

МИНОБРАЗОВАНИЯ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Расписание занятий для вузов «UnivTimetable»

Курсовой проект

09.03.02 Информационные системы и технологии
Программная инженерия в информационных системах

Зав. кафедрой _____ С. Д. Махортов, д.ф.- м.н., доцент
__._.20__

Обучающийся _____ Ф.Ю. Свиридов, 3 курс, д/о

Обучающийся _____ И.М. Молдавский, 3 курс, д/о

Обучающийся _____ П.Р. Сергеевич, 3 курс, д/о

Руководитель _____ В.С. Тарасов, ст. преподаватель

Руководитель _____ И.В. Клейменов, ассистент

Воронеж 2023

Содержание

Введение.....	3
1 Постановка задачи.....	4
1.1 Задачи проекта.....	4
2 Анализ предметной области	5
2.1 Терминология	5
2.2 Обзор аналогов	7
2.2.1 СтудЖурнал – Расписание занятий	7
2.2.2 Student Calendar – Timetable	9
2.2.3 School Planner.....	11
2.3 Средства реализации	15
3 Реализация	17
4 Интерфейс приложения	22
Заключение.....	28
Список использованных источников.....	29

Введение

Существует много разных способов составления и представления расписания учебных занятий в образовательных организациях, как в школах, так и в высших учебных заведениях: некоторые учреждения уже полностью перешли на современные решения в виде электронных журналов и других электронных информационных систем доступных благодаря Интернету, остальные же до сих пор прибегают к полностью ручному составлению расписания, а представляют его чаще всего в виде больших распечатанных листов, склеенных вместе. Так же присутствуют варианты реализации в виде использования электронных таблиц, что в какой-то степени облегчает работу, раннее исполняемую вручную. Подобные способы требуют слишком больших временных затрат и нередко могут приводить к ошибкам и несостыковкам в самом расписании. Именно поэтому практически ни в одном ВУЗе нет такой практики, что точное и окончательное расписание известно студентам и преподавателям уже в первую неделю учебных занятий, а не формируется постепенно в течении первых нескольких недель учебного периода.

Затрагивая тему современных реалий, абсолютно у каждого обучающегося и каждого члена преподавательского состава имеются свои мобильные устройства, что значительно упрощает многие повседневные вещи. При наличии мобильного приложения с расписанием, распорядок учебного дня всегда будет под рукой и необходимость спрашивать об этом самих преподавателей отпадёт. К тому же, если у подобного приложения будет функционал, реализующий не только просмотр расписания, но и его формирование на основе предоставленных самими преподавателями данных, тогда сам процесс составления значительно ускорится и облегчится. Для достижения поставленных целей для начала рассмотрим несколько популярных аналогов и затем перейдем к формулировке требований, на основе которых будет создаваться приложение.

1 Постановка задачи

1.1 Задачи проекта

Целью данной курсовой работы является создание мобильного приложения для отслеживания и формирования расписания учебных занятий в ВУЗах, которое будет реализовывать следующие требования:

- предоставление преподавателями нежелательного времени для проведения занятий;
- обеспечение возможности просмотра расписания на заданный промежуток времени как для старост (для их групп), так и для преподавателей (личное);
- предоставление преподавателям права переноса занятий через приложение;
- формирование расписания системой на основе предоставленных преподавателями данных;
- экспорт расписания в формате Excel таблицы;
- уведомление старост по электронной почте об изменениях в расписании.

2 Анализ предметной области

2.1 Терминология

- Мобильное приложение — программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы (iOS, Android и т. д.);
- Android-приложение — программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для платформы Android;
- Android — это операционная система с открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux;
- Клиент — это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу;
- Сервер — выделенный или специализированный компьютер для выполнения сервисного программного обеспечения;
- База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД);
- SQL-запросы — это наборы команд для работы с реляционными базами данных;
- Дизайн-макет — это схематичное изображение финальной идеи с указанием всех деталей. В нем указываются концепция, шрифты, тексты, изображения, расположение всех элементов и общая картина продукта;
- Аутентификация — процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных;

- Авторизация — предоставление определённому лицу или группе лиц прав на выполнение определенных действий;
- Фреймворк — программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта;
- SQL-инъекция — внедрение в запрос произвольного SQL-кода, который может повредить данные, хранящиеся в БД или предоставить доступ к ним;
- Frontend — клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса;
- Backend — программно-аппаратная часть сервиса, отвечающая за функционирование его внутренней части;
- REST — архитектурный стиль взаимодействия компонентов распределённого приложения в сети;
- API — описание взаимодействия одной компьютерной программы с другой;
- Аккаунт или учётная запись — это персональная страница пользователя или личный кабинет, который создается после регистрации;
- Пользователь — человек, использующий приложение;
- Преподаватель — работник высшего учебного заведения, преподающий какой-либо предмет(ы);
- Староста — человек из числа студентов, ответственный за организационные моменты, связанные с учебной деятельностью своей группы, в том числе за связь с преподавателями и своевременное уведомление одногруппников обо всех изменениях в расписании занятий;

- Администратор — человек, ответственный за обеспечение штатной работы приложения путём добавления/редактирования/удаления ВУЗов и пользователей;
- Расписание — это основной учебно-методический документ, определяющий педагогически целесообразную последовательность учебных занятий в образовательном учреждении на каждый день учебной недели и конкретизирующий таким образом учебный план.

2.2 Обзор аналогов

В данном параграфе будут рассмотрены аналоги приложения “UnivTimetable”, а также приведены их преимущества и недостатки.

2.2.1 СтудЖурнал – Расписание занятий

СтудЖурнал - приложение, которое позволяет более эффективно управлять своим учебным графиком. СтудЖурнал позволяет быстро организовать персональное расписание, следить за календарным графиком и записывать заметки, лабораторные работы, домашние задания.

С помощью приложения можно эффективно организовывать студенческую жизнь, моментально делиться учебными материалами с друзьями.

С интерфейсом приложения можно ознакомиться на рисунках 1-2.

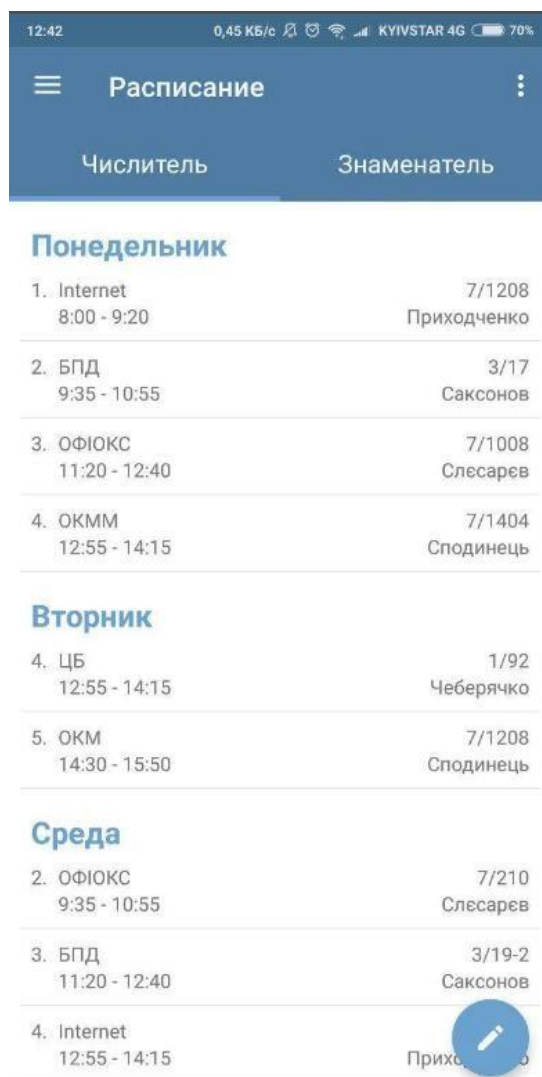


Рисунок 1 - Интерфейс вкладки расписание

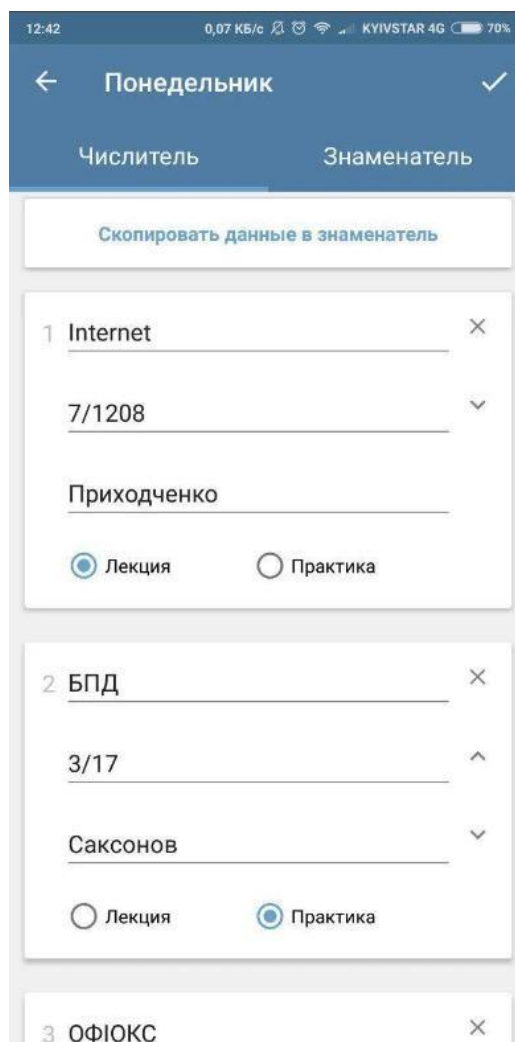


Рисунок 2 - Интерфейс вкладки редактирования

Преимущества:

- удобное создание заметок, конспектов;
- календарное отображение информации;
- можно экспортировать данные и так же импортировать себе.

Недостатки:

- нет интеграции со своим вузом, то есть придется вводить предметы и их время для каждого дня недели самостоятельно;
- отсутствие возможности экспортировать расписание в Excel-формат.

2.2.2 Student Calendar – Timetable

Student Calendar – приложение, которое создано для учащихся, чтобы помочь учащимся организовать и повысить успеваемость.

Цель использования этого приложения – выполнять задачи в совмещенные сроки, лучше распределять время между академической и личной жизнью, вести повседневную жизнь больше спокойствия и меньше стресса. В студенческом календаре важная информация о тестах, домашних заданиях, встречах и расписании всегда будет доступна на вашем смартфоне для проверок и новых расписаний, где бы вы ни находились. Есть также напоминания (с будильниками и уведомлениями), которые помогут вам не забыть о важных действиях.

Календарь учащихся перечисляет события в виде списка дел или контрольного списка, где события должны быть помечены как завершённые, чтобы они больше не выделялись. Кроме того, он группируется по прошлым и будущим событиям, и можно увидеть, когда какая-то активность задерживается. Интерфейс продемонстрирован на рисунках 3-4.



Рисунок 3 - Интерфейс вкладки расписание

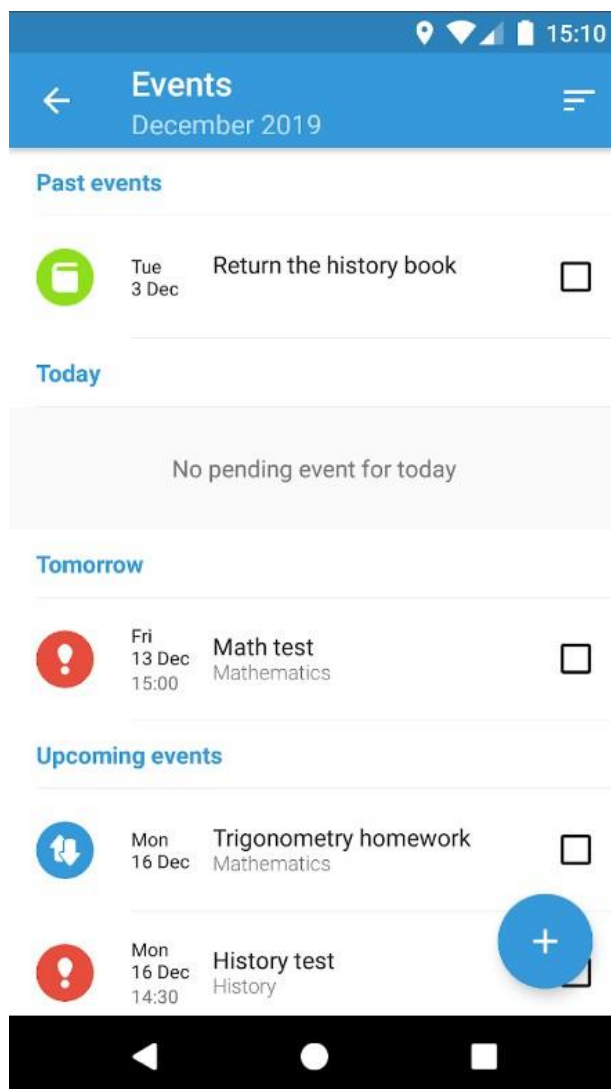


Рисунок 4 - Интерфейс вкладки создания события

Преимущества:

- планирование мероприятий;
- уведомления о приближающихся мероприятиях;
- управление оценками.

Недостатки:

- отсутствие русского языка;
- необходимо самостоятельно заполнять информацию о занятиях;
- редкие обновления приложения;
- невозможно экспортировать расписание в Excel-формат.

2.2.3 School Planner

School Planner — это удобная программа для учащихся всех возрастов, помогающая организовать учебный процесс и иметь все нужное под рукой. Независимо от места учебы эта программа будет очень полезна.

Данное приложение предоставляет удобную запись домашних заданий и материалов для экзаменов, а ежедневные уведомления помогут никогда ничего не забывать.

Встроенный календарь оптимизирован для нужд учащихся и позволяет управлять событиями и запланированными делами более легко. Благодаря приложению график и ежедневное расписание будет всегда под рукой, а это облегчит планирование учебного процесса. Расписание полностью настраиваемо: можно назначать различные цвета для каждого объекта и просматривать события, сохраненные в календаре.

Интерфейс приложения продемонстрирован на рисунках 5-6.



Рисунок 5 - Интерфейс вкладки расписание

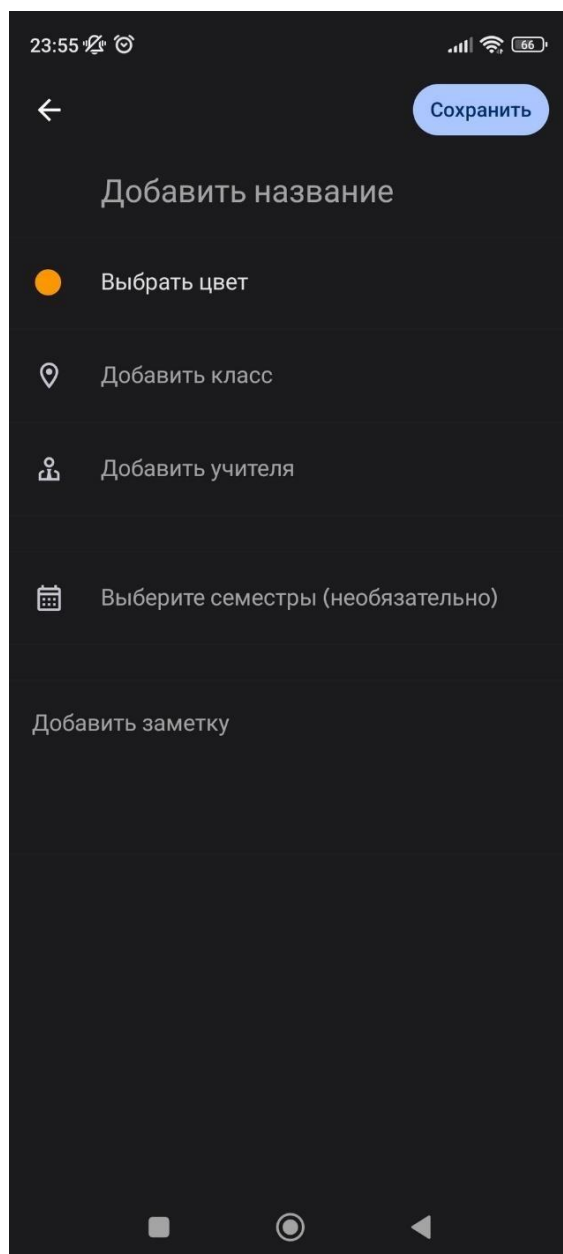


Рисунок 6 - Интерфейс вкладки добавления предмета

Преимущества:

- расписание для экзаменов, домашних заданий;
- управление оценками, предметами;
- настройка темы приложения;
- возможность записывать разную контактную информацию о преподавателях.

Недостатки:

- большое количество рекламы;

- для того, чтобы полностью использовать все функции приложения, необходимо оформить платную подписку;
- нужно вручную добавлять информацию о занятиях на тот или иной промежуток времени;
- невозможно экспортировать расписание в Excel-формат.

2.3 Средства реализации

- Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems;
- Spring Boot Framework – это веб-платформа Java на основе микрослужб с открытым кодом от Spring. Обладает большим набором инструментов;
- PostgreSQL – свободная объектно-реляционная система управления базами данных. Выбрана так как, поддерживает большое количество типов данных, имеет богатую функциональность, поддерживается многими серверами;
- Docker – платформа, предназначенная для помощи разработчикам в создании, совместном использовании и запуске современных приложений;
- Flyway – средство обеспечения версионности схем и управления миграциями базы данных. FlyWay имеет поддержку PostgreSQL и интегрируется с Spring Framework;
- Android SDK – универсальное средство разработки мобильных приложений для операционной системы Android;
- Kotlin – средство обеспечения версионности схем и управления миграциями базы данных. FlyWay имеет поддержку PostgreSQL и интегрируется с Spring Framework;
- Miro – платформа для совместной работы распределенных команд;
- Swagger – фреймворк для спецификации REST API;

- Draw.io – бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания диаграмм, таких как блок-схемы, каркасы, диаграммы UML, организационные диаграммы и сетевые диаграммы;
- Figma – онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени;
- Trello – облачная программа для управления проектами небольших групп;
- Github – веб-сервис для хостинга IT-проектов и их совместной разработки;
- Git – распределённая система управления версиями.

3 Реализация

В результате анализа было принято решение о разработке системы, описание которой приводится в данной главе.

3.1 Структура базы данных

Структура базы данных приведена на рисунке 7.



Рисунок 7 - Схема таблиц базы данных

Для хранения информации о расписании занятий высших учебных заведений и относящихся к ним факультетов, групп студентов и пользователей используются следующие таблицы:

1. **University** – справочник со всеми ВУЗами, в которых используется приложение. Содержит полное официальное название ВУЗа и город, в котором он находится;
2. **Faculty** – список факультетов для каждого отдельного ВУЗа, содержит название факультета и id университета;
3. **University_group** – таблица со всеми группами, относящимися к конкретному факультету определённого ВУЗа. В ней представлена соответствующая информация о номере группы, номере курса, количестве студентов, id факультета и старосты;
4. **System_user** – непосредственно сами пользователи, представляющие собой преподавателей, старост и администраторов приложения. Вся необходимая информация о них отражена в виде колонок, содержащих город, email, ФИО, пароль, никнейм, роль, id ВУЗа, группы и факультета;
5. **Class** – справочник для проводимых учебных занятий. Имеет следующие поля: название дисциплины, время начала занятия, тип занятия (семинар/лекция), день недели, тип недели (числитель/знаменатель), id соответствующих аудитории, преподавателя и дня недели;
6. **Timetable** – перечисление всех существующих дней недели, в которые могут проводиться занятия;
7. **Audience** – аудитории, в которых проводятся учебные занятия. Описываются полями номера аудитории, вместимости, id ВУЗа и факультета;
8. **Group_class** – таблица, связывающая между собой занятия и группы для которых они проводятся.

Остальная часть таблиц используется для хранения информации о вспомогательных деталях, используемых при составлении расписания. В их число входят таблицы для оборудования в аудиториях, заявки для каждой отдельной учебной дисциплины, а также невозможное для проведения занятий преподавателями время. Ко всему перечисленному относятся следующие таблицы:

1. **Request** – заявка на включение предмета в расписание, либо перенос занятия по предмету. Содержит в себе название дисциплины, количество часов в неделю, тип занятия (семинар/лекция), id преподавателя и группы;
2. **Impossible_time** – используется для хранения информации о времени, в которое преподаватели по тем или иным причинам не могут присутствовать на занятиях и соответственно проводить их. Содержит в себе день недели и соответствующее время, связано с заявкой её id;
3. **Equipment** – справочник для всего доступного оборудования. Name – наименование оборудования на английском, displayName - на русском, name будет передаваться между слоями приложения, displayName – отображаться для конечного пользователя;
4. **Audience_equipment** – связывает между собой аудитории и оборудование в них;
5. **Request_equipment** – таблица, связывающая заявки по предметам и требуемое оборудование для проведения занятий.

3.2 Диаграмма состояния

Диаграмма состояния позволяет определить все возможные сценарии поведения системы, выделить её ключевые состояния и переходы между ними. Для данного проекта всего было спроектировано 3 диаграммы для состояний администратора, старосты и преподавателя соответственно. Данные диаграммы представлены на рисунке 8.

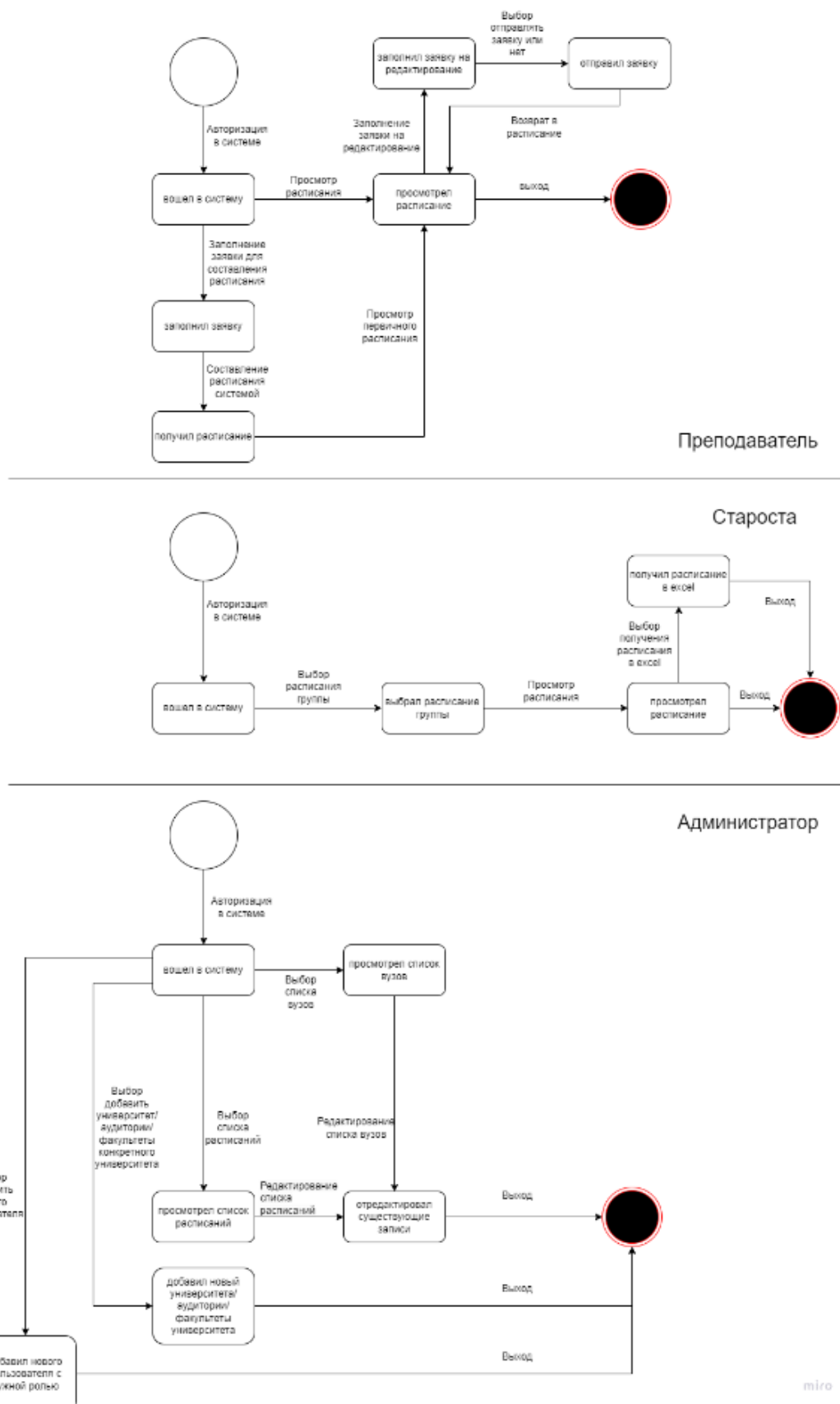


Рисунок 8 - Диаграмма состояний

3.3 Сценарии использования

Сценарии использования для всех групп пользователей приведён на рисунке 9.

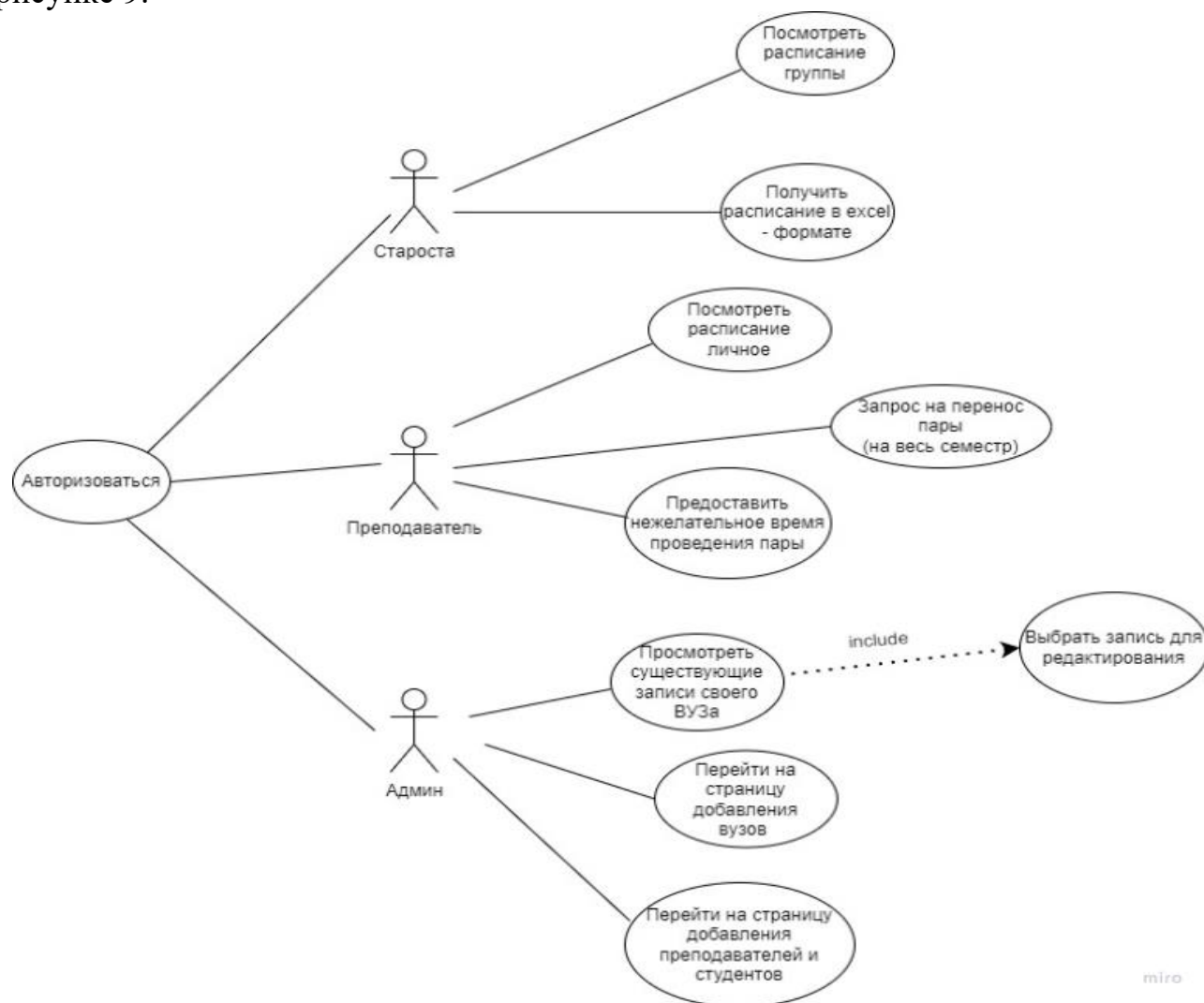
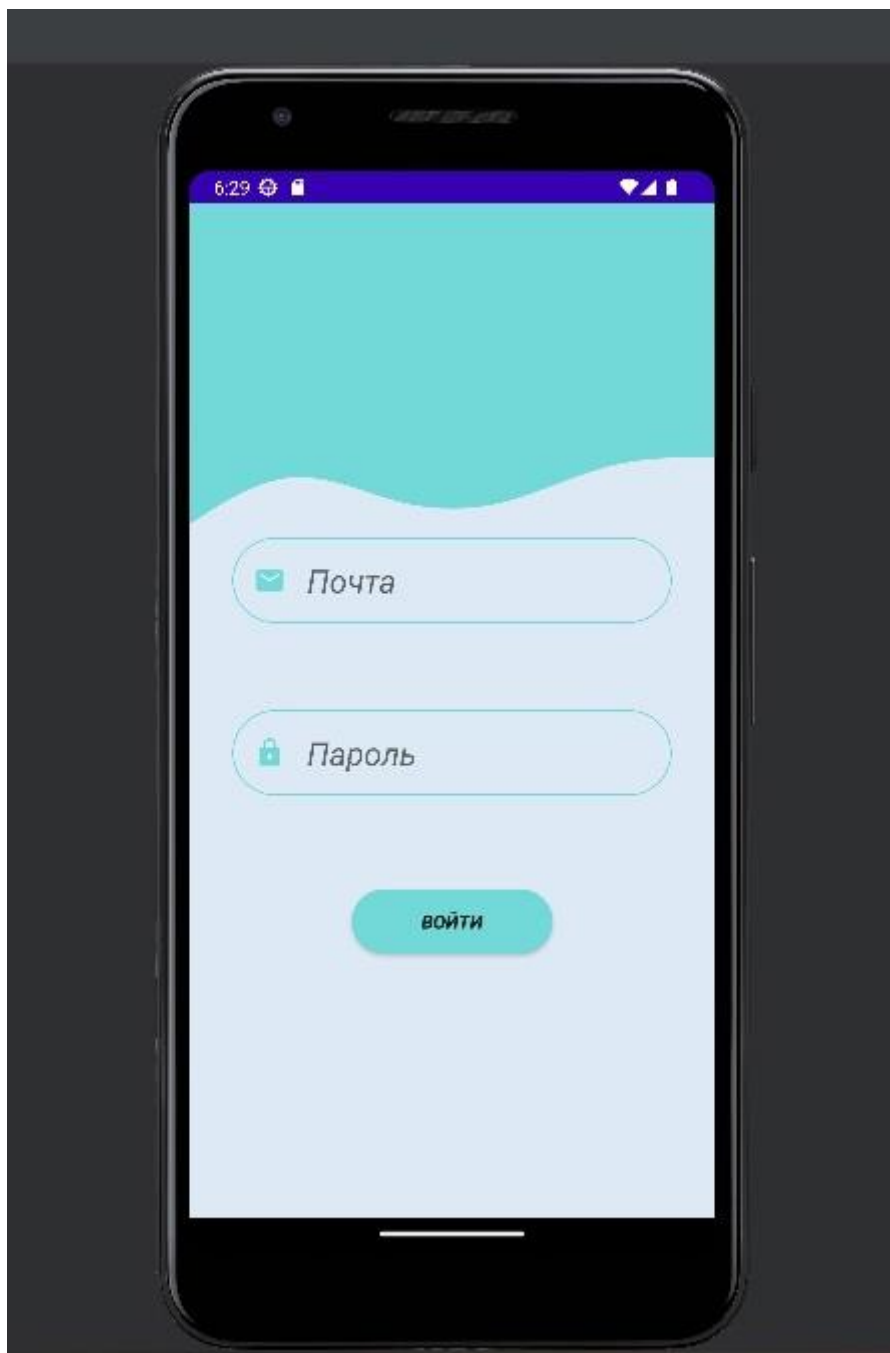
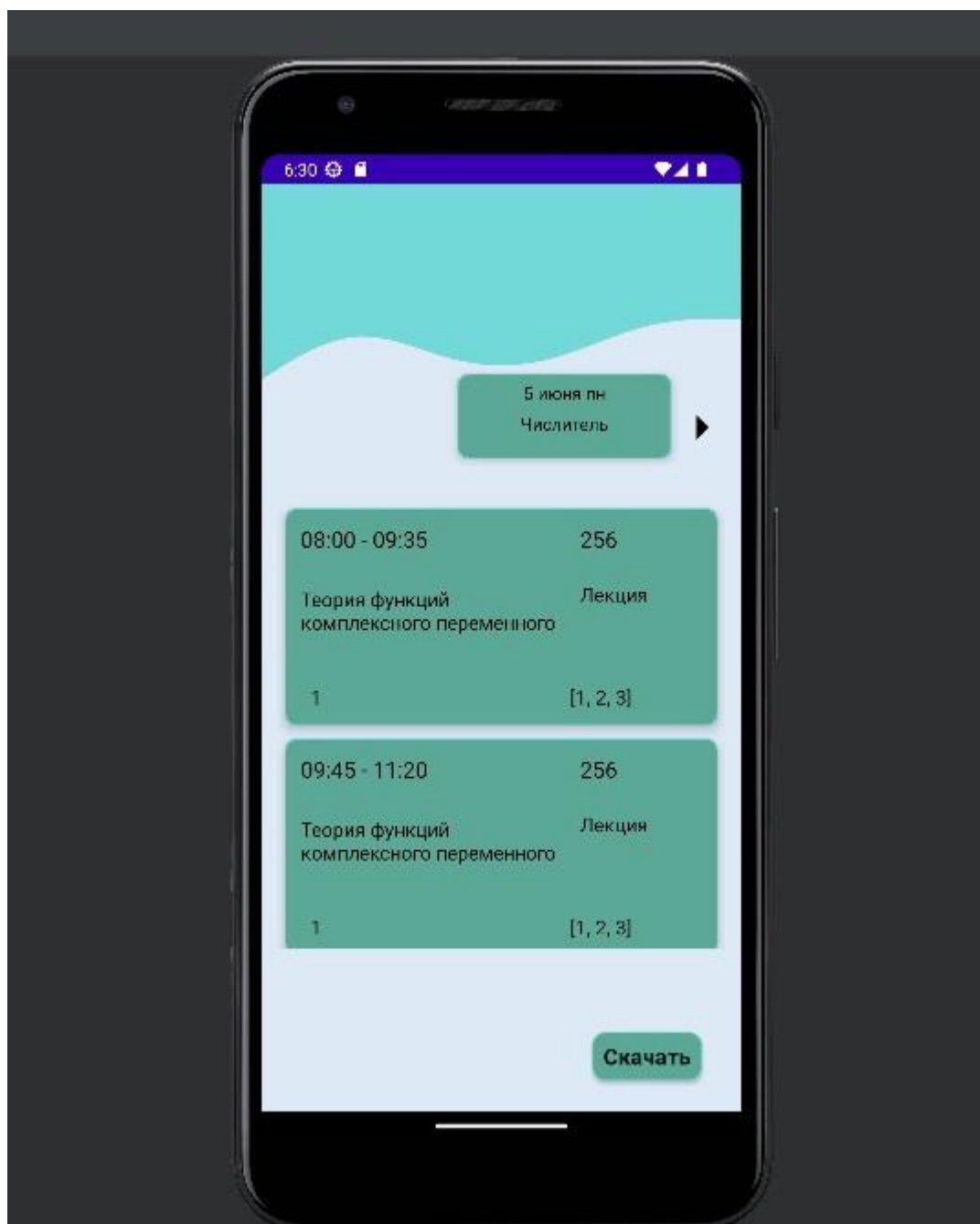


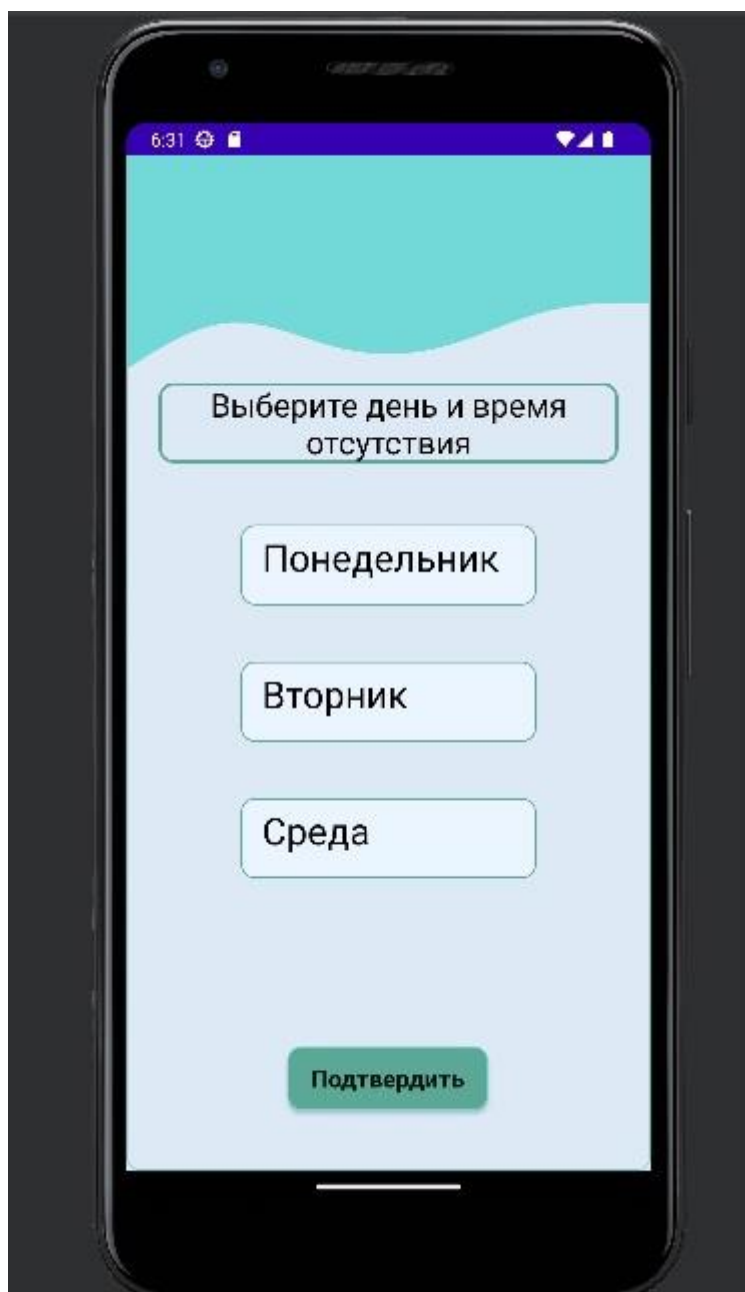
Рисунок 9 - Сценарии использования

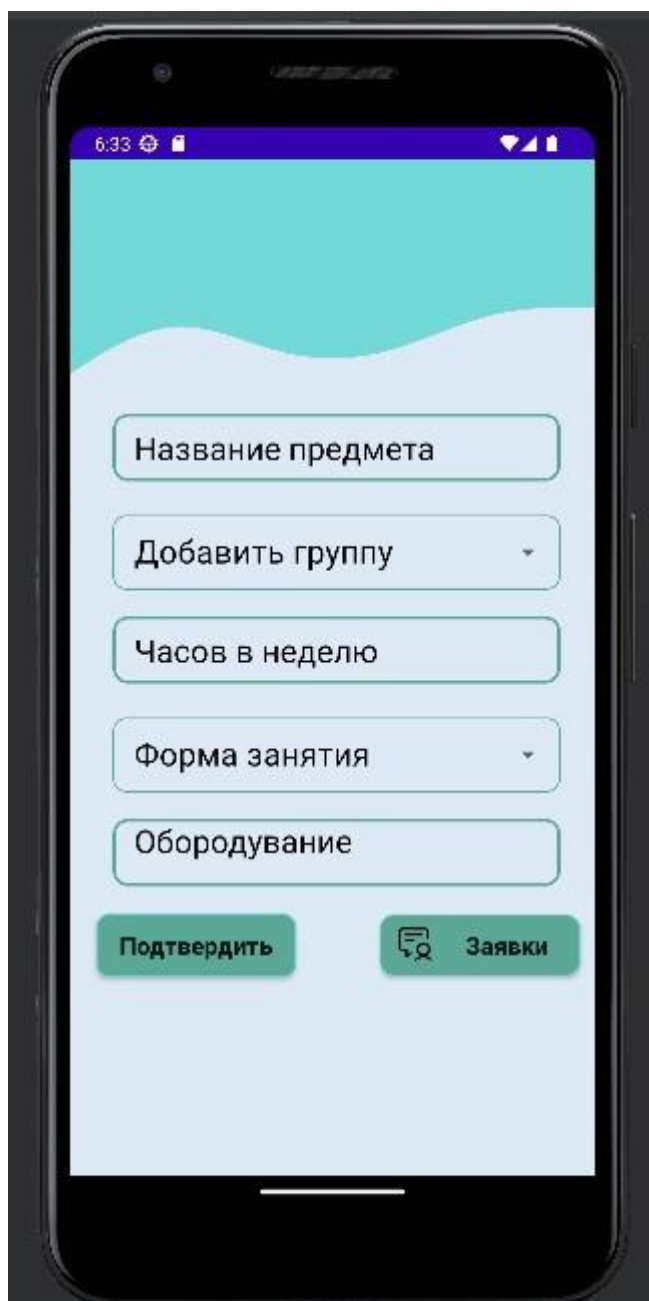
Рассмотрим полную диаграмму для использования приложения разными группами пользователей. В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Она может помочь в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определённых целей. Данная диаграмма также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы. Данная диаграмма представлена на рисунке 9.

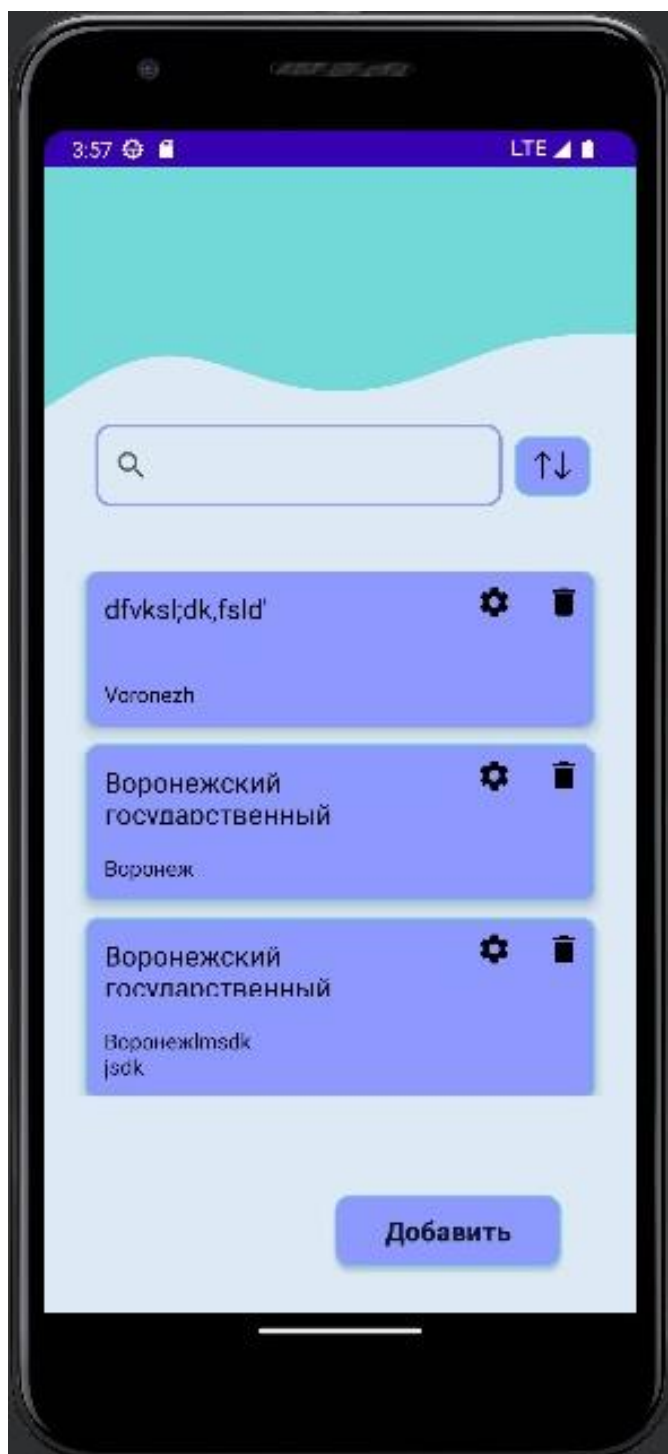
4 Интерфейс приложения

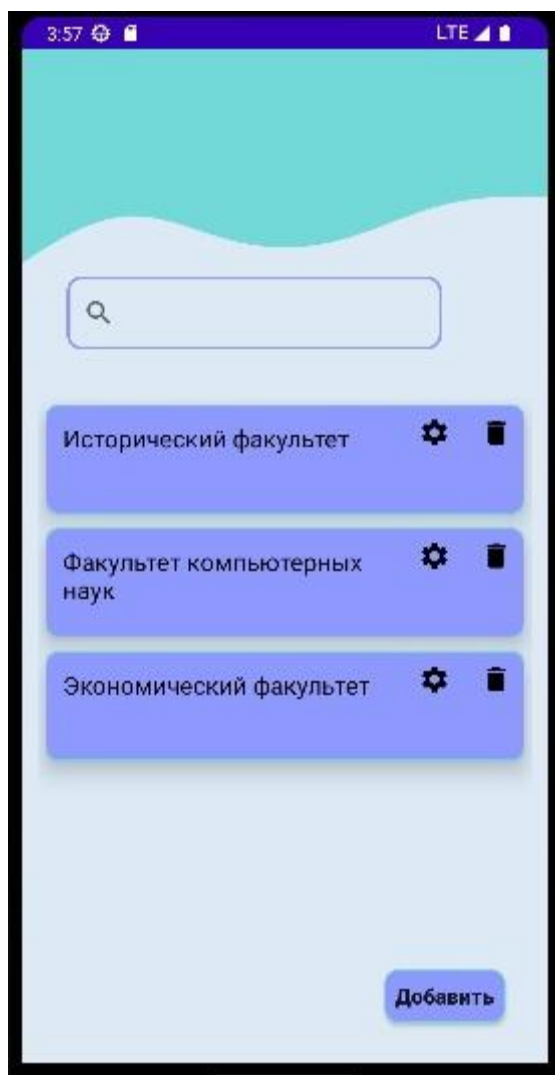












Заключение

При выполнении работы было рассмотрены аналогичные решения для отслеживания и формирования расписания, и, после их анализа и выделения недостатков, были выработаны основные идеи проекта, требования к функционалу и особенности, способные сделать разрабатываемое приложение лишённым этих недостатков.

Для разработки приложения были разработаны дизайн-макеты интерфейса, выбраны средства для разработки приложения, построены все необходимые UML диаграммы и написана сопутствующая документация.

Для контроля версий приложения было создано 3 репозитория GitHub – для документации, клиентской и серверной части соответственно.

При разработке приложения были реализованы следующие задачи:

- предоставление преподавателями нежелательного времени для проведения занятий;
- обеспечение возможности просмотра расписания на заданный промежуток времени как для старост (для их групп), так и для преподавателей (личное);
- предоставление преподавателям права переноса занятий через приложение;
- формирование расписания системой на основе предоставленных преподавателями данных;
- экспорт расписания в формате Excel таблицы;
- уведомление старост по электронной почте об изменениях в расписании.

Разработанное приложение удовлетворяет всем ранее поставленным требованиям.

Все поставленные задачи были выполнены в полном объёме.

Список использованных источников

1. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.
2. The Docker Book: Containerization is the new virtualization Kindle Edition / James Turnbull. - James Turnbull, 2014. – 388 с.
3. Pro Git: Everything you need to know about Git / Scott Chacon, Ben Straub. – 2-е изд. – изд. Apress, 2014. – 458 с.
4. Что такое JWT-токен [Электронный ресурс]. - Режим доступа: <https://struchkov.dev/blog/ru/what-is-jwt/>. – Заглавие с экрана. - (Дата обращения: 25.05.2023).
5. Руководство по языку Kotlin [Электронный ресурс]. – Режим доступа: <https://metanit.com/kotlin/tutorial/>. - Руководство по языку Kotlin. – (Дата обращения: 31.05.2023).
6. Spring Framework Documentation [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-boot>. – Spring Boot. – (Дата обращения 31.05.2023).
7. Android Application Development Cookbook: 93 Recipes for Building Winning Apps / L. Wei-Meng. – New York: Wrox, 2013. – 408 с.