ докум.	Подпись	Дата		

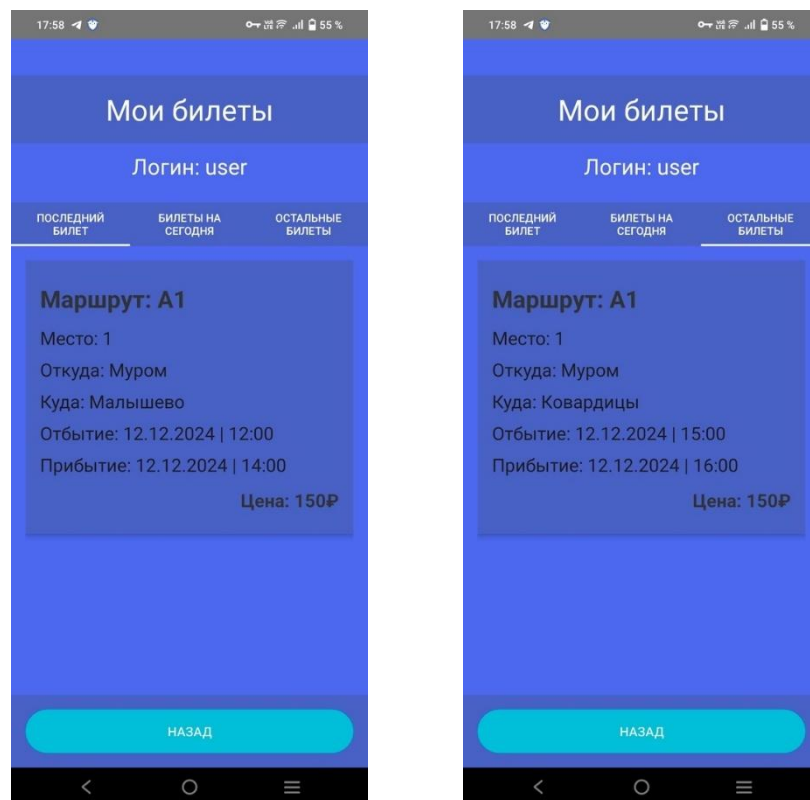


Рисунок 8, 9 – Профиль пользователя

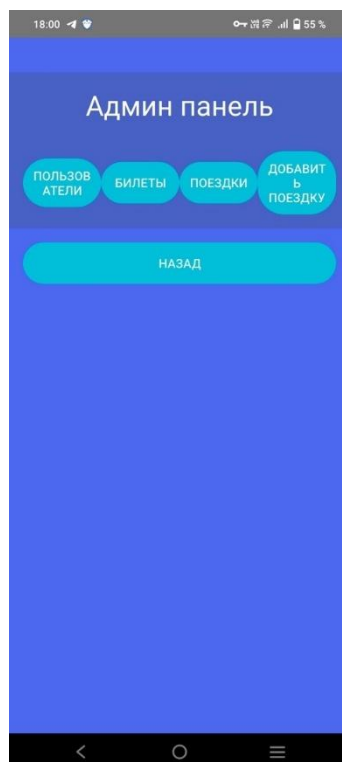


Рисунок 10 – Панель админа

					МИВЛГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

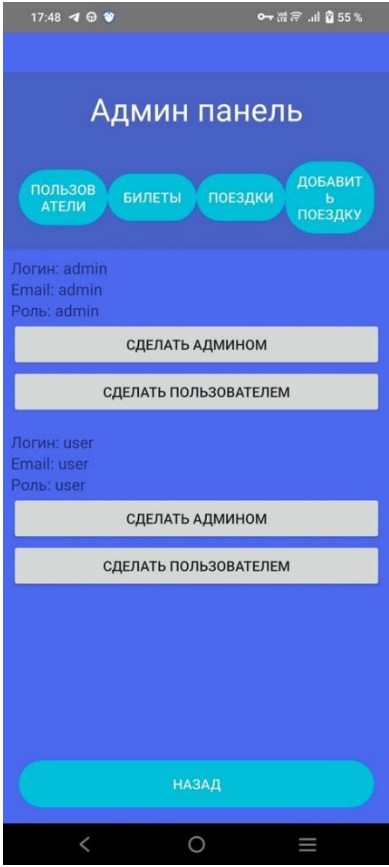


Рисунок 11 – Список всех пользователей

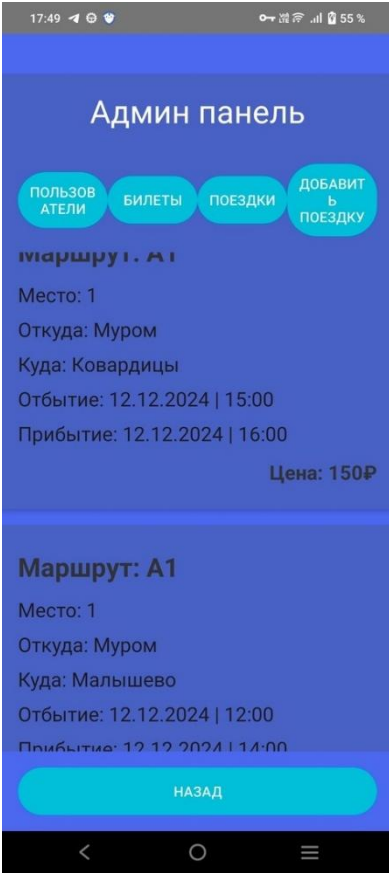


Рисунок 12 – Список всех маршрутов

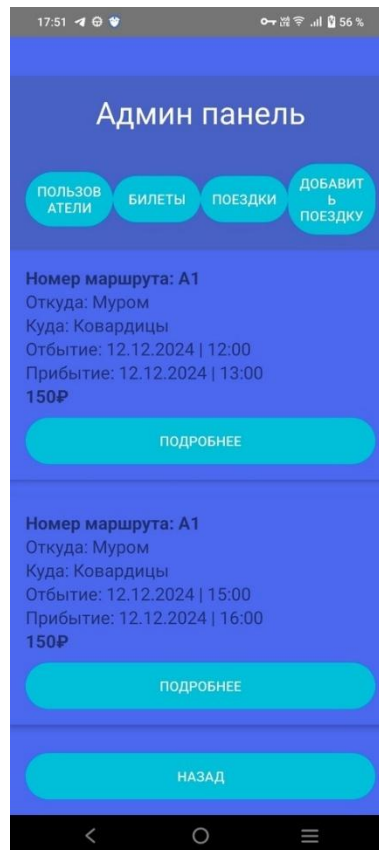


Рисунок 13 – Список всех продаж

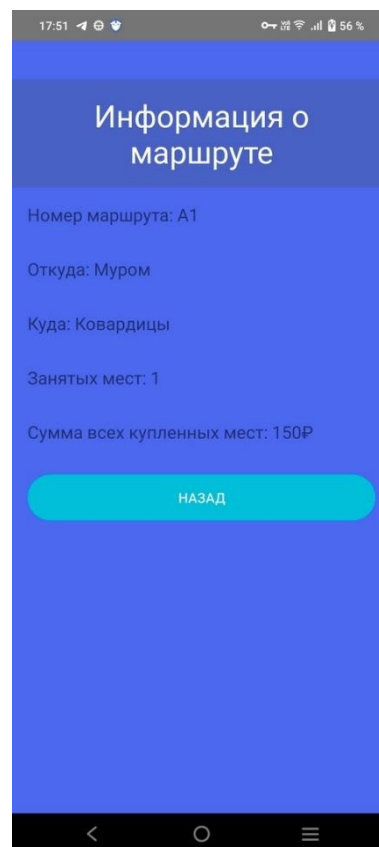


Рисунок 14 – Подробности о маршруте

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39