

Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О.В. Непомнящий  
" \_\_\_\_ " \_\_\_\_\_ 2023 г.

**БАКАЛАВРСКАЯ РАБОТА**

090301 Информатика и вычислительная техника

Мобильное приложение «The Bands»

Руководитель	_____	_____	доцент, канд. техн. наук	С.Н. Титовский
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		М.О. Прягин
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____		С.Н. Титовский
	<i>подпись</i>	<i>дата</i>		

Красноярск 2023

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	
1 Сравнительный анализ существующих аналогов.....	
1.2 Сравнительный анализ средств разработки мобильных приложений. ....	
1.2.1 PhoneGap.....	
1.2.2 Xamarin.....	
1.2.3 NativeScript.....	
1.2.4 Flutter.....	
1.3 Постановка задачи разработки.....	
1.4 Требования к мобильному приложению.....	
1.5 Выводы по главе.....	
2 Проектирование приложения.....	
2.1 Определение функциональных требований.....	
2.1.1 Авторизация.....	
2.1.2 Регистрация.....	
2.1.3 Навигация.....	
2.1.4 Просмотр групп.....	
2.1.5 Просмотр анкет.....	
2.1.6 Поиск анкет.....	
2.2 Структура приложения.....	
2.2.1 Определение структуры.....	
2.2.2 Структура представлений.....	
2.2.3 Структура моделей представления.....	
2.2.4 Структура моделей.....	
2.3 Выводы по главе.....	
3 Реализация и документация.....	
3.1 Выбор технологий.....	

3.2 База данных.....	
3.3 Реализация .....	
3.4 Выводы по главе.....	
ЗАКЛЮЧЕНИЕ .....	
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	

## ВВЕДЕНИЕ

Актуальность темы.

На сегодняшний день почти у каждого человека на Земле есть смартфон, в котором десятки мобильных приложений. Люди практически не выпускают телефон из рук.

Музыкантов становится всё больше, но далеко не каждый может найти себе музыкальную группу, в которую можно вступить.

Целью работы является разработка мобильного приложения “The bands” для поиска музыкантов и музыкальных групп, в которые музыканты смогут вступить. Разрабатываемое приложение предназначено для обеспечения упрощённого доступа пользователя к музыкальным группам, ищущего себе группу, а также предоставление упрощенного доступа для самих музыкальных групп к музыкантам.

Для достижения цели работы должны быть решены следующие задачи:

- провести анализ предметной области, изучив особенности уже существующих аналогов и проведя обзор современных средств разработки;
- спроектировать мобильное приложение;
- реализовать мобильное приложение.

## 1 Сравнительный анализ существующих аналогов.

В настоящий момент нет подобных приложений, решающих поставленную задачу. Но существуют подобные web-сайты:

– Splice — на этом web-сайте можно найти не только единомышленников в написании музыки, но и скачать бесплатные плагины, создать личное сообщество и открыть публичную сессию [2];

– ProCollabs — это web-сайт, который предлагает не только авторов музыки, но и других людей в сфере музыки. Если стать членом клуба, то можно, верифицировать платежную систему и тем самым продавать свои услуги и проекты [2].

– BandHub — это огромное сообщество музыкантов и продюсеров, которые ищут музыкантов для совместного создания музыки. Можно создавать свои собственные музыкальные группы по всей планете [2].

– Melboss — это музыкальный web-сайт, где музыканты и продюсеры объединяются. Здесь есть возможность выгружать свои композиции и находить соавторов [2].

Данные сайты объединяет то, что на всех нужно выгружать свои музыкальные композиции, чтоб найти себе соавтора. Однако это не подходит, если у вас нет музыкального произведения. Исходя из этого было принято решение сделать такое приложение, где не нужно выгружать своё музыкальное произведение, чтоб найти музыкальную группу или соавтора.

## **1.2 Сравнительный анализ средств разработки мобильных приложений.**

### **1.2.1 PhoneGap**

PhoneGap — это среда разработки кроссплатформенных мобильных приложений. С её помощью разработчики могут создавать приложения с помощью HTML, JavaScript и CSS [1].

Особенности PhoneGap:

- простота разработки: PhoneGap работает с HTML5, CSS3 и JavaScript;
- кроссплатформенность: Разработчики не тратят много времени для разработки приложения на все мобильные устройства.

К достоинствам среды разработки относят:

- код проекта разработчики могут использовать на разных мобильных устройствах;
- быстрое тестирование кода.

Недостатки системы — интерфейс может быть менее интерактивным для некоторых пользователей.

### **1.2.2 Xamarin**

Разработчики смогут пользоваться Xamarin для разработки кроссплатформенных мобильных приложений, пользуясь знаниями JavaScript, CSS и HTML [1].

Особенности Xamarin

- полное связывание: существуют привязки почти для всех SDK базовой платформы в iOS и Android;
- Objective-C, Java, C и C++ Interop: позволяет использовать различный код других разработчиков;

– современные языковые конструкции: Приложения Xamarin разрабатываются на C#, превосходящий Objective-C и Java;

– кроссплатформенная поддержка: возможность разрабатывать одно приложение для всех мобильных ОС.

К достоинствам среды разработки относят:

– разработчики имеют возможность создавать нативные мобильные приложения с помощью бесплатной среды разработки.

Система Xamarin не рекомендуется для разработчиков, которые желают создавать мобильные игры.

### **1.2.3 NativeScript**

NativeScript — это открытый фреймворк, предназначенный для создания нативных мобильных приложений для Android и iOS. Для разработки в этом фреймворке используются языки программирования, такие как JavaScript и CSS [1].

Особенности NativeScript:

– расширяемость: обеспечивает полный доступ ко всем API-интерфейсам iOS и Android. Это позволяет иметь лучшую доступность и использовать бесплатные плагины, Android SDK и CocoaPods;

– многофункциональность: Разработчики имеют возможность заниматься различными задачами, начиная от интеграции определенных платформ до адаптации и распространения приложений для конкретных платформ или устройств. Благодаря улучшениям, установка плагинов и отладка приложений стали более удобными и быстрыми.

К достоинствам среды разработки относят:

– ПО работает быстрее конкурентов;

– ПО позволяет разрабатывать кроссплатформенные приложения.

Недостатки системы — частые обновления, что затрудняют разработчикам привыкнуть к одной версии.

#### **1.2.4 Flutter**

Фреймворк Flutter от Google — это набор средств разработки ПО для мобильных приложений. Среда позволяет разрабатывать мобильные приложения, работающие на Android и iOS с использованием единой базы данных [1].

Особенности Flutter:

- перезагрузка в режиме реального времени: это позволяет сразу видеть изменения в мобильном приложении при разработке;

- кроссплатформенная разработка: разработанное приложение можно использовать на разных мобильных операционных системах.

К достоинствам среды разработки относят:

- разработчик имеет возможность сохранить ценную документацию в виде закладки.

Недостатки системы — Flutter не поддерживает API, использующие GPS и Bluetooth.

Исходя из вышеперечисленного, был сделан вывод о том, что более целесообразно будет выбрать фреймворк React Native, в связи с опытом разработки на нём.

### **1.3 Постановка задачи разработки**

Разработка мобильных приложений сейчас очень востребована ввиду популярности мобильных устройств.



Музыкантов становится всё больше ввиду популярности музыки и простоты доступа к музыкальным программам. Музыканты всё чаще остаются одни и не могут найти подходящую группу и делают музыку в одиночку.

The Bands – мобильное приложение, с помощью которого музыканты смогут легко найти себе подходящую группу. Приложение будет работать с помощью анкет. Люди будут регистрироваться и оставлять анкеты, где описано, на каких музыкальных инструментах они играют и как оценивают свои способности. Также сами группы смогут оставлять объявления, что им нужен музыкант на каком-либо инструменте.

#### **1.4 Требования к мобильному приложению**

Пользователь будет иметь возможность регистрироваться и авторизоваться. Данные для входа в аккаунт будут храниться в базе данных и проверяться при входе. Также пользователю будет предложено создать свою анкету, если он хочет найти себе музыкальную группу или музыканта, если группа у него уже существует. Будет возможность просматривать анкеты других пользователей. Пользователь сможет отредактировать свою анкету, изменив данные, все изменения в анкете будут изменяться и в базе данных. Если пользователю не нужно уже искать музыканта или группу, он сможет легко удалить анкету. Анкета также удалится из базы данных.

#### **1.5 Выводы по главе**

Подводя итоги, мы можем выделить, какие возможности будут у мобильного приложения:

- регистрация и авторизация;
- создание и просмотр анкет;
- редактирование анкет;
- удаление анкеты;



## **2 Проектирование приложения**

### **2.1 Определение функциональных требований**

При проектировании предметной области были сформулированы требования к функционалу. Функционал разбит на модули и выглядит следующим образом:

1. Модуль аутентификации
  - Авторизация и регистрация пользователя
2. Модуль навигации
  - Перемещение между модулями
3. Модуль анкет групп
  - Отображения списка групп
  - Отображения страницы группы
  - Редактирование страницы группы
  - Просмотр заявок на присоединение к группе
  - Назначение пользователей на свободные роли
4. Модуль анкет пользователей
  - Просмотр анкеты пользователя
  - Редактирование анкеты
  - Создание анкеты
5. Модуль профиля
  - Просмотр профиля

С учетом всех ранее перечисленных возможностей, можно создать диаграмму прецедентов, которая покажет действия, доступные пользователю для выполнения. Диаграмма прецедентов изображена на рисунке 2.8.

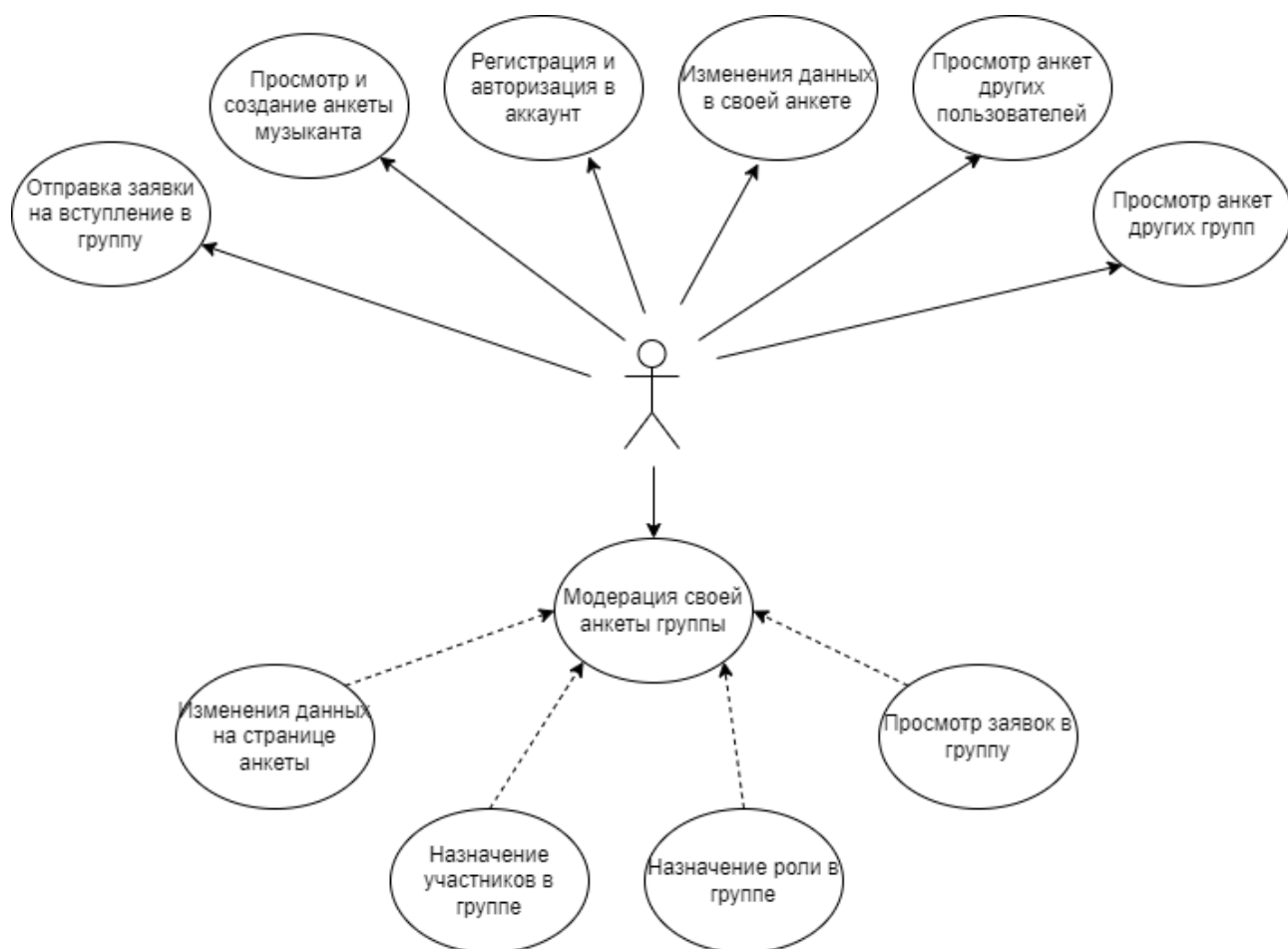


Рисунок 2.1 – Диаграмма прецедентов

### 2.1.1 Авторизация

В данном модуле осуществляется процесс авторизации в системе. Пользователь вводит свои учетные данные, такие как "логин" и "пароль", после чего приложение отправляет запрос на сервер для проверки правильности аутентификации. Сервер может либо подтвердить успешную аутентификацию, либо вернуть ошибку в ответ. При положительном ответе от сервера, приложение переключается на экран профиля пользователя. Процесса аутентификации в виде диаграммы последовательностей представлена на рисунке 2.1.

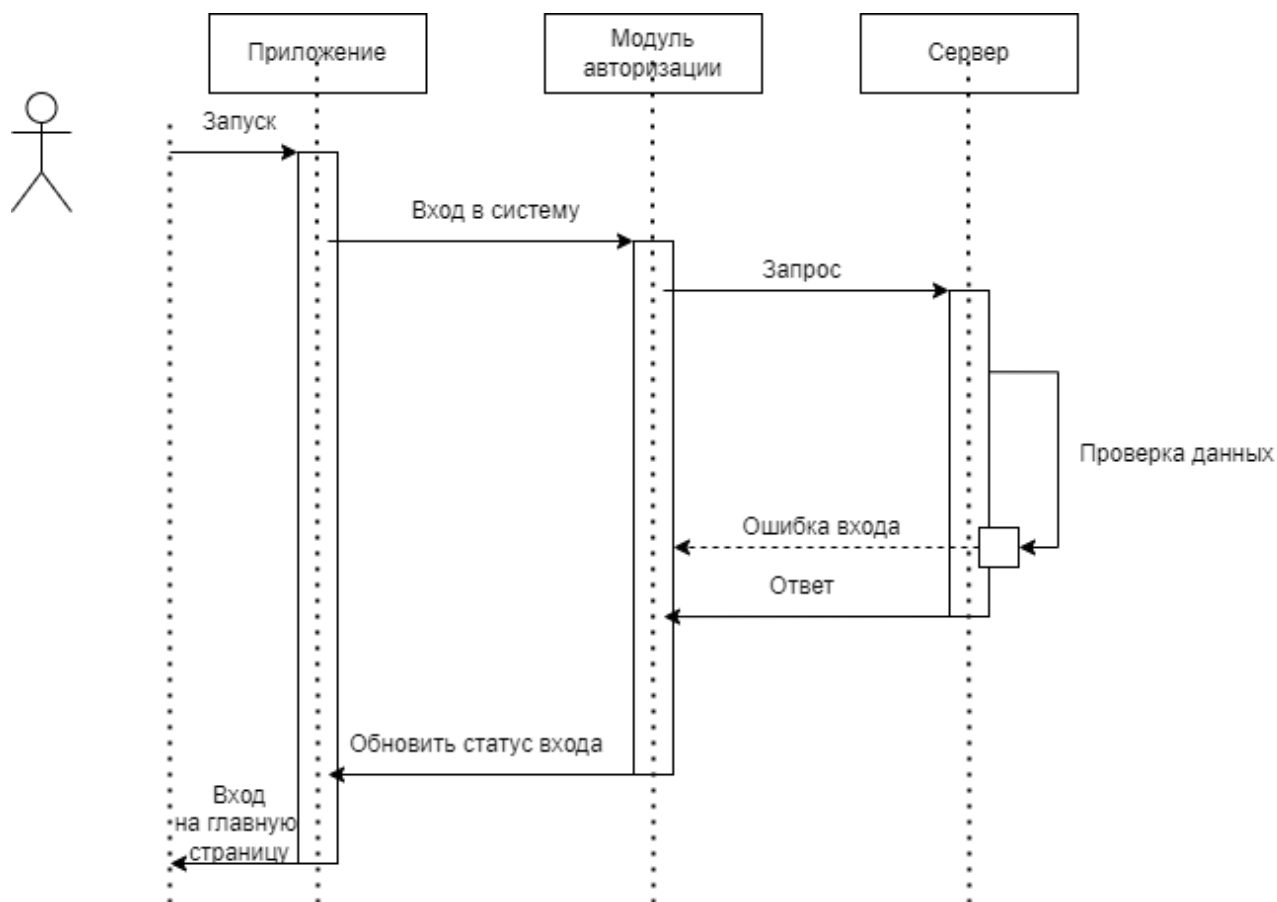


Рисунок 2.2 – Диаграмма последовательности процесса авторизации

### 2.1.2 Регистрация

В этом модуле осуществляется процесс регистрации пользователя. После ввода пользователем своих данных, таких как "логин", "пароль" и "имя", приложение отправляет запрос на сервер для выполнения регистрации. Сервер отвечает либо согласием, либо ошибкой. В случае получения положительного ответа, приложение переключается на экран авторизации. Диаграмма последовательности процесса регистрации представлена на рисунке 2.2.

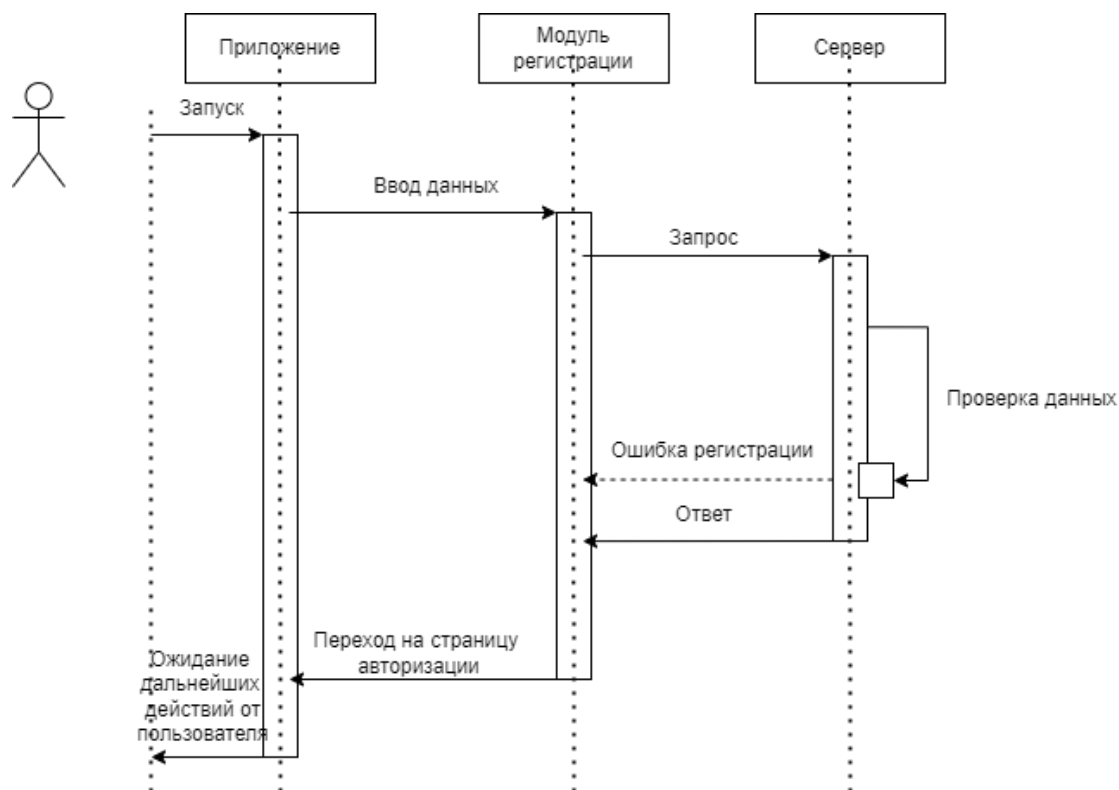


Рисунок 2.3 – Диаграмма последовательности процесса регистрации

### 2.1.3 Навигация

Приложение разделено на различные модули, каждый из которых отвечает за определенный функционал. Модули взаимосвязаны через меню навигации, что обеспечивает быстрый переход пользователей между различными функциональными возможностями приложения. Однако для получения доступа к основному навигационному меню пользователю необходимо успешно пройти процесс авторизации. В меню находятся три кнопки: "Анкеты групп", "Анкеты музыкантов" и "Профиль". Навигационная схема между модулями приложения изображена на рисунке 2.3.

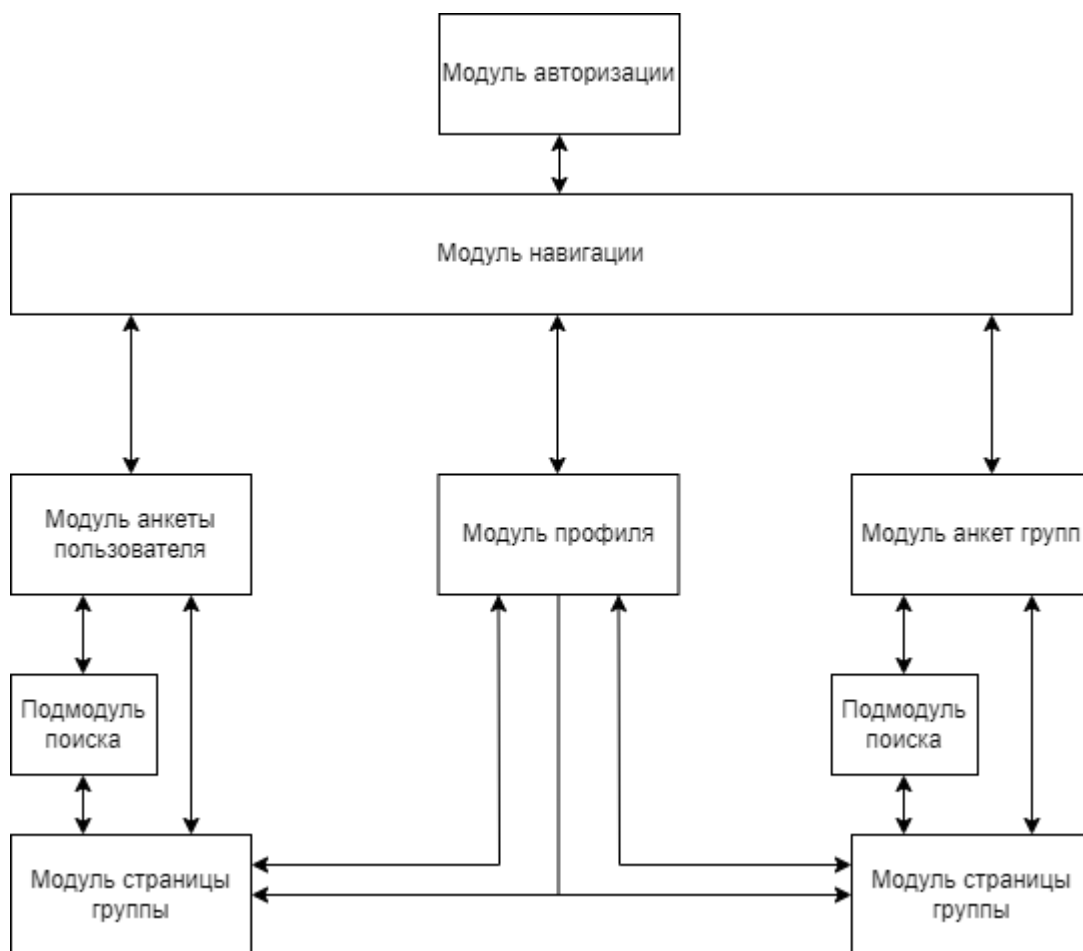


Рисунок 2.4 – Навигационная схема между модулями

#### 2.1.4 Просмотр групп

– Модуль анкет групп предоставляет пользователю информацию о группе и доступен через меню навигации. Данный модуль состоит из двух подмодулей.

Подмодуль группы: показывает пользователю, в какой группе он состоит;

– Подмодуль страницы группы: выполняет вывод страницы запрашиваемой группы;

– Подмодуль поиска: выполняет вывод анкет групп, который выводит список анкет групп.

### 2.1.4.1 Группы

Модуль списка групп выполняет отображение группы, в которой пользователь состоит или отправил заявку. Здесь отображается: название группы, роль в группе. Также в модуле списка групп есть функции для поиска или создание своей группы.

Взаимодействие с модулем списка групп показано на рисунке 2.4.

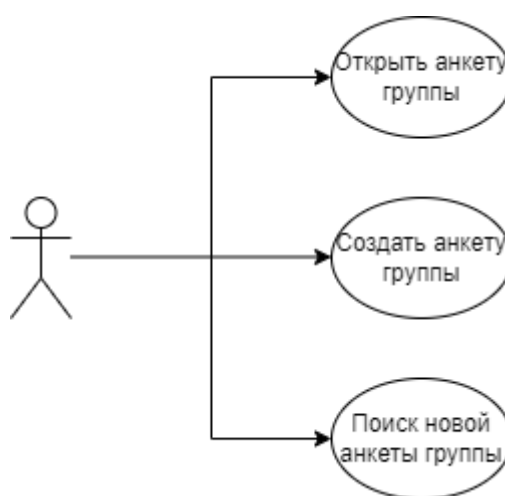


Рисунок 2.5 – Использование модуля списка анкет групп

### 2.1.4.2 Страница группы

Функциональность модуля "Страница группы" включает в себя возможности создания, редактирования и просмотра страницы группы. На этой странице основатели группы имеют возможность указать название и описание своей группы.

Основатель группы может перейти в меню редактирования группы, назначение ролей и просмотр заявок.

Требуется также реализовать функционал просмотра страницы группы для пользователей, которые еще не являются участниками грёуппы, чтобы они могли отправить заявку на вступление в группу.



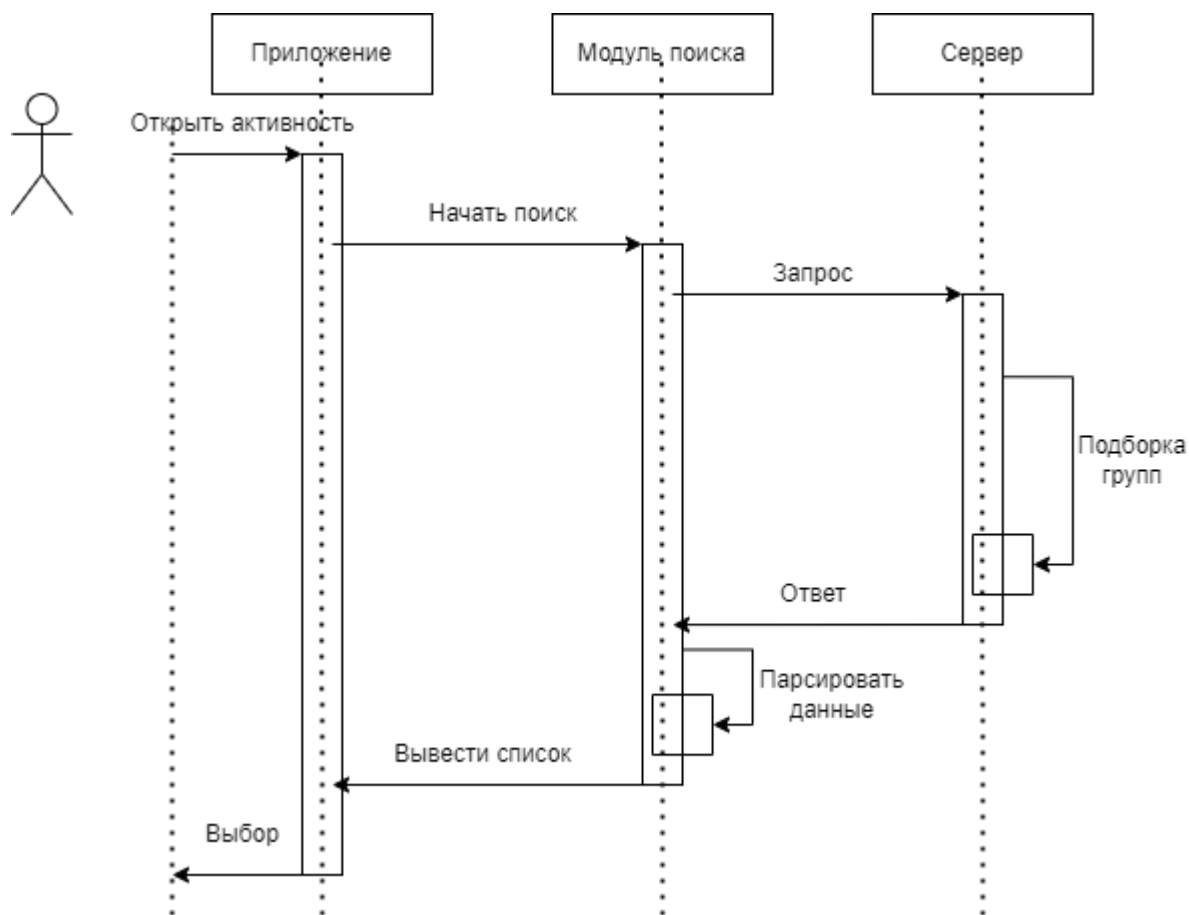
Методы взаимодействия с модулем страницы группы изображены на рисунке 2.5.



Рисунок 2.6 – Использование модуля страницы группы.

### 2.1.4.3 Поиск групп

Модуль "Поиск новых групп" выполняет запрос на сервер с целью получения полного списка групп. После отправки запроса на сервер, модуль поиска получает необходимые данные, преобразует их в удобный для пользователя формат и передает результаты модулю списка групп. Диаграмма последовательности работы модуля изображена на рисунке 2.6.



2.7 – Диаграмма последовательности поиска новых групп

### 2.1.5 Просмотр анкет

Модуль "Анкета пользователя" обеспечивает отображение анкеты пользователя. Пользователь может просматривать анкеты других пользователей и редактировать свою собственную анкету.

Диаграмма возможностей использования модуля анкет пользователей изображена на рисунке 2.7.



Рисунок 2.8 – Использование модуля анкеты пользователя

### 2.1.6 Поиск анкет

Модуль "Поиск анкет музыкантов" осуществляет отправку запроса на сервер с целью получения списка всех анкет музыкантов. После отправки запроса на сервер, модуль поиска получает необходимые данные, преобразует их в удобный для пользователя формат и передает результаты модулю списка анкет музыкантов.

Диаграмма последовательности работы модуля изображена на рисунке 2.9.

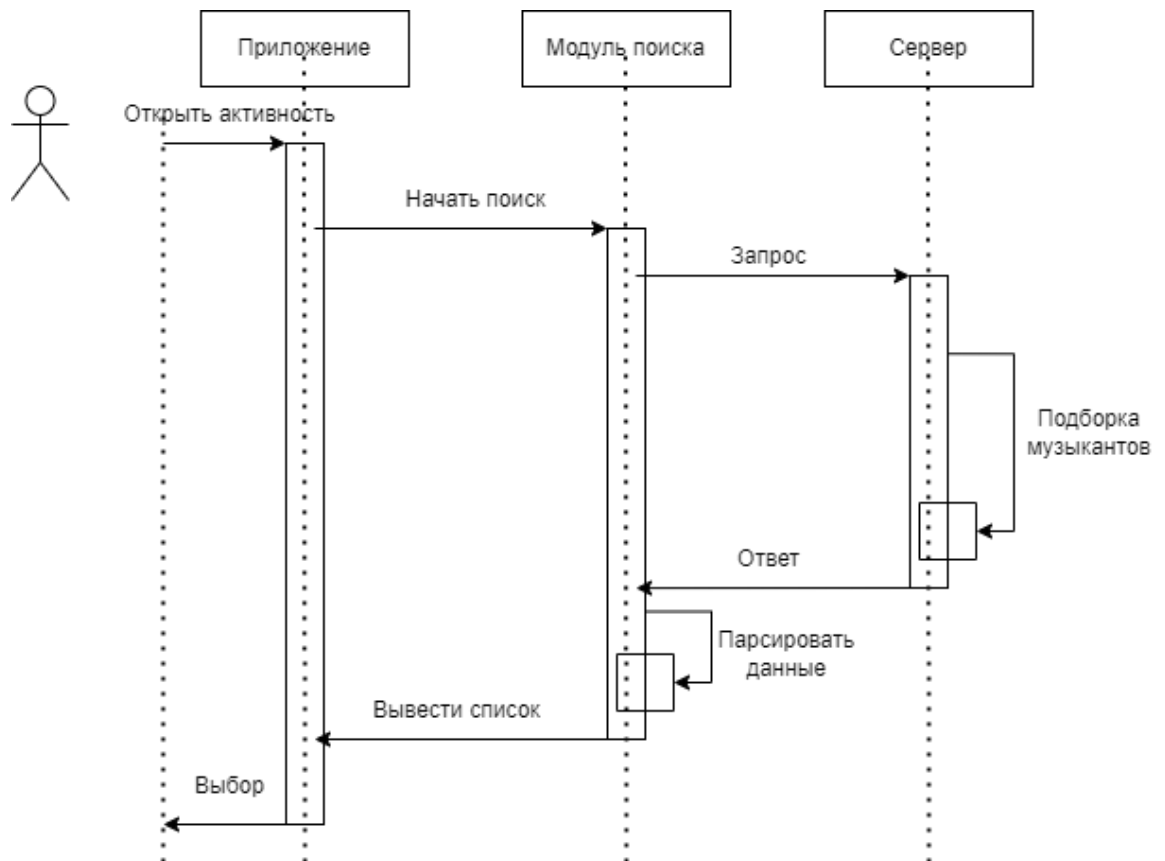


Рисунок 2.9 – Диаграмма последовательностей поиска анкет музыкантов

## 2.2 Структура приложения

### 2.2.1 Определение структуры

Сперва стоит определить будущую структуру приложения, перед тем, как начать его проектировать. Рассмотрим наиболее распространённые структуры:

– MVC (Model-View-Controller) – это классическая структура, которая широко используется в разработке мобильных приложений. Данная структура позволяет разделить приложения на три ключевых компонента: модель, представление и контроллер. Модель отвечает за хранение данных и реализацию бизнес-логики, представление отвечает за отображение данных на

экране, а контроллер управляет потоком данных между моделью и представлением. [3].

– MVP (Model-View-Presenter) - это вариант модификации MVC, в котором контроллер заменяется презентером. Презентер отвечает за управление потоком данных между моделью и представлением, а также обрабатывает пользовательские события [3].

– MVVM (Model-View-ViewModel) – это ещё более современная структура, которая предлагает более чёткое разделение ответственности. ViewModel отвечает за предоставление данных, необходимых для отображения на экране, а также за управление состоянием представления. Она связывает модель и представление, но не имеет прямого доступа к представлению [3].

Для данного приложения была выбрана структура MVVM. В MVVM структуре, ViewModel будет отвечать за предоставление данных, необходимых для отображения на экране, например, списка групп и музыкантов. ViewModel будет связываться с моделью, чтобы получать данные, и с представлением, чтобы обновлять его состояние. Представление будет отображать данные, которые предоставляет ViewModel. Принцип работы паттерна MVVM изображён на рисунке 2.9.

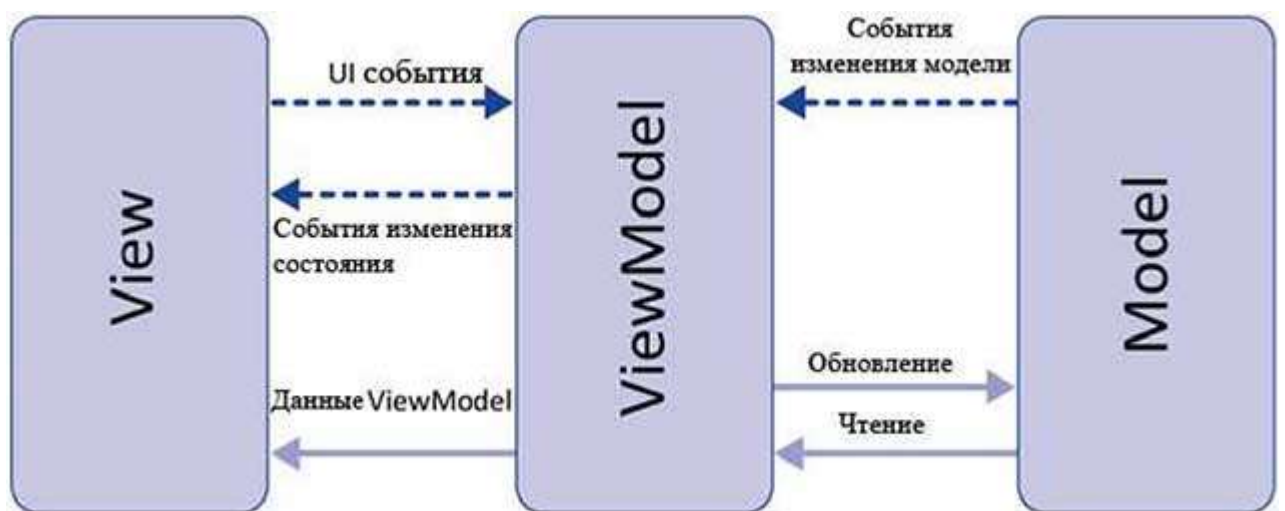


Рисунок 2.9 – Паттерн MVVM

## 2.2.2 Структура представлений

Представления являются неотъемлемой частью приложения, с которыми пользователь взаимодействует. В соответствии с требованиями к функционалу, нам потребуются следующие представления:

- RegisterScreen – это стартовая активность, которая отвечает за процесс регистрации пользователя;

- LoginScreen – это активность, предназначенная для авторизации пользователя;

- MainScreen – это главная активность приложения, которая содержит фрагменты для навигации между различными функциональными модулями и подмодулями;

- ProfileScreen – активность, отображающая профиль пользователя;

- PeopleScreen – активность, отображающая все анкеты;

- CreateFormScreen – активность, позволяющая создать свою анкету музыканта;

- FormDetailsScreen – активность, отображающая полную информацию об анкете музыканта;

- EditFormScreen – активность, позволяющая редактировать свою анкету музыканта;

- MyFormScreen – активность, отображающая личную анкету музыканта;

- BandScreen – активность, отображающая все анкеты музыкальных групп;

- CreateBandScreen – активность, позволяющая создать анкету группы;

- BandDetailsScreen – активность, отображающая полную информацию об анкете группы;

- EditBandScreen – активность, позволяющая редактировать свою анкету группы;

- BandMembersScreen – активность, отображающая участников группы;

- MyBandScreen – активность, отображающая личную анкету группы;
- MyBandMembersScreen – активность, отображающая участников личной группы;
- MyBandApplicationsScreen – активность, отображающая заявок на участие в группе.

### 2.2.3 Структура моделей представления

Конечная структура представления может зависеть от конкретных требований проекта и архитектуры приложения, но для данной структуры представлений можно сделать следующую структуру моделей представления:

- RegisterViewModel – отвечает за регистрацию пользователя;
- LoginViewModel – отвечает за авторизацию пользователя;
- ProfileViewModel – передаёт данные о пользователе;
- PeopleViewModel – передаёт данные о анкетах музыкантов;
- CreateFormViewModel – выполняет создание анкеты музыканта;
- MyFormViewModel – передаёт данные о личной анкете музыканта, выполняет редактирование и удаление анкеты;
- BandsViewModel – передаёт данные анкет всех групп и личной анкеты, выполняет создание, редактирование и удаление анкеты;
- ApplicationsViewModel – передаёт данных всех заявок в группу, выполняет создание, принятие или отклонения заявки в группу;
- BandMembersViewModel – передаёт всех участников группы и выполняет удаление участника.

### 2.2.4 Структура моделей

Модель — это составная часть приложения, которая описывает данные и логику, а также обеспечивает взаимодействие между данными и

пользовательским интерфейсом. Модель включает в себя базу данных и репозиторий. В смартфонах под управлением ОС Android предустановлена база данных SQLite, которая используется для этой цели. В базе данных нам потребуется создать 5 таблиц:

Таблица User (Таблица 1).

Таблица 1 – Таблица User

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничения</b>
id	Ключ	Integer	Primary key, not null
login	Логин пользователя	Text	Unique, not null
password	Пароль пользователя	Text	Not null
userName	Имя пользователя	Text	Не более 200 символов

Таблица form (Таблица 2)

Таблица 2 – таблица Form

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничение</b>
id	Ключ	Integer	Primary key, not null
userId	Идентификатор пользователя, создавшего анкету	Integer	Foreign key(user), not null
userName	Имя	Text	Не более 200 символов
Age	Возраст	Integer	От 14 до 100



Окончание таблицы 2

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничение</b>
City	Город	Text	Не более 200 символов
Gender	Пол	Text	Не более 20 символов
MusicalInstrument	Музыкальный инструмент	Text	Не более 200 символов
Description	Описание	Text	Не более 1000 символов
Communication	Телефон или ссылка для связи	Text	Не более 500 символов

Таблица Band (Таблица 3)

Таблица 3 – Таблица Band

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничение</b>
id	Ключ	Integer	Primary key, not null
userId	Идентификатор пользователя, создавшего анкету	Integer	Foreign key(user), not null
bandName	Название группы	Text	Не более 200 символов
Description	Описание	Text	Не более 1000

			СИМВОЛОВ
--	--	--	----------

Окончание таблицы 3

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничение</b>
Contact	Номер телефона или ссылка на создателя групп	Text	Не более 500 СИМВОЛОВ

Таблица member (таблица 4)

Таблица 4 – Таблица Member

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничение</b>
id	Ключ	Integer	Primary key, not null
userId	Идентификатор пользователя, состоящего в группе	Integer	Foreign key(user), not null
BandId	Идентификатор группы, в которой состоит пользователь	Integer	Foreign key(band), not null
Role	Роль пользователя	Text	Не более 200 СИМВОЛОВ

Таблица Application (таблица 5)

Таблица 5 – таблица Application

<b>Имя</b>	<b>Назначение</b>	<b>Тип данных</b>	<b>Ограничение</b>
id	Ключ	integer	Primary key, not null
userId	Идентификатор пользователя, отправившего заявку	integer	Foreign key(user), not null
BandId	Идентификатор группы, в которую отправлена заявка	integer	Foreign key(band), not null
Status	Статус заявки	text	Не более 200 СИМВОЛОВ
Role	Роль пользователя	text	Не более 200 СИМВОЛОВ

Схема взаимосвязей таблиц представлена на рисунке 2.10.

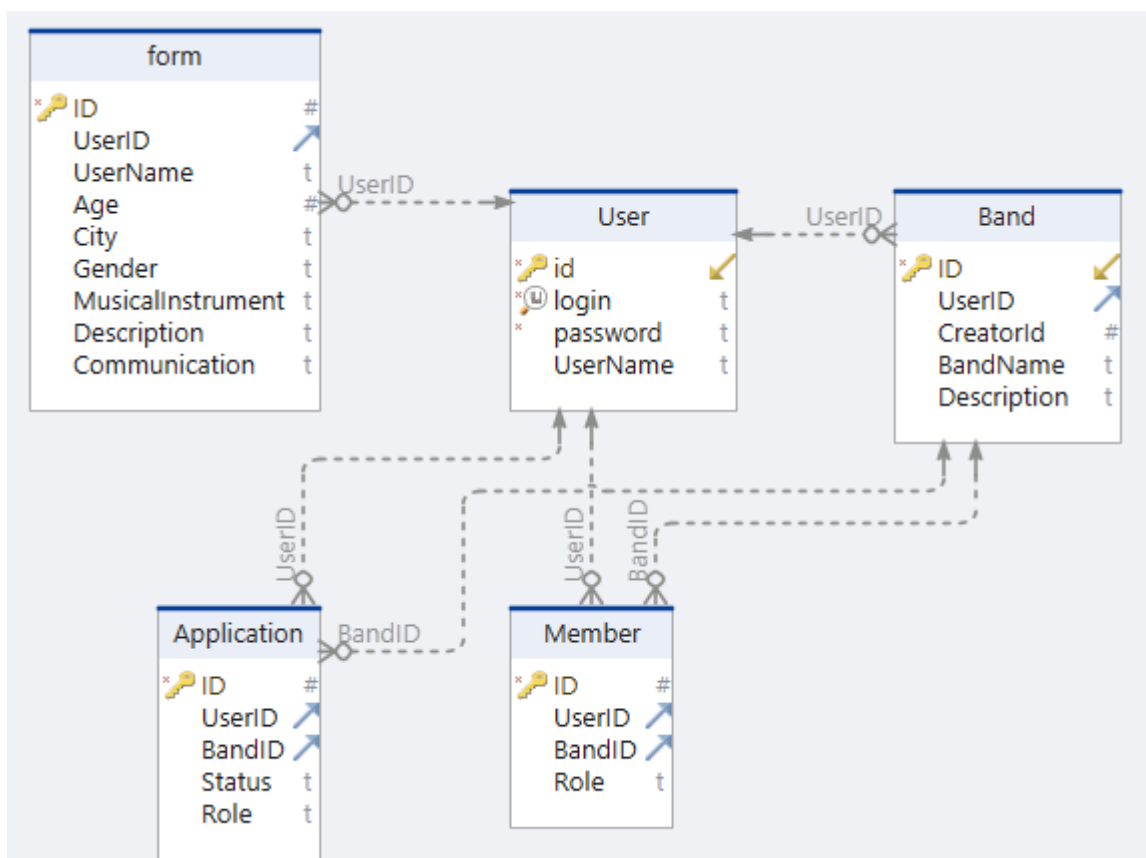


Рисунок 2.10 – Схема взаимосвязей таблиц

### 2.3 Выводы по главе

- В процессе проектирования приложения были выполнены следующие задачи:
- определен выбранный шаблон для архитектуры приложения;
- разработана архитектура приложения;
- обозначены необходимые функциональные возможности для каждого модуля приложения;
- создана диаграмма прецедентов, которая отражает все взаимодействия пользователя с приложением.

## **3 Реализация и документация**

### **3.1 Выбор технологий**

Для реализации клиентской части был выбран фреймворк React Native[4].

React Native - это фреймворк для разработки мобильных приложений, который позволяет использовать JavaScript и React для создания высокопроизводительных приложений, работающих на различных платформах, включая iOS, Android и веб-платформу. Он базируется на React, одном из самых популярных фреймворков для создания пользовательских интерфейсов веб-приложений. React Native применяет ту же архитектуру и принципы, что и React, но позволяет разработчикам создавать мобильные приложения с использованием JavaScript и нативных компонентов. Это обеспечивает более высокую производительность и создает пользовательский интерфейс, более близкий к нативным приложениям.

Для серверной части был выбран фреймворк Nest.js [5].

Nest.js — это фреймворк для разработки серверных приложений на Node.js с использованием TypeScript. Он основан на концепции модулей и зависимостей, и предоставляет множество инструментов для создания масштабируемых и модульных приложений. Nest.js значительно упрощает процесс разработки серверных приложений на Node.js, позволяя создавать более удобный и безопасный код, организовывать приложение в модули, а также предоставляет удобное API для взаимодействия с сервером.

### **3.2 База данных**

В качестве базы данных была выбрана SQLite[6]. SQLite — это легковесная реляционная база данных, которая использует локальный файл в качестве хранилища данных. Она не требует установки каких-либо

дополнительных компонентов или серверов баз данных, и ее можно использовать на большинстве популярных операционных систем.

SQLite хранит данные в виде таблиц, и поддерживает стандартный язык запросов SQL для выполнения операций вставки, обновления, удаления и выборки данных. Она имеет поддержку для транзакций и индексов, что обеспечивает хорошую производительность при работе с большими объемами данных.

В качестве ORM был выбран TypeORM[7]. TypeORM — это ORM (Object-Relational Mapping) для TypeScript и JavaScript, который облегчает работу с реляционными базами данных. Он поддерживает несколько баз данных, включая MySQL, PostgreSQL, SQLite, MS SQL Server и другие.

TypeORM позволяет определять модели данных, используя TypeScript-классы, что обеспечивает статическую типизацию при работе с базой данных. Это делает процесс разработки более безопасным и предсказуемым, уменьшая количество ошибок в коде.

TypeORM использует язык SQL для выполнения запросов к базе данных, но предоставляет более высокоуровневый интерфейс, который позволяет использовать объекты и методы для работы с данными. Он также обладает продвинутыми функциями, такими как генерация миграций и валидация данных.

### **3.3 Реализация**

В ходе разработки мобильного приложения были реализованы следующие функции:

– Авторизация и регистрация: была создана система авторизации и регистрации пользователей. Пользователи могут создать учетные записи, вводя свои данные, и использовать их для входа в приложение. Экраны регистрации и авторизации представлены на рисунке 3.1.

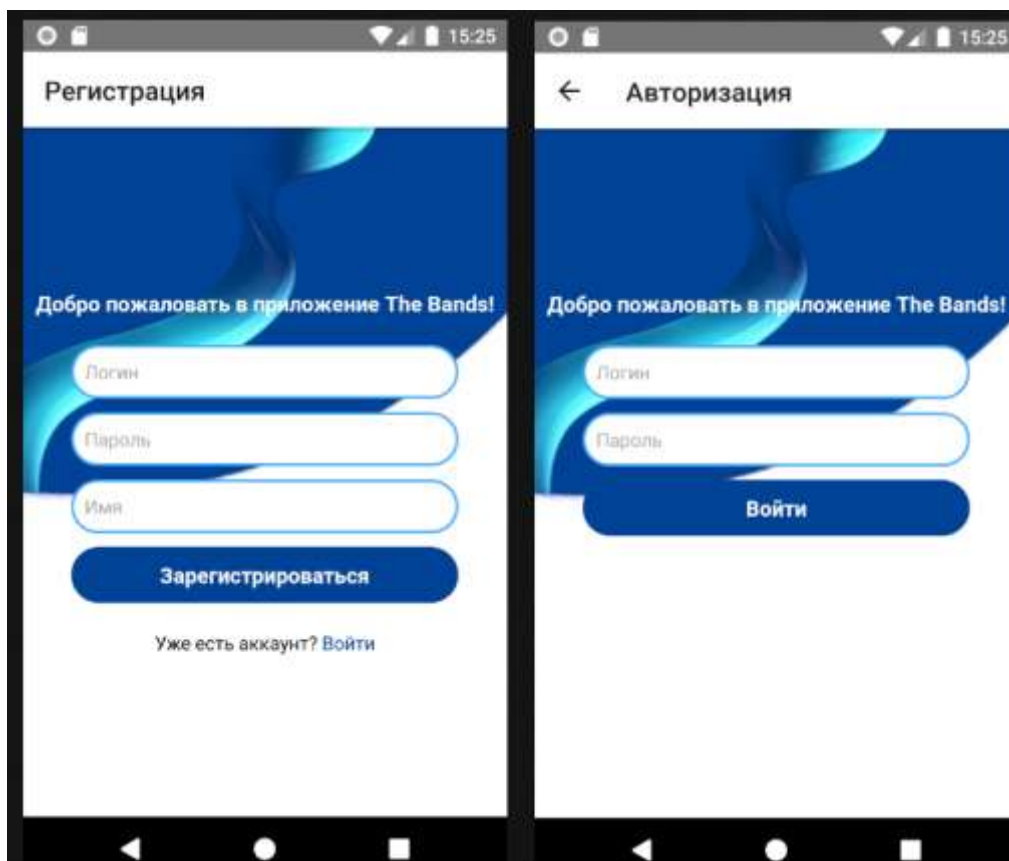


Рисунок 3.1 – Экраны регистрации и авторизации

– Просмотр и создание анкеты музыканта: в приложении предусмотрена возможность просмотра анкет музыкантов. Пользователи могут просматривать информацию о других музыкантах, такую как их инструменты, город, описание музыканта и т.д. Кроме того, реализована функция создания собственной анкеты музыканта, где пользователь может указать свои данные. Экраны просмотра и создание анкет представлены на рисунке 3.2.

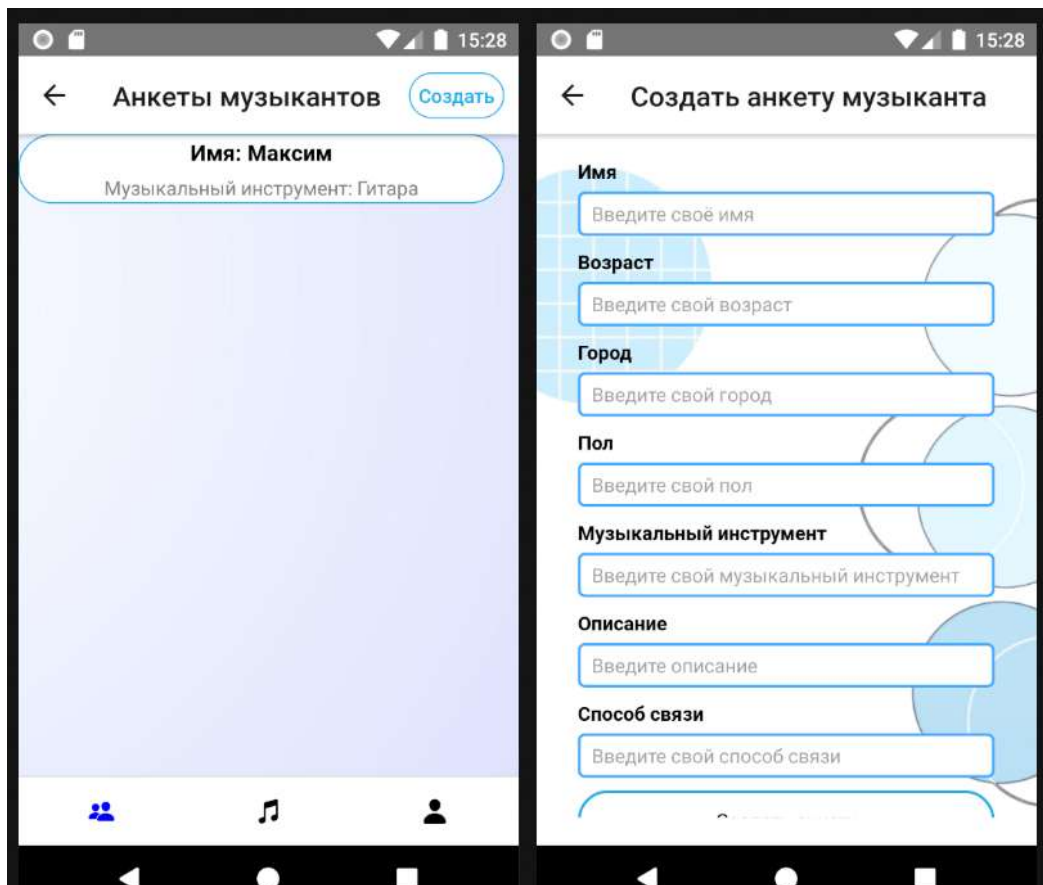


Рисунок 3.2 – Экраны просмотра и создания анкет

– Изменение данных в анкете музыканта: пользователи имеют возможность изменять данные в своей анкете музыканта. Был создан функционал, который позволяет пользователям обновлять информацию о своих данных. Экран изменение данных в анкете музыканта представлен на рисунке 3.3.



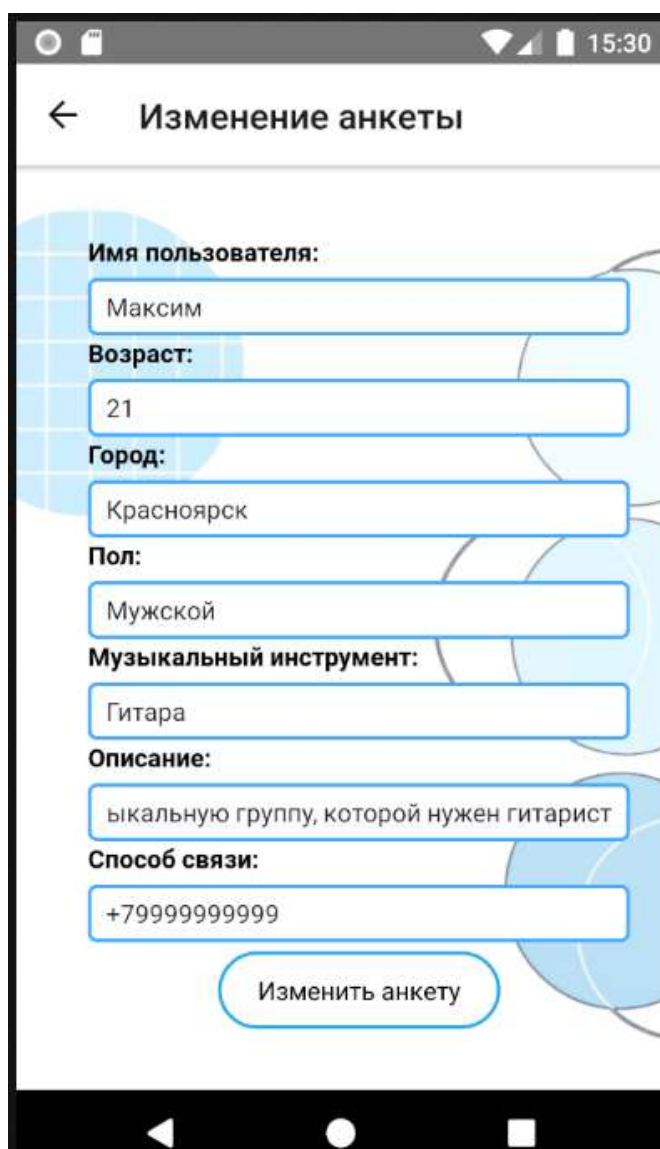


Рисунок 3.3 – Изменение данных в анкете музыканта

– Просмотр анкет других музыкальных групп: в мобильном приложении реализован функционал просмотра анкет музыкальных групп. Это позволяет пользователям искать и находить интересующие их группы. Экран с анкетами музыкальных групп представлен на рисунке 3.4.

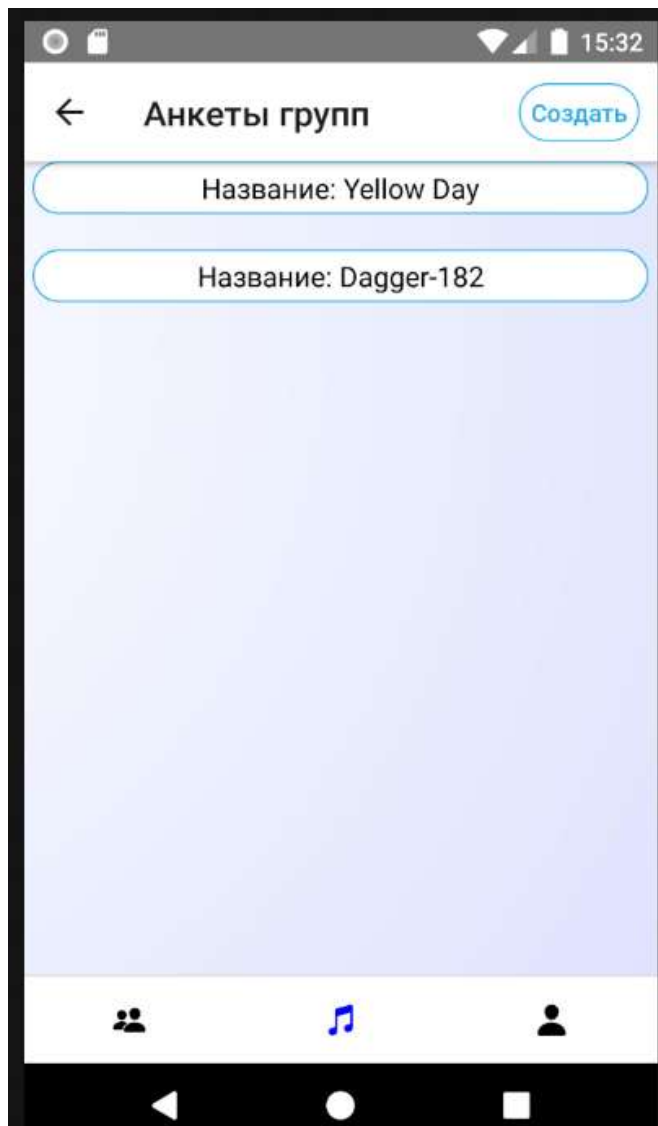


Рисунок 3.4 – Анкеты музыкальных групп

– Создание своей анкеты музыкальной группы: пользователи имеют возможность создать анкету для своей группы. Был разработан функционал, который позволяет пользователям указывать информацию о своей группе, такую как название и описание. Создание своей анкеты музыкальной группы представлено на рисунке 3.5.

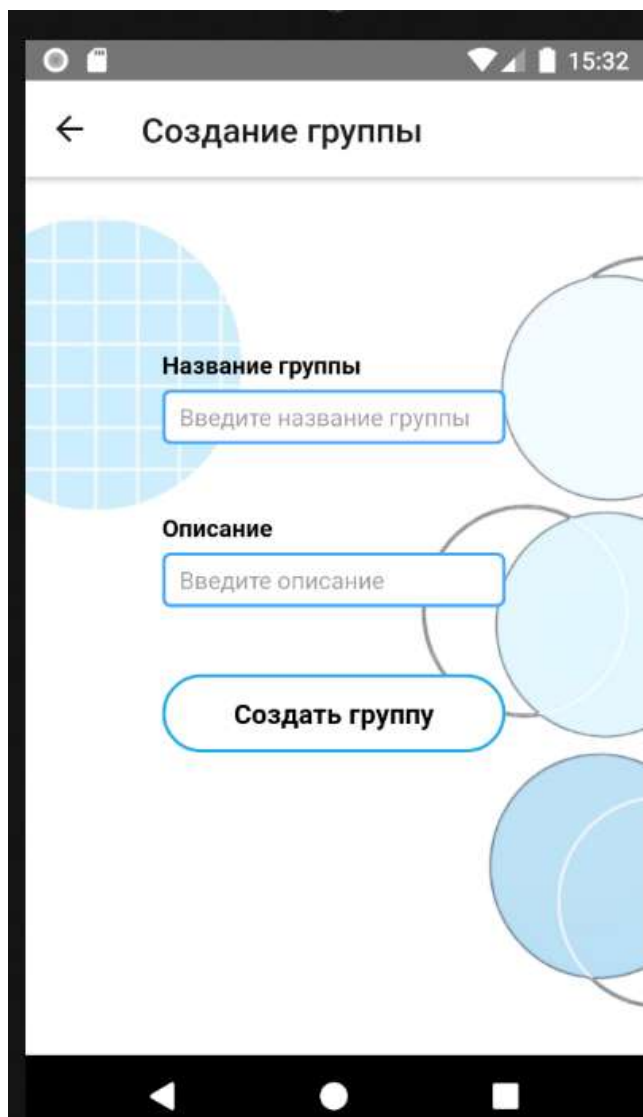


Рисунок 3.5 – Создание анкеты музыкальной группы

– Изменение данных на странице своей музыкальной группы: создатели музыкальных групп имеют возможность изменять данные на странице своей группы. Был создан функционал, позволяющий обновлять информацию о группе. Изменение данных анкеты группы представлен на рисунке 3.6.

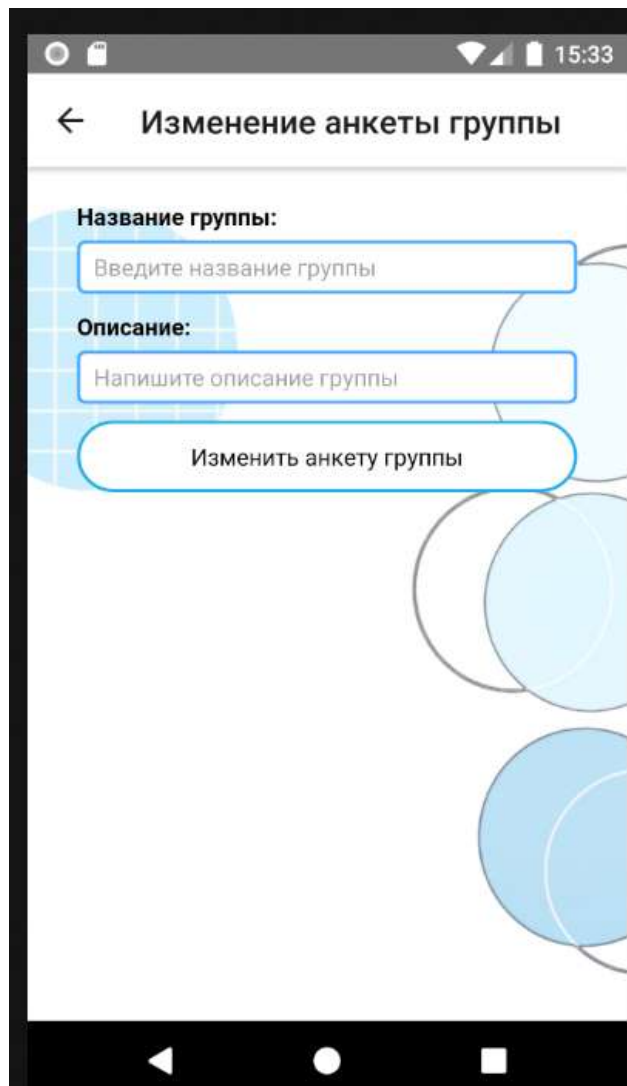


Рисунок 3.6 – Изменение данных анкеты группы

– Отправка заявки в музыкальную группу: пользователи имеют возможность отправлять заявки на вступление в музыкальные группы. Был разработан соответствующий функционал, позволяющий пользователям выбирать группу, в которую они хотят вступить, и отправлять соответствующую заявку. Отправка заявки в группу представлена на рисунке 3.7.



Рисунок 3.7 – Отправка заявки в группу

– Просмотр заявок других пользователей на вступление в музыкальную группу: создатели музыкальных групп имеют возможность просматривать заявки от других пользователей, желающих вступить в их группу. Это позволяет администраторам принимать решение о принятии или отклонении заявок. Просмотр заявок в музыкальную группу представлен на рисунке 3.8.

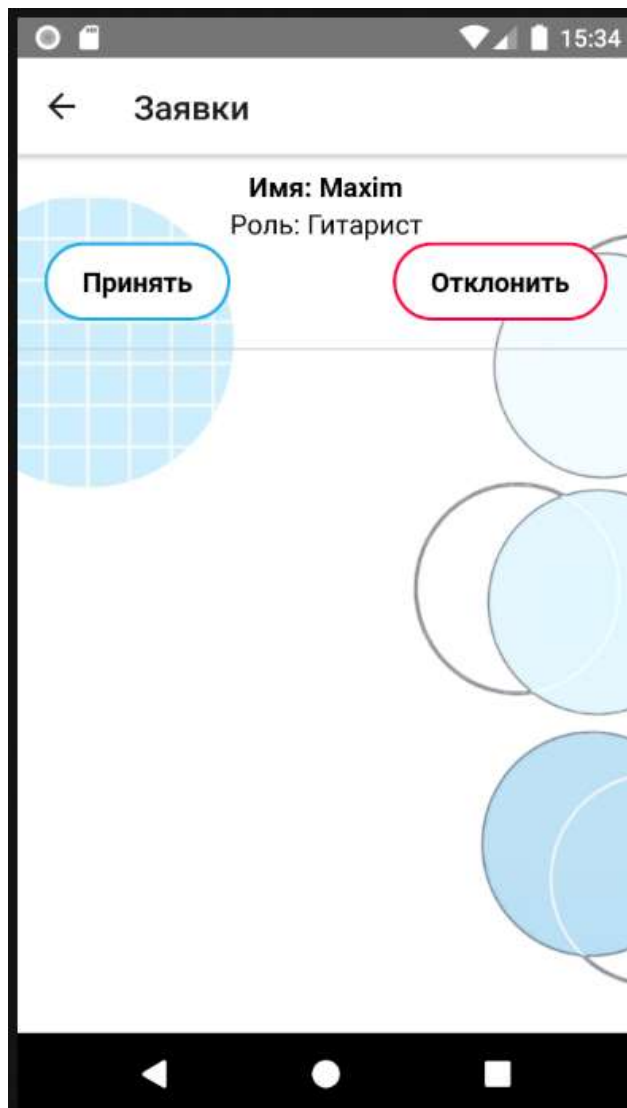


Рисунок 3.8 – Просмотр заявок в музыкальную группу

### 3.4 Выводы по главе

В результате работы над реализацией приложения были выполнены следующие задачи:

- Были выбраны технологии, с помощью которых было разработано мобильное приложение;
- Были описаны основные функции и элементы приложения.

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы были выполнены следующие этапы:

- сравнительный анализ существующих аналогов;
- сравнительный анализ современных средств разработки мобильных приложений;
- постановка задачи разработки;
- определение требований к мобильному приложению;
- определение функциональных требований к мобильному приложению;
- определение структуры приложения;
- выбор технологий для разработки приложения;
- реализация приложения;

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Clickfraud: русскоязычный интернет ресурс о разработке мобильных приложений. — URL: <https://clickfraud.ru/luchshee-programmnoe-obespechenie-dlya-razrabotki-mobilnyh-prilozhenij-v-2022-godu/> (дата обращения: 23.11.2022).
2. Arefyevstudio: русскоязычный интернет ресурс о музыке. — URL: <https://arefyevstudio.com/2017/04/17/7-luchshix-sajtov-dlya-poiska-muzykantov/> (дата обращения: 28.11.2022).
3. Habr: русскоязычный веб-сайт в формате системы тематических коллективных блогов с элементами новостного сайта. — URL: <https://habr.com/ru/post/344184/> (Дата обращения: 15.01.2023).
4. React Native: официальный веб-сайт фреймворка React Native. — URL: <https://reactnative.dev> (Дата обращения: 26.02.2023)
5. NestJS: официальный веб-сайт фреймворка NestJS. — URL: <https://nestjs.com> (Дата обращения: 26.02.2023).
6. SQLite: официальный веб-сайт базы данных SQLite. — URL: <https://www.sqlite.org/index.html> (Дата обращения: 26.02.2023).
7. TypeORM: официальный веб-сайт TypeORM. — URL: <https://typeorm.io> (Дата обращения 26.02.2023).
8. Draw.io: Бесплатное кроссплатформенное программное обеспечение для рисования графиков. — URL: <https://www.drawio.com> (Дата обращения 14.03.2023)
9. СТО 4.2-07-2014 "Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности".
10. Github: Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. — URL: <https://github.com> (Дата обращения 24.04.2023)
11. DbSchema: инструмент для автоматического запуска миграций баз данных. — URL: <https://dbschema.com> (Дата обращения: 15.05.2023)



