

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. Постановка задачи	8
1.2. Назначение разрабатываемого программного продукта	8
2. Разработка алгоритмов	11
2.1. Концептуальная модель данных.....	11
2.2. Логическая модель данных	12
2.3. Физическая модель данных	13
2.4. Разработка диаграммы прецедентов	13
2.5. Разработка и реализация приложения	14
2.6. Тестирование приложения	15
3. Руководство программиста	17
4. Руководство пользователя.....	24
Заключение	27
Список литературы:	28
Приложение 1.	29
Приложение 2	31
Приложение 3	32

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

Введение

В настоящее время для достижения успехов любому предприятию необходимо точно понимать свои затраты, прибыли, ресурсы, бизнес-процессы и многое другое. Наглядная информация о происходящем поможет глубже проанализировать процесс и сделать правильные выводы, что в конечном итоге приведет к повышению эффективности. Лидерами становятся наиболее эффективные предприятия, имеющие минимальные издержки, высочайший уровень производительности и полностью контролируемые и четко отлаженные процессы. Мобильные приложения способствуют повышению производительности, улучшению связи с клиентами или оптимизации бизнес-процессов.

Мобильные приложения позволяют компаниям лучше взаимодействовать со своими клиентами, предоставляя им удобный способ запросить информацию, обращаться в службу поддержки или совершать покупки. Приложения для бизнеса могут служить дополнительным каналом продаж, позволяя клиентам совершать покупки и ознакомиться с продуктами или услугами непосредственно через приложение. Мобильные приложения могут помочь компаниям собирать данные о поведении пользователей, предпочтениях и трендах, что позволяет проводить аналитику и улучшать стратегии маркетинга и продаж.

В данной курсовой работе рассматривается создание мобильного приложения для транспортной компании, базирующееся на операционной системе Android с использованием языка программирования Kotlin. Транспортная компания занимается продажей билетов на маршруты и предоставляет услуги по перевозке пассажиров. Одной из возможностей повышения рентабельности транспортной компании является использование

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

клиентского приложения для оформления клиентами заявок на покупку билетов.

Целями данной работы являются разработка моделей информационной системы транспортной компании с учетом требуемых обработки данных для последующего проектирования клиентского мобильного приложения. Задачей проекта является программная реализация мобильного приложения, проверка его работоспособности.

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

1. Анализ технического задания.

1.1. Постановка задачи

Разрабатываемое приложение «Транспортная компания» предназначено для удобного оформления клиентами заявок на покупку билетов на маршруты. Программный продукт будет использоваться частными лицами, желающими приобрести билеты на транспортные средства. Приложение предоставляет пользователю следующие возможности:

- регистрация и авторизация пользователей;
- просмотр доступных маршрутов;
- покупку билета;
- просмотр истории покупок.

1.2. Назначение разрабатываемого программного продукта

Разрабатываемое мобильное приложение «Транспортная компания» создается с целью обеспечения удобства покупки билетов на маршруты. Назначение разрабатываемого программного продукта включает в себя:

1) Упрощение процедуры покупки билетов: приложение предоставляет легкий и удобный способ регистрации пользователей, просмотра доступных маршрутов и оформления заявок на покупку билетов.

2) Хранение истории покупок: в приложении присутствует возможность просмотра списка оформленных заявок. В данных заявках пользователь может наблюдать статус заявки.

3) Предоставление информации по запросу: пользователи могут получить доступ к информации о доступных маршрутах и их деталях. Это упрощает процесс принятия решений при выборе маршрутов.

Преимущества автоматизации включают в себя:

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

1) Ускорение обработки информации: система обеспечивает быстрый и легкий доступ к данным, что ускоряет процесс оформления заявки на покупку билета и снижает временные траты.

2) Удобство и доступность: пользователи могут легко и быстро взаимодействовать с приложением, вне зависимости от их технических навыков.

3) Адаптация к современным тенденциям: учитывается рост цифровых технологий и предоставляется возможность пользователям взаимодействовать с системой в рамках современных цифровых тенденций.

1.3. Анализ исходных данных к курсовой работе

Выбор использования операционной системы Android и языка программирования Kotlin обусловлен следующими факторами:

- Поддержка Google: Kotlin является официальным языком для разработки под Android. Google активно поддерживает его и внедряет новые функции, что делает язык более стабильным и перспективным для мобильной разработки.

- Современные возможности: Kotlin предлагает множество современных возможностей, к примеру, расширенные функции работы с коллекциями и мощную систему типизации, что упрощает и ускоряет разработку.

- Интероперабельность с Java: Kotlin полностью совместим с Java, что позволяет использовать существующие Java-библиотеки и постепенно переходить на новый язык без необходимости переписывать весь код.

- Сообщество и ресурсы: сообщество разработчиков Kotlin растёт, появляются новые библиотеки, фреймворки и учебные материалы, что делает процесс изучения более доступным и поддерживаемым.

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

Приложения, написанные на Android, могут быть запущены на множестве устройств, включая виртуальные машины и домашние приставки.

Разрабатываемое приложение должно иметь возможности:

- 1) Регистрация и авторизация пользователей.
- 2) Просмотр доступных маршрутов.
- 3) Оформление заявок на покупку билетов.
- 4) Просмотр истории покупок.
- 5) Получение уведомлений о статусе заявок.

При разработке мобильного приложения на Android для хранения и организации информации было принято решение об использовании базы данных. В качестве базы данных использовалась встраиваемая реляционная база данных SQLite.

SQLite представляет собой встраиваемую реляционную базу данных, которая обеспечивает легкий доступ к базе данных без необходимости отдельного сервера. SQLite широко используется в мобильных приложениях, встроенных системах и других проектах, где требуется компактная и простая в использовании база данных.

2. Разработка алгоритмов

Разработка алгоритмов является ключевым этапом в создании приложения. Этот процесс не только обеспечивает эффективное функционирование программы, но и является основой для ряда других важных шагов в разработке. В данном контексте осуществление следующих пунктов является неотъемлемой частью разработки алгоритмов.

Разработка концептуальной модели данных: данная модель необходима для представления в визуальном удобном виде связей между объектами данных и их характеристиками, впоследствии необходима для разработки логической и физической моделей данных.

Разработка логической и физической модели данных: эти модели служат основой для оптимизированного проектирования алгоритмов работы с информацией.

Разработка диаграммы прецедентов: диаграмма прецедентов помогает выделить основные функциональные возможности приложения и отобразить возможное взаимодействие пользователя с системой. Разработка алгоритмов тесно связана с определением шагов, которые пользователь может предпринять, и соответствующих им процессов обработки данных.

2.1. Концептуальная модель данных

Концептуальная модель хранилища данных представляет собой описание главных (основных) сущностей и отношений между ними. Концептуальная модель является отражением предметных областей, в рамках которых планируется построение хранилища данных. При проектировании концептуальной модели структурируют данные и выявляют взаимосвязи

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

между ними, без рассмотрения особенностей реализации и вопросов эффективности обработки, поэтому концептуальная модель не является полностью подходящей для дальнейшей разработки, все таблицы должны быть нормализованы для реляционной базы данных. Составленная концептуальная модель представлена на рисунке 1 приложения 1.

2.2. Логическая модель данных

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними. Концептуальная модель изменяется так, чтобы она могла быть обеспечена конкретной моделью данных. В результате формируется логическая модель. Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения. Логическая модель может быть реляционной, иерархической или сетевой.

В качестве способа организации информационной базы выбрана реляционная база данных. Именно такой способ хранения всех данных является наиболее подходящим для проектируемой информационной системы по следующим причинам:

- наглядность модели для пользователя: все данные в реляционной модели представлены в табличной форме;
- независимость данных от программного продукта для их обработки;
- реляционные базы данных являются наиболее распространенными среди разработчиков ПО, следовательно, использование этих баз позволит сэкономить время и бюджет на внедрение нового типа БД.

Составленная логическая модель представлена на рисунке 2 приложения 1.

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

2.3. Физическая модель данных

Физические модели данных служат для отображения моделей данных. Основными понятиями модели данных являются поле, логическая запись, логический файл. Слово «логический» введено, чтобы отличать понятия, относящиеся к логической модели данных, от понятий, относящихся к физической модели данных. Основными понятиями физической модели данных, используемыми для представления логической модели данных, являются поле, физическая запись, физический файл. В частности, логическая запись, состоящая из полей, может быть представлена в виде физической записи (из тех же полей), логический файл – в виде физического файла. Имена таблиц и колонок будут сгенерированы на основе сущностей и атрибутов логической модели, учитывая максимальную длину имени и другие синтаксические ограничения, накладываемые СУБД. Если в имени сущности или атрибута встречается пробел, он заменяется на символ «_».

Физическая модель описывает способ хранения данных в базе данных. В физической модели мы учитываем типы данных, индексы, ограничения целостности и другие технические детали. Составленная физическая модель представлена на рисунке 3 приложения 1.

2.4. Разработка диаграммы прецедентов

Разработка диаграммы прецедентов является одним из ключевых этапов анализа и проектирования системы, особенно в методологии разработки программного обеспечения. Диаграмма прецедентов позволяет идентифицировать и описать функциональные возможности системы, включая взаимодействие между пользователями и системой. Диаграмма прецедентов представляет собой графическое представление взаимодействия

пользователей с системой через выполнение различных прецедентов. Она отражает функциональные требования системы и ее возможности.

Диаграмма прецедентов помогает разработчикам и заинтересованным сторонам понять функциональные требования системы, определить основные актеры и их роли, а также определить взаимодействие между актерами и прецедентами. Она служит основой для дальнейшего проектирования и разработки системы, а также для проверки соответствия системы требованиям пользователей. На рисунке 4 в приложении 1 представлена диаграмма прецедентов для пользователя системы.

На диаграмме представлены действия, которые могут выполняться пользователями. Эти действия описывают функциональность системы и включают в себя последовательность шагов, выполняемых пользователем. Функциональность пользователя направлена на оформление заявки на покупку билета.

2.5. Разработка и реализация приложения

Приложение «Транспортная компания», созданное в рамках данной курсовой работы, включает в себя несколько компонентов, обеспечивающих удобное оформление заявок на покупку билетов. Первым компонентом приложения является экран списка маршрутов. Здесь пользователь может просматривать все доступные маршруты и ознакомиться с подробностями о заинтересованном маршруте с помощью перехода на страницу информации о маршруте.

Следующим компонентом является экран списка пользователей, на нём пользователь может перейти на экран добавления пользователя или экран со списком маршрутов выбранного пользователя. После появляется возможность

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

оформления заявки на покупку билета для выбранного из списка маршрута на соответствующих экранах.

На экране оформленных заявок пользователю предоставляется возможность просмотра существующей заявки и добавления новой. При добавлении необходимо выбрать маршрут и дату его оказания, после чего заявка перейдёт в статус «Ожидание», а на телефон придёт уведомление об оставленной заявке.

Таким образом, приложение «Транспортная компания» представляет удобный интерфейс для ознакомления с маршрутами и последующего оформления заявок на покупку билетов. Функционал приложения обеспечивает эффективное взаимодействие с пользователями, соответствуя их потребностям при покупке билетов.

2.6. Тестирование приложения

Целью проведения тестирования является подтверждение реализации требуемой функциональности системы. В процессе тестирования, результат каждого теста фиксируется. Считается, что тест прошёл успешно в случае, если результат совпадает с ожидаемым результатом. Если результат отличается от ожидаемого, он фиксируется в протоколе тестирования. В таблице 1 приведена методика тестирования разработанного программного продукта.

Таблица 1 - Методика тестирования разработанной программы

Выполненное действие	Полученный результат
Запуск приложения	Отображение стартовой страницы

Нажатие на кнопку маршрута «подробно» из списка	Переход на экран с информацией о маршруте
Свайп влево	Возврат к предыдущему экрану
Нажатие на кнопку у администратора «Пользователи»	Отображение экрана со списком пользователей
Нажатие на кнопку «зарегистрироваться»	Отображение экрана с полями для ввода данных о пользователе
Повторное нажатие «зарегистрироваться»	В случае корректности введенных данных происходит проверка существует ли пользователь с указанным номером телефона в системе. Если не существует - добавление пользователя в список, возврат к предыдущему экрану с обновлённым списком. В другом случае показ пользователю уведомления о том, что введенный номер почты используется другим пользователем. В случае ввода некорректных данных показ пользователю уведомления что введенные данные неверны
Нажатие на пользователя из списка	Отображение экрана со списком маршрутов данного пользователя
Нажатие на кнопку «Добавить маршрут»	Отображение экрана с полями для ввода данных о маршруте пользователя
Нажатие на маршрут из списка	Отображение экрана со списком заявок на покупку билетов для данного маршрута
Нажатие на кнопку «Купить» (Билет)	Уведомление об успешной покупке билета, просмотреть билет можно на экране профиля

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

3. Руководство программиста

В контексте разработки приложения «Транспортная компания», основным требованием являлось обеспечение эффективной работы с базой данных и визуального отображения этих данных. Для достижения этой цели были выполнены следующие пункты.

1) Создание базы данных: для обеспечения эффективной работы с данными была создана физическая модель базы данных. Разработана SQL-структура базы данных с использованием названия «transport.db». Создан класс для подключения и взаимодействия с базой данных, обеспечивающий необходимые операции чтения и записи.

2) Разработка интерфейса: имеющиеся таблицы базы данных были интегрированы в удобный пользовательский интерфейс. Параметры отображения элементов интерфейса были адаптированы для обеспечения максимального удобства пользователя.

3) Алгоритмы проверки входных данных: написаны алгоритмы для проверки введенных пользователем данных. Реализованы механизмы сопоставления входных данных с данными в базе для исключения возможных ошибок в работе приложения.

4) Основные подпрограммы: подключение к базе данных: разработаны функции для установления связи с базой данных, обеспечивающие безопасное подключение и отключение.

5) Валидация данных: созданы алгоритмы валидации входных данных, гарантируя корректность и безопасность обработки информации.

Программа является полностью функциональной, однако при желании может быть доработана. Например, можно добавить возможность создания напоминания при оформлении заявки.

Класс Ticket.kt

Ticket представляет собой билет, который содержит информацию о маршруте, пользователе, номере места и цене.

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

Класс TicketDto.kt

TicketDto представляет собой данные билета, включая информацию о маршруте, пользователе, номере места, цене, месте отправления, месте назначения, времени отправления и времени прибытия.

Класс Trip.kt

Trip представляет собой маршрут, который содержит информацию о номере маршрута, месте отправления, месте назначения, количестве занятых мест, времени отправления, времени прибытия и цене.

Класс User.kt

User представляет собой пользователя, который содержит информацию о логине, email, пароле и роли.

Класс AddTripActivity.kt

AddTripActivity представляет собой активность для добавления нового маршрута.

Методы класса:

onCreate - Инициализирует активность и устанавливает обработчики для добавления нового маршрута.

Класс AdminActivity.kt

AdminActivity представляет собой активность для администратора, которая позволяет просматривать пользователей, билеты и добавлять новые маршруты.

Методы класса:

onCreate - Инициализирует активность и устанавливает обработчики для просмотра пользователей, билетов и добавления новых маршрутов.

showUsers - Отображает список пользователей.

showTickets - Отображает список билетов.

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

addTrip - Открывает активность для добавления нового маршрута.

updateUserRole - Обновляет роль пользователя и отображает уведомление.

Класс DbHelper.kt

DbHelper представляет собой помощник базы данных, который управляет операциями с базой данных, такими как добавление, обновление и получение данных.

Методы класса:

onCreate - Создает таблицы в базе данных.

onUpgrade - Обновляет таблицы в базе данных.

addUser - Добавляет нового пользователя в базу данных.

getUser - Проверяет наличие пользователя в базе данных по логину и паролю.

addTicket - Добавляет новый билет в базу данных.

getCurrentUserId - Получает идентификатор текущего пользователя по логину.

updateOccupiedSeats - Обновляет количество занятых мест для маршрута.

addTrip - Добавляет новый маршрут в базу данных.

getAllTrips - Получает все маршруты из базы данных.

searchTrips - Ищет маршруты по месту отправления и месту назначения.

getUserTickets - Получает все билеты пользователя.

getUserRole - Получает роль пользователя по логину.

getAllUsers - Получает всех пользователей из базы данных.

getAllTickets - Получает все билеты из базы данных.

updateUserRole - Обновляет роль пользователя.

getTripById - Получает маршрут по идентификатору.

updateTrip - Обновляет информацию о маршруте.

getUserTicketsForToday - Получает билеты пользователя на сегодня.

getLastPurchasedTicket - Получает последний купленный билет пользователя.

					МИВлГУ 09.03.04 – 0.010	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

getOtherPurchasedTickets - Получает остальные купленные билеты пользователя.

hashPassword - Хэширует пароль пользователя.

Класс LastTicketFragment.kt

LastTicketFragment представляет собой фрагмент, который отображает последний купленный билет пользователя.

Методы класса:

onCreateView - Создает представление фрагмента и отображает последний купленный билет.

Класс LoginActivity.kt

Описание класса: Класс LoginActivity представляет собой активность для входа пользователя в систему.

Методы класса:

onCreate - Инициализирует активность и устанавливает обработчики для входа и регистрации.

Класс OtherTicketsFragment.kt

OtherTicketsFragment представляет собой фрагмент, который отображает остальные купленные билеты пользователя.

Методы класса:

onCreateView - Создает представление фрагмента и отображает остальные купленные билеты.

Класс ProfileActivity.kt

ProfileActivity представляет собой активность для отображения профиля пользователя и его билетов.

Методы класса:

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

onCreate - Инициализирует активность и устанавливает адаптер для отображения билетов пользователя.

Класс SignupActivity.kt

SignupActivity представляет собой активность для регистрации нового пользователя.

Методы класса:

onCreate - Инициализирует активность и устанавливает обработчики для регистрации и входа.

Класс TicketsAdapter.kt

TicketsAdapter представляет собой адаптер для отображения списка билетов в RecyclerView.

Методы класса:

onCreateViewHolder - Создает новый ViewHolder для отображения элемента списка.

getItemCount - Возвращает количество элементов в списке.

onBindViewHolder - Привязывает данные билета к ViewHolder.

Класс TodayTicketsFragment.kt

TodayTicketsFragment представляет собой фрагмент, который отображает билеты пользователя на сегодня.

Методы класса:

onCreateView - Создает представление фрагмента и отображает билеты на сегодня.

Класс TripActivity.kt

TripActivity представляет собой активность для отображения информации о маршруте.

Методы класса:

onCreate - Инициализирует активность и отображает информацию о маршруте.

Класс TripsActivity.kt

TripsActivity представляет собой активность для отображения списка маршрутов и поиска маршрутов.

Методы класса:

onCreate - Инициализирует активность и устанавливает адаптер для отображения списка маршрутов.

Класс TripsAdapter.kt

TripsAdapter представляет собой адаптер для отображения списка маршрутов в RecyclerView.

Методы класса:

onCreateViewHolder - Создает новый ViewHolder для отображения элемента списка.

getItemCount - Возвращает количество элементов в списке.

onBindViewHolder - Привязывает данные маршрута к ViewHolder.

Класс UsersAdapter.kt

UsersAdapter представляет собой адаптер для отображения списка пользователей в RecyclerView.

Методы класса:

onCreateViewHolder - Создает новый ViewHolder для отображения элемента списка.

getItemCount - Возвращает количество элементов в списке.

onBindViewHolder - Привязывает данные пользователя к ViewHolder.

Класс ViewPagerAdapter.kt

ViewPagerAdapter представляет собой адаптер для ViewPager, который управляет фрагментами для отображения билетов пользователя.

Методы класса:

getItemCount - Возвращает количество фрагментов.

createFragment - Создает фрагмент для отображения в ViewPager.

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

4. Руководство пользователя

Рекомендованные системные требования к мобильным устройствам на платформе Android:

1. Операционная система: Android;
2. Процессор: Рекомендуется устройство с процессором не ниже среднего уровня;
3. Оперативная память: Рекомендуется устройство с оперативной памятью не менее 2 ГБ;
4. Место на устройстве: Наличие свободного места на устройстве не менее 100 МБ;
5. Экран: Рекомендуется устройство с экраном не менее 5 дюймов и разрешением не менее 720x1280 пикселей.

Использование основных функций:

- 1) Просмотр подробной информации о маршруте: перейти в раздел «Посмотреть список маршрутов» и выбрать интересующий маршрут из списка.
- 2) Оформление заявки на покупку билета: перейти в раздел «Пользователи», выбрать пользователя, выбрать маршрут пользователя, перейти в раздел «Запросить билет», выбрать маршрут для покупки, выбрать дату для покупки билета, подтвердить запрос билета.
- 3) Выход из приложения: для выхода из приложения используйте стандартные функции управления задачами Android.

Приложение располагается локально на устройстве пользователя, для доступа к приложению не требуется доступ в интернет. Работа с приложением и основной функционал:

Пользователь

Запуск приложения производится нажатием на иконку приложения. При запуске приложения перед пользователем отывается главная форма

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

приложения, представленная на рисунке 1 в приложении 3. На ней расположены две кнопки: «Sign In», «You aren't here. Sign up», нажатие на каждую из кнопок переводит пользователя на формы, соответствующие названиям кнопок. Для авторизации необходимо пользователю ввести логин и пароль, после нажать на кнопку «Sign In». В ином случае, если пользователь не авторизован необходимо на форме нажать на надпись «You aren't here. Sign up», после чего появится форма регистрации (рисунке 2 в приложении 3), в которой нужно ввести поля: email, login, password. После необходимо вернуться на форму авторизации и войти в систему

При успешной авторизации пользователь попадает на форму «Список маршрутов транспортной компании», представленная на рисунке 3 в приложении 3, отображает список доступных маршрутов. При нажатии на маршрут произойдёт переход на страницу с информацией о маршруте, представленной на рисунке 4 приложения 3.

При выборе интересующего маршрута пользователь может приобрести билет на данный маршрут, купив его, нажав на кнопку «Купить». После нажатия информация о билете перейдет в профиль, в который можно перейти на форме «Список маршрутов транспортной компании», кнопка «Профиль», сам профиль предоставлен на рисунке 5,6 приложения 3.

Администратор

Запуск приложения производится нажатием на иконку приложения. При запуске приложения перед пользователем отывается главная форма приложения, представленная на рисунке 1 в приложении 3. На ней расположены две кнопки: «Войти», «Нет аккаунта? Зарегистрировать», нажатие на каждую из кнопок переводит пользователя на формы, соответствующие названиям кнопок. Для авторизации необходимо пользователю ввести логин и пароль, после нажать на кнопку «Sign In». В ином

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

случае, если пользователь не авторизован необходимо на форме нажать на надпись «Нет аккаунта? Зарегистрировать», после чего появится форма регистрации (рисунке 2 в приложении 3), в которой нужно ввести поля: email, login, password. После необходимо связаться с главным администратором для выдачи роли «администратор», чтобы войти в приложение с ролью администратора.

Форма «Админ панель», представленная на рисунке 7 в приложении 3, отображает список пользователей приложения. На форме доступен весь функционал для администратора кнопки: «Пользователи», «Билеты», «Поездки», «Добавить поездку»

На форме при нажатии на кнопку «Пользователи» можно получить список всех пользователей в системе «Сделать админом» или «Сделать пользователем» выбранному пользователю присвоится новая роль, в зависимости от выбора действия. Форма представлена на рисунке 8 в приложении 3

Нажатие на кнопку «Билеты», представленная на рисунке 10 в приложении 3, отображает список всех билетов в системе. При нажатии на кнопку «Добавить маршрут» появится форма добавления маршрута, представленная на рисунке 9 в приложении 3. При нажатии на кнопку «Маршрут» появится список всех маршрутов в системе на рисунке 11 в приложении 3, при нажатии на «Подробнее» из списка произойдёт переход на форму с информацией о занятых билетах для выбранного маршрута, представленную на рисунке 12 в приложении 3.

Заключение

Результатом выполнения курсовой работы по разработке приложений для мобильных операционных систем является приложение «Транспортная компания», выполненное с использованием среды разработки Android Studio и языка программирования Kotlin. В ходе данной работы были составлены концептуальная, логическая, физическая модели, создано приложение, отвечающее тем требованиям, которые были заложены перед его созданием.

В ходе разработки приложения в Android Studio были успешно пройдены этапы создания пользовательских интерфейсов для взаимодействия с пользователем, реализации логики взаимодействия с базой данных, а также обработки событий в приложении. Дальнейшее развитие приложения может включать в себя расширение функциональности, оптимизацию пользовательского интерфейса, адаптацию под новые стандарты и технологии в области мобильной разработки.

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

Список литературы:

1. Аллен Г. Android. Программирование приложений для смартфонов и планшетов — СПб.: Питер, 2019. — 480 с.
2. Бурд Б. Android. Разработка приложений для чайников — СПб.: Питер, 2017. — 416 с.
3. Гриффитс Дэвид, Гриффитс Дон Head First. Программирование для Android. 2-е изд. — СПб.: Питер, 2018. — 912 с.
4. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е изд. — СПб.: Питер, 2016.
5. Колисниченко Д.Н. Программирование для Android 5. Самоучитель. — СПб.: БХВ-Петербург, 2015. — 303 с.
6. Климов И. Android. Разработка приложений на Java — СПб.: БХВ-Петербург, 2018. — 464 с.
7. Соломатин А. Android. Программирование приложений — М.: Издательский дом Вильямс, 2015. — 416 с.
8. Чернышев В.Н. Android. Разработка профессиональных мобильных приложений — М.: Издательство Эксмо, 2017. — 352 с.

Приложение 1.

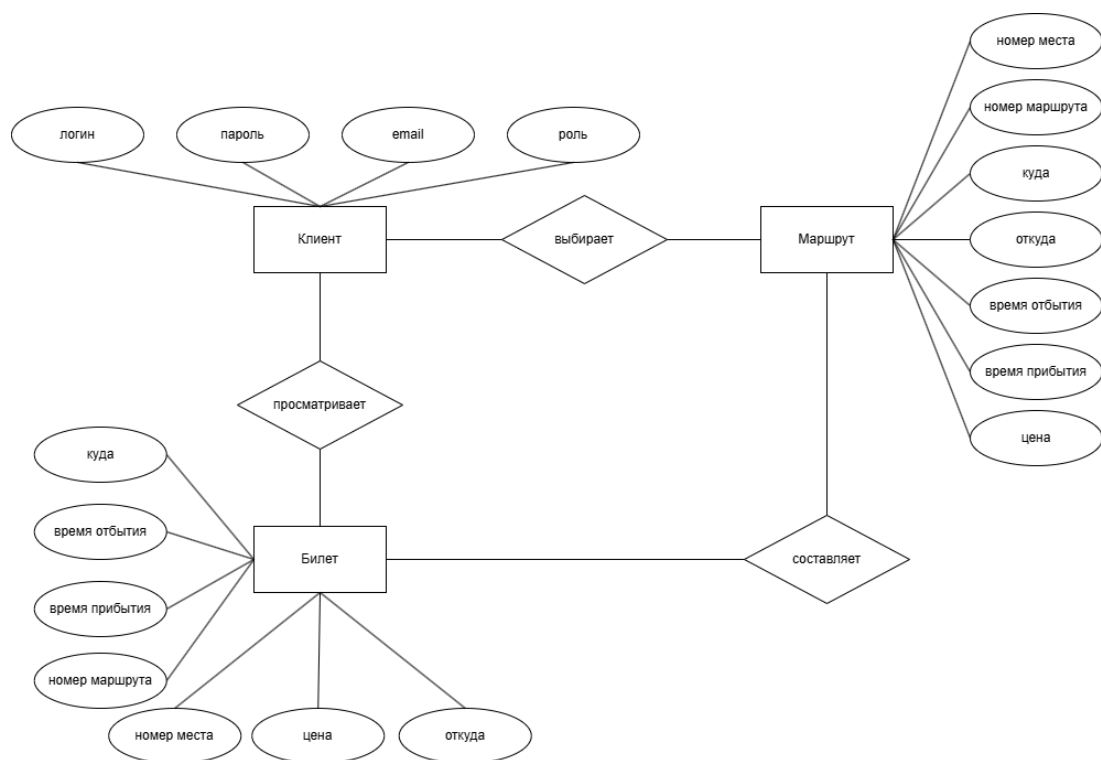


Рисунок 1 - Концептуальная модель данных

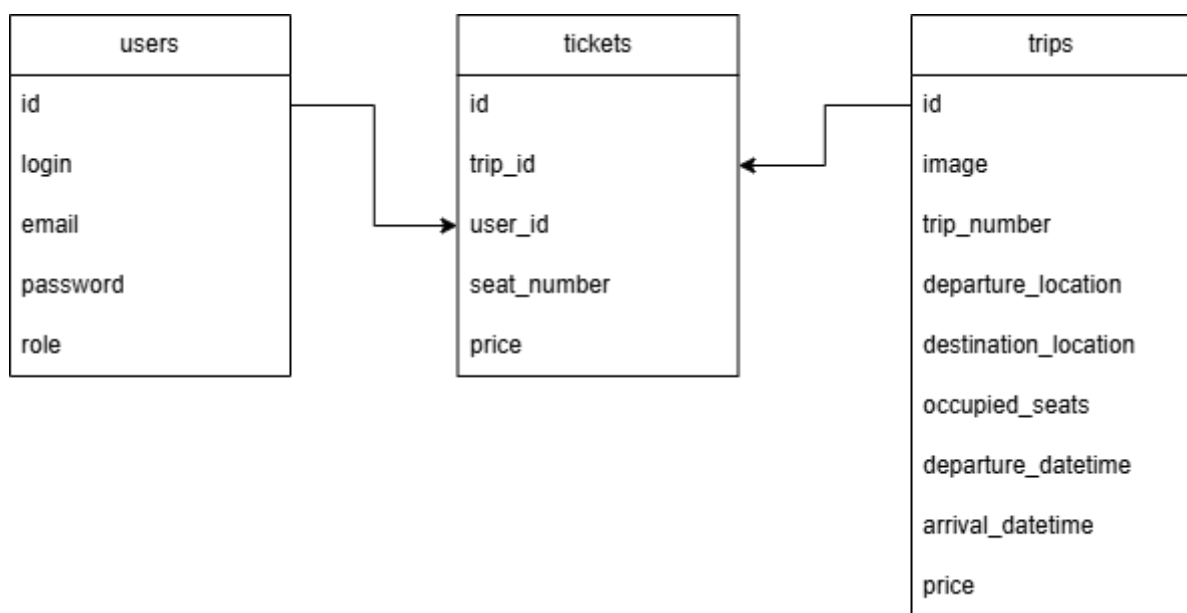


Рисунок 2 – Логическая модель данных

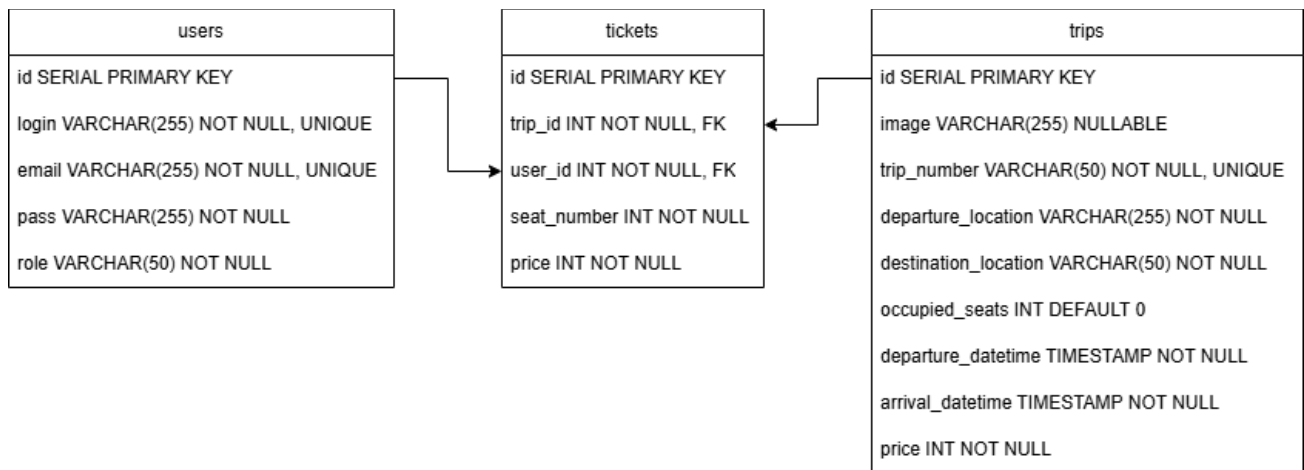


Рисунок 3 – Физическая модель данных

Приложение 2

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

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

Приложение 3

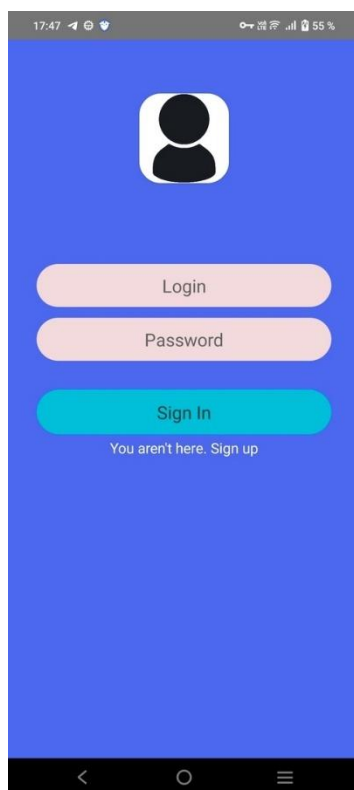


Рисунок 1 – Экран авторизации

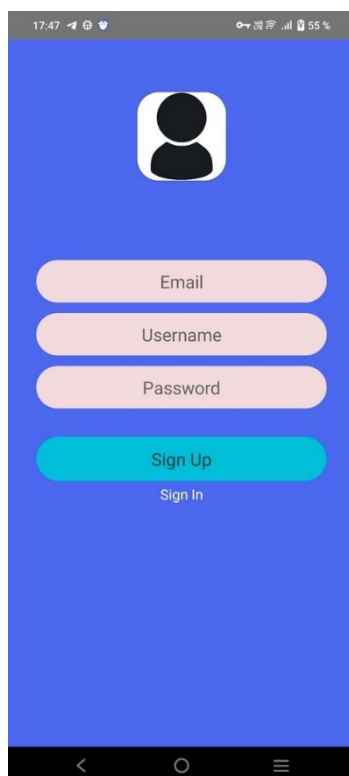


Рисунок 2 – Экран регистрации

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

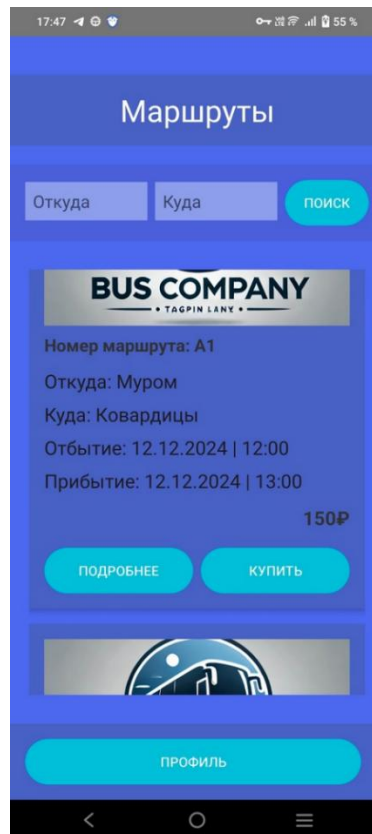


Рисунок 3 – Главный экран со списком маршрутов

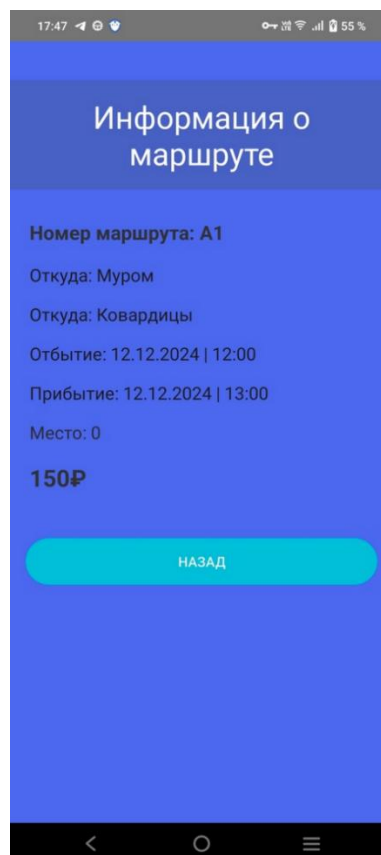


Рисунок 4 – Экран просмотра маршрута

					МИВлГУ 09.03.04 – 0.010	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

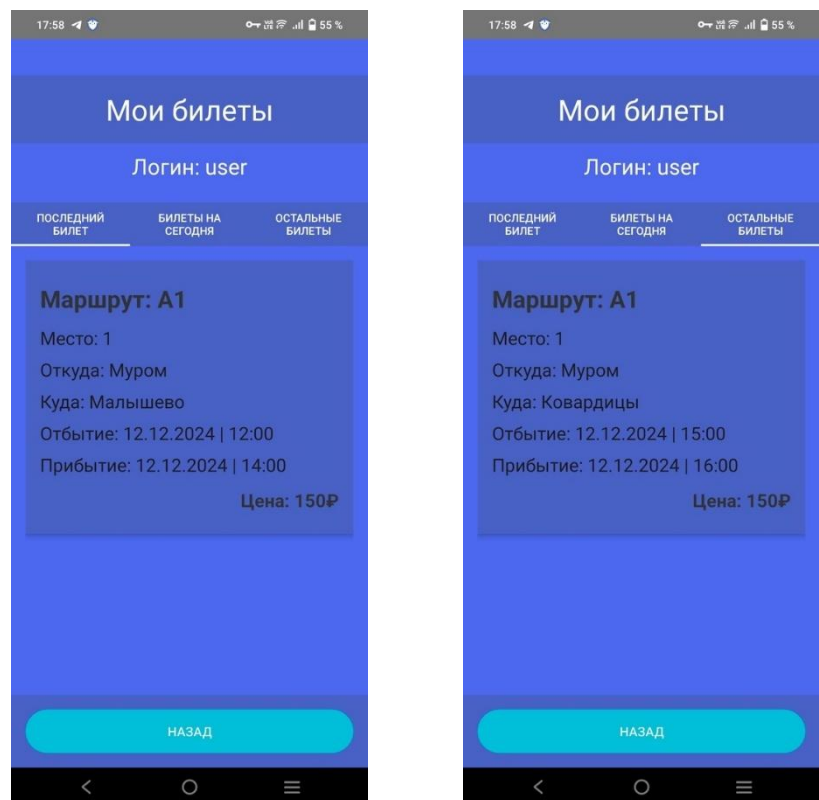


Рисунок 5, 6 – Профиль пользователя

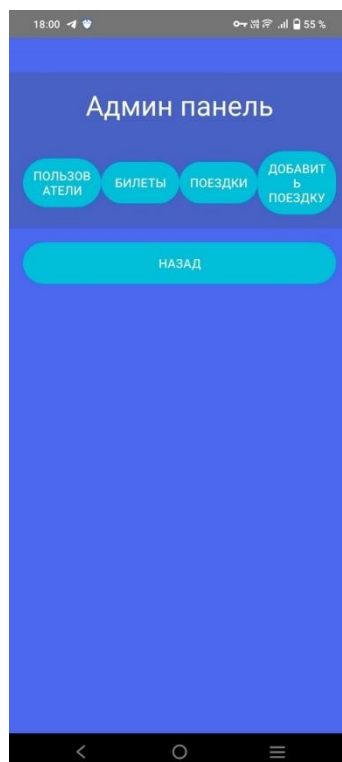


Рисунок 7 – Панель админа

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

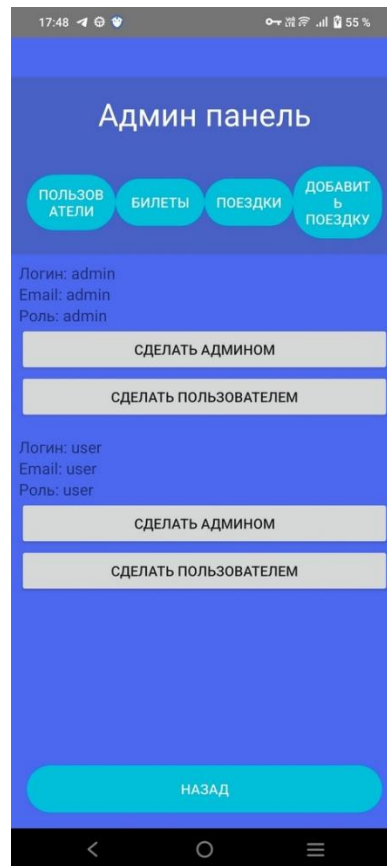


Рисунок 8 – Список всех пользователей

Номер маршрута
Место отправления
Место назначения
Время отправления
Время прибытия
Цена
ДОБАВИТЬ ПОЕЗДКУ

Рисунок 9 – форма добавления маршрута

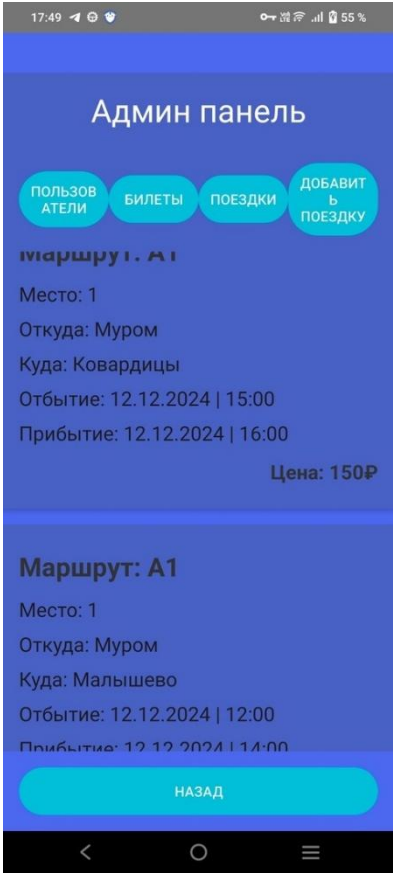


Рисунок 10 – Список всех купленных билетов

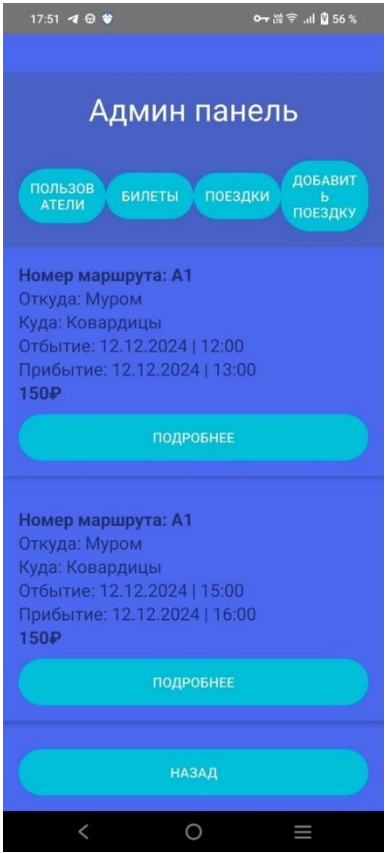


Рисунок 11 – Список маршрутов

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

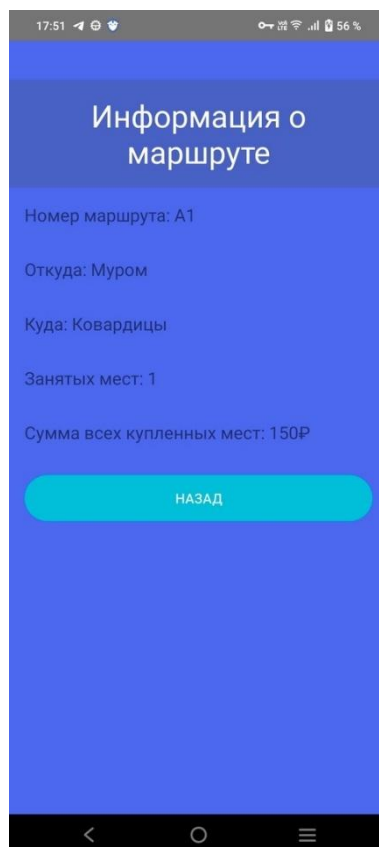


Рисунок 12 – Подробности о маршруте

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