МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Курсовая работа

Разработка мобильного приложения для организации деятельности книжных клубов bookTalk

09.03.02 Информационные системы и технологии

Обработка информации и машинное обучение

Зав. кафедрой _____ Махортов С.Д., д.ф.-м.н., профессор

Обучающийся _____ Ларина М.С., 3 курс, д/о

Обучающийся _____ Федосеев Е.П., 3 курс, д/о

Обучающийся _____ Котов А.В., 3 курс, д/о

Руководитель _____ Тарасов В.С., ст. преподаватель

Воронеж 2024

Введение

Существует множество видов досуга, которые играют огромную роль в жизни человека. Без искусства в различном его проявлении, людям было бы очень тяжело справится с проблемами на работе и в быту. Но на наш взгляд то свободное время, которое человек тратит на чтение, ценнее всего. Не зря многие великие личности отмечали важность книги в жизни человека. Максим Горький однажды сказал: «Всем хорошим во мне я обязан книгам…».

Появление книжных клубов не является чем-то удивительным. Человек, прочитавший хорошую книгу, непременно захочет поделиться своими впечатлениями и обсудить интересующие детали. Организация собраний, посвященных различным произведениям, помогает единомышленникам не только встречаться друг с другом, но и развивать свои умственные и ораторские способности, а также расширять свой кругозор. Можно сказать, что книжный клуб – это тоже своего рода образовательная платформа.

На данный момент сфера книжных клубов находится в упадке. Организаторы книжных клубов столкнулись с тем, что многие желающие присоединиться просто не могут найти информацию о клубах, так как она разбросана по разным социальным сетям. Также развитие технологий и явное ускорение темпа жизни привело к тому, что книжные клубы, существующие только как личные встречи в установленном месте в установленное время, уже не подходят многим людям. Стоит отметить, что во многих городах просто нет книжных клубов, а поиск интернет-сообществ часто очень затруднителен опять же из-за огромного количества платформ, на которых можно вести деятельность клуба. Помимо этого, если нет структурированной системы записи на мероприятие, то участник должен сам позаботиться о добавлении этого мероприятия в свой планнер или ежедневник.

Отсутствие удобного приложения сказывается и на организаторах клубов. Вести деятельность сообщества в приложениях, не предназначенных

для книжного клуба, значит потратить огромное количество времени на ненужные технические детали.

Исходя из вышеперечисленных фактов, мы подумали, что было бы неплохо, если бы существовал сервис, в котором люди могли бы создавать книжные клубы и присоединяться к ним. Также там должна быть возможность планировать очные мероприятия, а для тех, кто не может присоединится к личной встречи или просто срочно хочет что-нибудь обсудить, должна быть возможность создания обсуждений и добавления комментариев. Для тех, кто решил прийти на личную встречу, должно быть организовано удобное отображение запланированных мероприятий.

Актуальность обусловлена необходимостью автоматизации процесса организации деятельности клуба и повышения удобства функционирования книжных клубов.

Целью курсовой работы является разработка мобильного приложения для организации деятельности книжных клубов.

1 Постановка задачи

1.1 Цели создания приложения

Целями создания приложения являются:

— автоматизация деятельности книжных клубов для повышения удобства организации работы клубов и поиска клубов;

— осуществление системы рекомендации клубов на основе интересов пользователя, что позволит увеличить клиентскую базу.

1.2 Задачи приложения

Приложение позволяет решать следующие задачи:

— просматривать ближайшие мероприятия клубов, в которых состоит авторизованный пользователь;

— просматривать мероприятия клубов, которые планирует посетить авторизованный пользователь;

— просматривать список книжных клубов, в том числе список рекомендованных клубов;

— вступать в клубы;

— участвовать в обсуждениях клуба, в котором состоит авторизованный пользователь, и создавать их;

— создавать клуб;

— осуществлять редактирование данных своего аккаунта после регистрации или авторизации.

1.3 Требования к приложению

1.3.1 Требования к приложению в целом

Данное приложение должно удовлетворять следующим основным требованиям:

— приложение должно корректно работать на устройствах, работающих на операционной системе Android 8.0 и новее;

— реализовывать все поставленные задачи.

1.3.2 Требования к функциям (задачам), выполняемым приложением

Разрабатываемое приложение должно соответствовать следующим функциональным требованиям:

Неавторизованный пользователь должен обладать возможностью:

— авторизоваться/зарегистрироваться в приложении;

— просматривать список рекомендуемых клубов;

— просматривать информацию о клубе (за исключением списка участников, списка ближайших мероприятий, списка обсуждений).

Авторизованный пользователь должен обладать возможностью:

— просматривать список клубов, в которых состоит;

— просматривать список рекомендуемых клубов;

— просматривать список клубов, которыми управляет;

— просматривать информацию о клубе, в котором не состоит (за исключением списка ближайших мероприятий);

— вступить в клуб;

— просматривать всю информацию о клубе, в котором состоит;

— участвовать в обсуждениях и создавать их в клубе, в котором состоит;

— просматривать список мероприятий и отмечать те, на которые пойдет;

— покинуть клуб;

— создать клуб;

— редактировать свой клуб;

— редактировать список ближайших мероприятий (добавлять, изменять, удалять);

— создавать, удалять и участвовать в обсуждениях своего клуба;

— удалить клуб;

— редактировать свой профиль;

— выйти из профиля.

1.3.3 Требования к оформлению и верстке страниц

Оформление и верстка страниц должны удовлетворять следующим требованиям:

— приложение должно быть оформлено в едином стиле;

— должно быть разработанное название, присутствующее в оформление страниц;

— приложение должно быть разработано в одной цветовой палитре с использованием ограниченного набора шрифтов;

— цветовая палитра должна быть контрастной;

— необходимо корректное и одинаковое отображение страниц на экранах различного размера.

1.3.4 Требования к защите информации

Для защиты информации будут использоваться JWT tokens. Даже если злоумышленник получит этот токен, с помощью которого он может получить доступ ко всем функциям приложения, через заданное количество времени токен не будет действителен (обычно это от 2 до 10 минут) и ему придется вылавливать новый.

1.4 Задачи, решаемые в процессе разработки

Были поставлены следующие задачи:

— Анализ предметной области;

— Обзор аналогов;

— Постановка задачи;

— Создание репозитория GitHub и доски в Trello;

— Разработка требований: к приложению в общем, к функциям, к структуре, к программному обеспечению, к оформлению и верстке страниц, к защите информации;

— Создание диаграмм: use case, состояний, активностей, последовательностей, IDEF0, сотрудничества, ER, классов, объектов, развертывания.

— Разработка дизайна приложения;

— Написание технического задания в соответствии с ГОСТ 34.602 – 2020;

— Реализация интерфейса приложения;

— Реализация серверной части приложения;

— Развертывание приложения;

— Написание курсовой работы.

2 Анализ предметной области

2.1 Глоссарий

В настоящей работе используются следующие термины и сокращения с соответствующими определениями:

— **Frontend** – это клиентская часть продукта (интерфейс, с которым взаимодействует пользователь).

— **Backend** – программно-аппаратная часть приложения (логика приложения, скрытая от пользователя).

— **Серверная часть** – это программа, которая обеспечивает взаимодействие клиента и сервера.

— **Сервер** – это устройство, в частности компьютер, которое отвечает за предоставление услуг, программ и данных другим клиентам посредством использования сети.

— **Клуб** – книжный клуб.

— **Встреча** – встреча участников книжного клуба.

2.2 Обзор аналогов

На русскоязычном рынке не существует приложения для организации деятельности книжных клубов, поэтому клубы в основном ведут свою деятельность в социальных сетях в виде групп и сообществ.

На зарубежном рынке есть несколько приложений для автоматизации деятельности книжных клубов. Далее мы рассмотрим их достоинства и недостатки, чтобы разработать функционал приложения, основываясь на том, чего не хватает пользователю в существующих решениях.

2.2.1 Bookclubs: Book Club Organizer

«Bookclubs: Book Club Organizer» – мобильное приложение для организации деятельности книжных клубов. Данная онлайн платформа предлагает возможность создавать свои книжные клубы или присоединяться к интересующим. Помимо этого, есть возможность вести список читаемых или прочитанных книг и отправлять личные сообщения пользователям.

Достоинства:

— Возможность создания нескольких клубов;

— Возможность создания запланированного мероприятия;

— Возможность вести списки прочитанных или читаемых книг;

— Наличие мобильного приложения под Android;

Недостатки:

— Многие ключевые возможности приложения (например, написание личных сообщений) доступны только по подписке;

— Отсутствует система обсуждений внутри клуба;

— Пользователь не может обозначить, что придет на мероприятие, а, следовательно, у пользователя нет возможности просматривать мероприятия, на которые он хочет пойти, в удобном формате;

— Отсутствует рекомендательная система клубов.

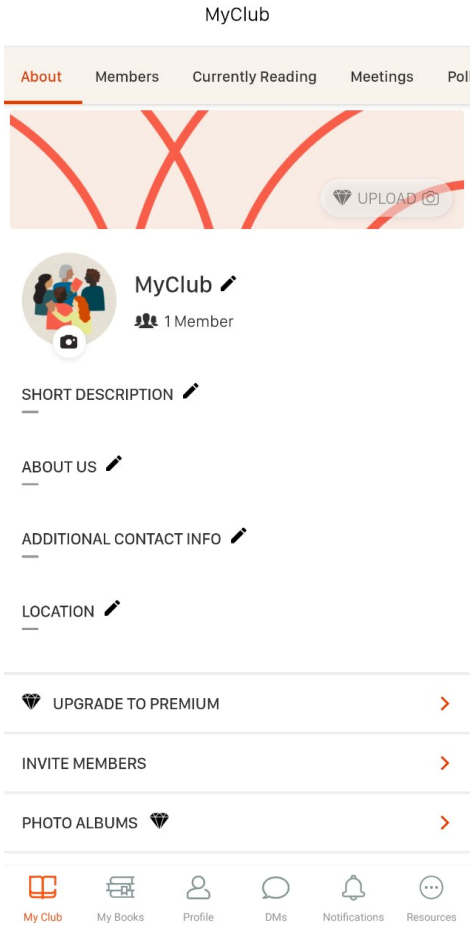

Рисунок 1 - Страница клуба в приложении Bookclubs

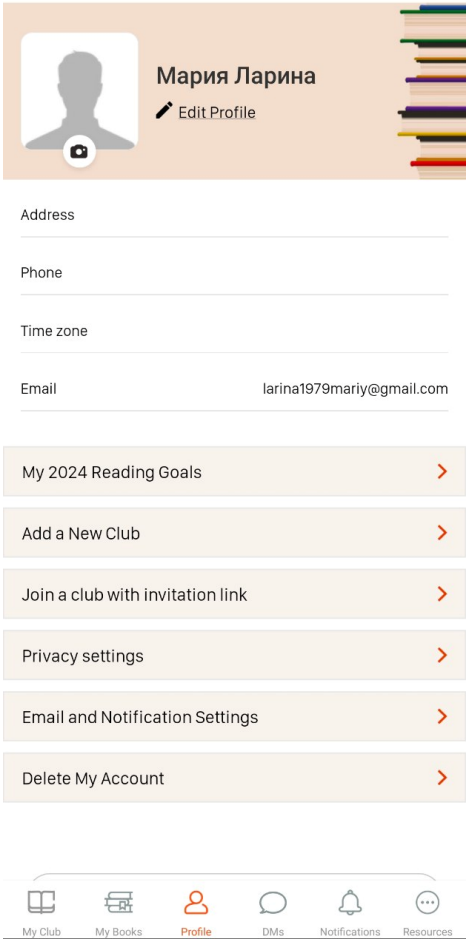

Рисунок 2 - Страница профиля приложения Bookclubs

2.2.2 Book Movement

«Book Movement» также является зарубежной онлайн платформой для организации деятельности книжных клубов. По своему функционалу почти полностью совпадает с «Bookclubs», за исключением отсутствия платной подписки.

Достоинства:

— Возможность создания нескольких клубов;

— Возможность создания запланированного мероприятия;

— Возможность вести списки прочитанных или читаемых книг;

— Рекомендательная система книг.

Недостатки:

— Пользователь не может обозначить, что придет на мероприятие, а, следовательно, у пользователя нет возможности просматривать мероприятия, на которые он хочет пойти, в удобном формате;

— Отсутствует рекомендательная система клубов;

— Отсутствует мобильное приложения под Android;

— Непонятный интерфейс.

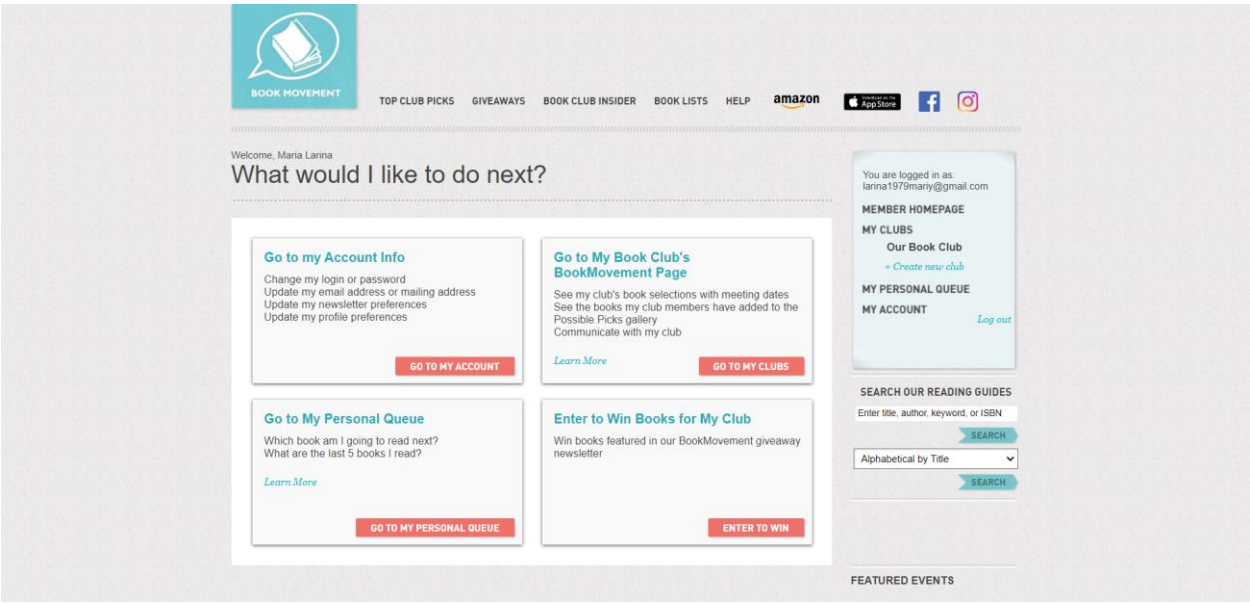

Рисунок 3 - Домашняя страница приложения Book Movement

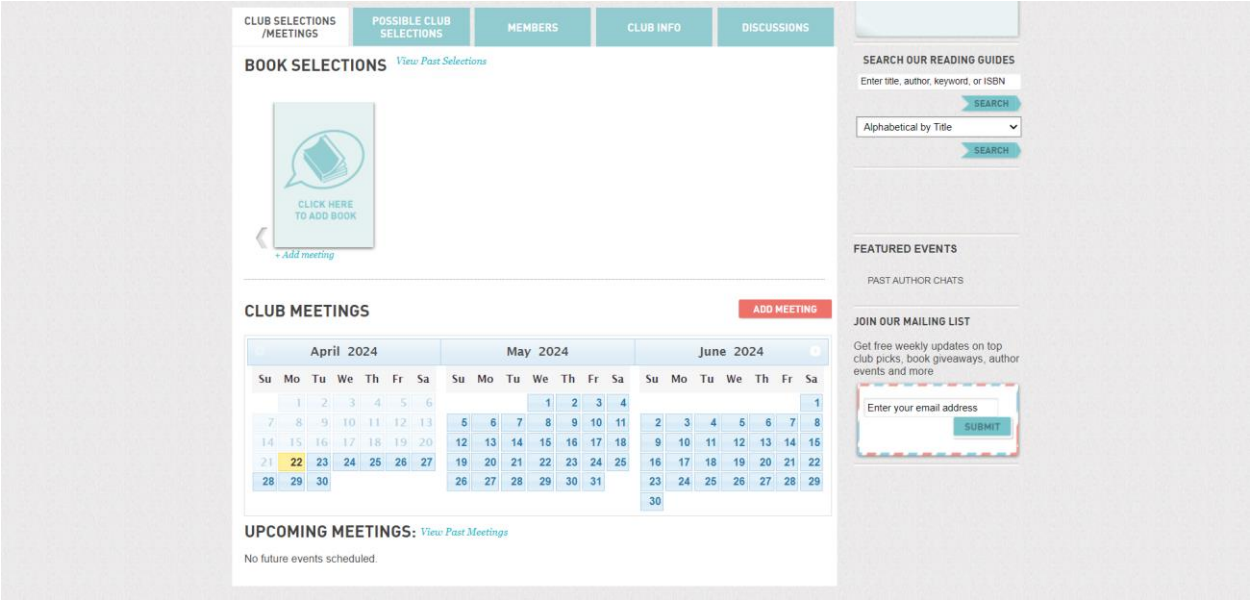

Рисунок 4 - Страница клуба приложения Book Movement

2.2.3 Booksloth и Bookship

Эти два приложения не организуют деятельность книжных клубов, а являются платформой для любителей книг, где они могут обсуждать прочитанные или читаемые книги.

Booksloth и Bookship – довольно устаревшие приложения, но мы рассматривали эти два приложения как косвенных конкурентов, так как основной причиной создания книжного клуба является желание общаться. Поэтому наше приложение также должно стать платформой для общения пользователей.

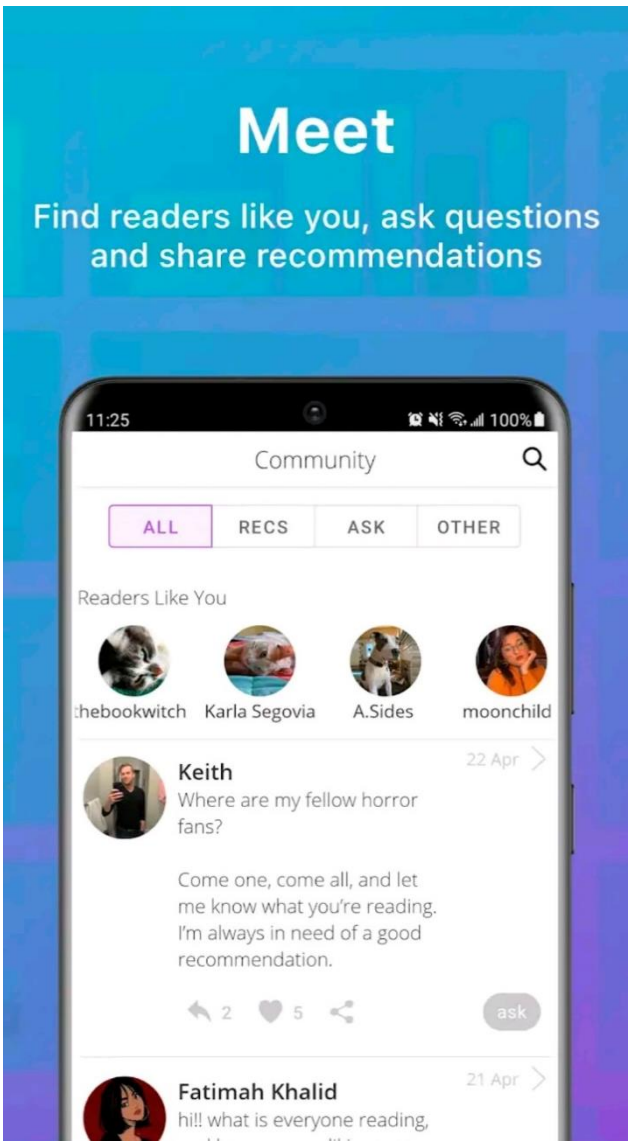

Рисунок 5 - Интерфейс приложения Booksloth

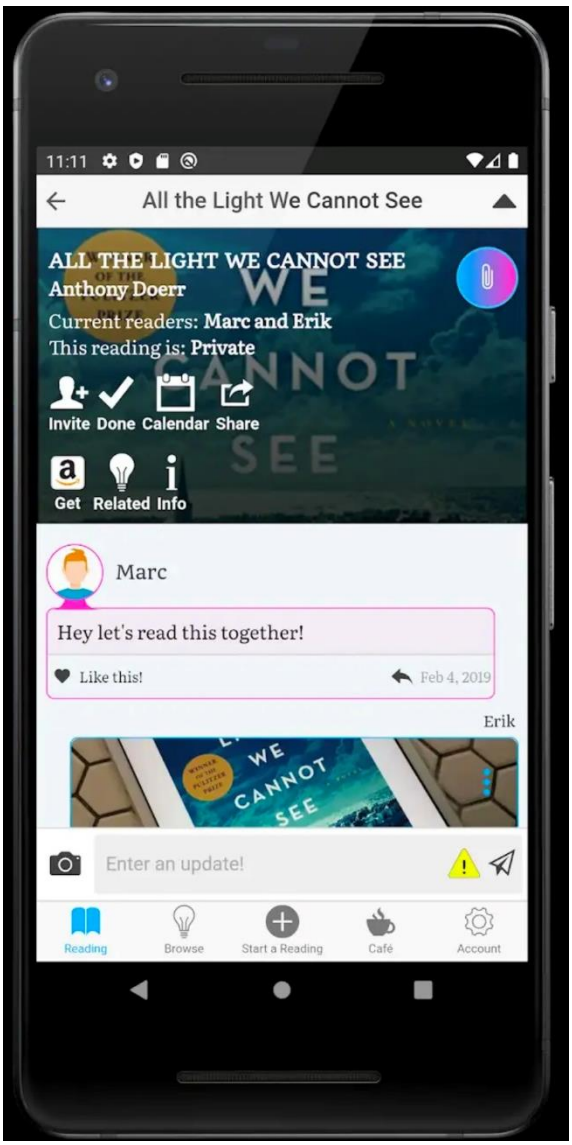

Рисунок 6 - Интерфейс Bookship

2.2.4 Вывод по обзору аналогов

На рисунке 7 представлена таблица сравнения аналогов по критериям.

Критерии / Приложения	Создание нескольких клубов	Запланированные мероприятия	Списки книг	Рекомендации клубов	Приложение под Android
Bookclubs: Book Club Organizer	+	+	+	−	+
Book Movement	+	+	+	−	−

Критерии / Приложения	Наличие подписки	Понятный интерфейс	Отслеживание личных мероприятий пользователя
Bookclubs: Book Club Organizer	+	−	−
Book Movement	−	−	−

Рисунок 7 - Обзор аналогов

3 Реализация

3.1 Средства реализации

Для реализации серверной части приложения будут использоваться следующие средства:

— язык программирования Python 12 версия;

— фреймворк Django Rest Framework;

— СУБД MySQL;

— инструмент для создания документации API Swagger.

Для реализации клиентской части приложения будут использоваться следующие средства:

— язык программирования Dart версия 2.19.0;

— Flutter SDK версия 3.7.6.

Для развертывания приложения будут использоваться следующие средства:

— клиент Certbot для создания и получения SSL сертификата;

— Docker для автоматизации развертывания приложения;

— Nginx – прокси-сервер с поддержкой SSl.

В качестве преимуществ выбранных технологий можно отметить следующее:

Для Python и Django:

— готовые решения для реализации RESTful архитектуры;

— удобные инструменты для работы с MySQL;

— готовые встроенные серверы (Tomcat), обеспечивающие ускоренное и более продуктивное развертывание приложений.

Для MySQL:

— предоставляет большой бесплатный функционал;

— надежная и высокопроизводительная.

Для Flutter:

— мультиплатформенность;

— понятная и полная документация;

— возможность быстро проектировать мобильные приложения.

3.2 Диаграмма прецедентов (Use Case)

Диаграмма прецедентов – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

На рисунке 8 представлена диаграмма прецедентов для разрабатываемого приложения bookTalk. На ней представлены 2 основные роли – неавторизованный пользователь и авторизованный пользователь. Также есть роль администратора (создателя клуба), которая наследуется от роли авторизованного пользователя.

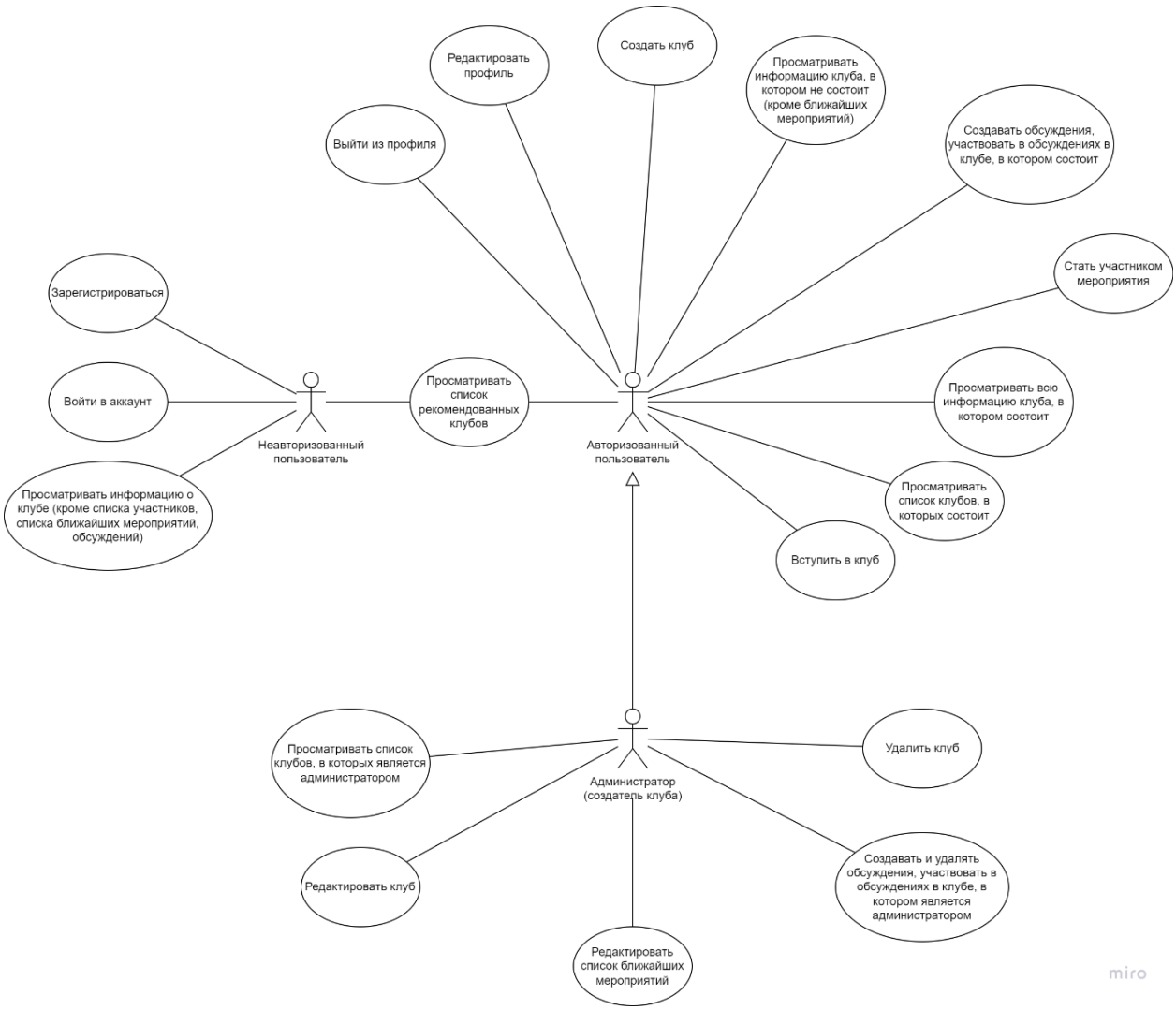

Рисунок 8 - Use Case

3.3 Диаграмма состояний (Statechart diagram)

Диаграмма состояний определяет последовательность состояний объекта, вызванных последовательностью событий. Данная диаграмма полезна при моделировании жизненного цикла объекта.

На рисунках 9 и 10 представлены диаграммы состояний для разрабатываемого приложения bookTalk для неавторизованного пользователя и авторизованного пользователя соответственно.

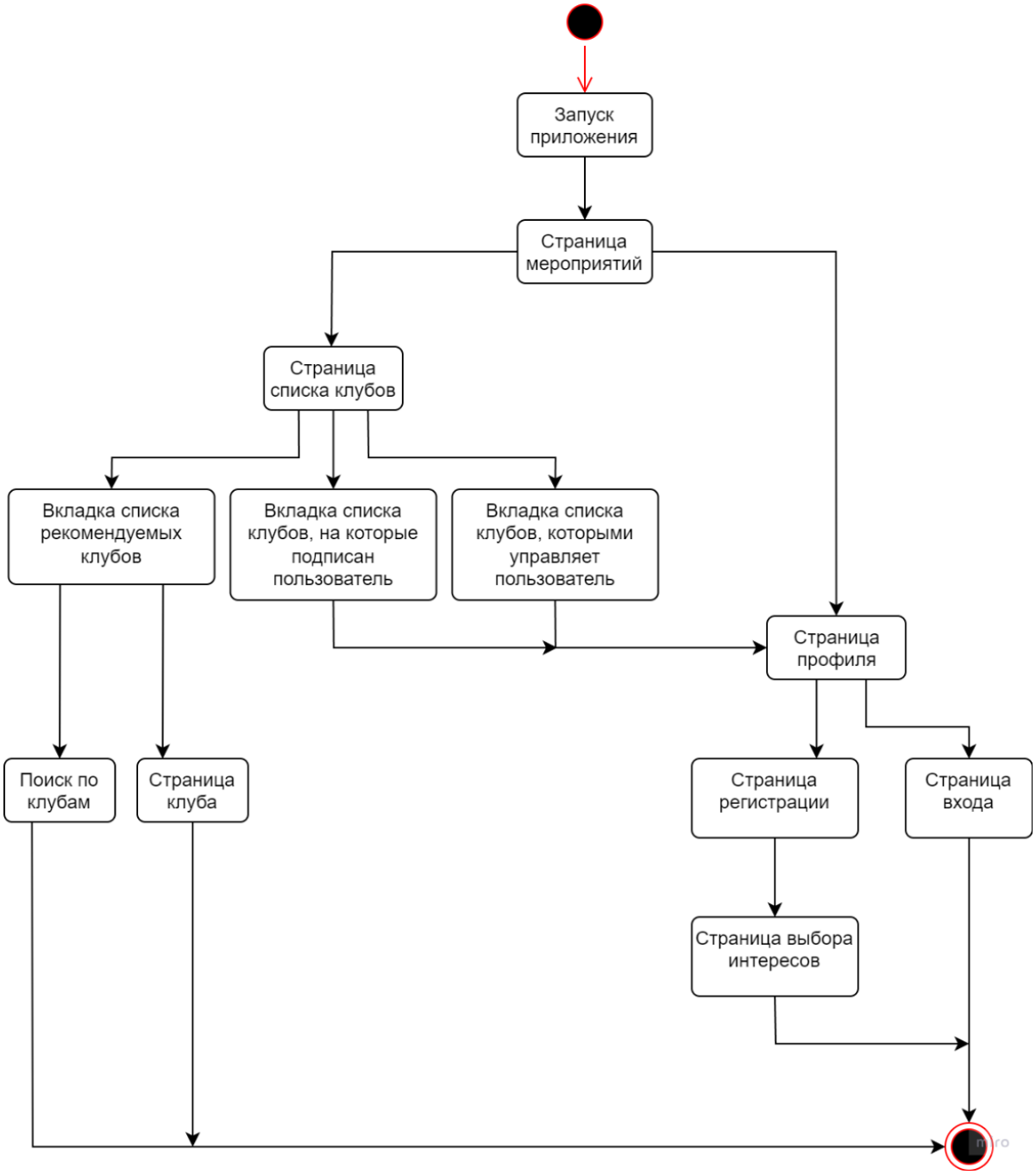

Рисунок 9 - Диаграмма состояний для неавторизованного пользователя

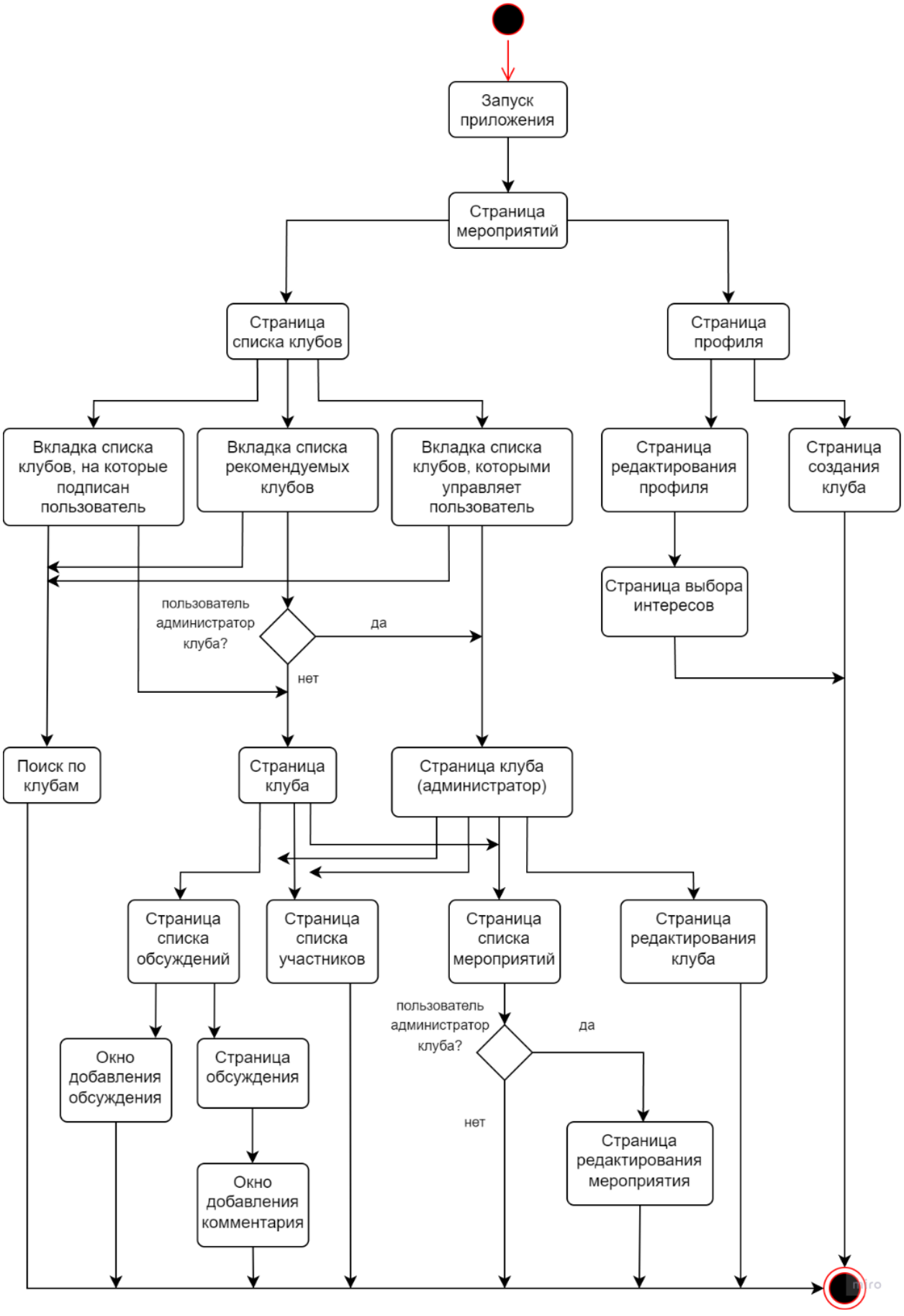

Рисунок 10 - Диаграмма состояний для авторизованного пользователя

3.4 Диаграмма активностей

Диаграмма активностей – UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний.

На рисунке 11 представлена диаграмма активностей для разрабатываемого приложения bookTalk.

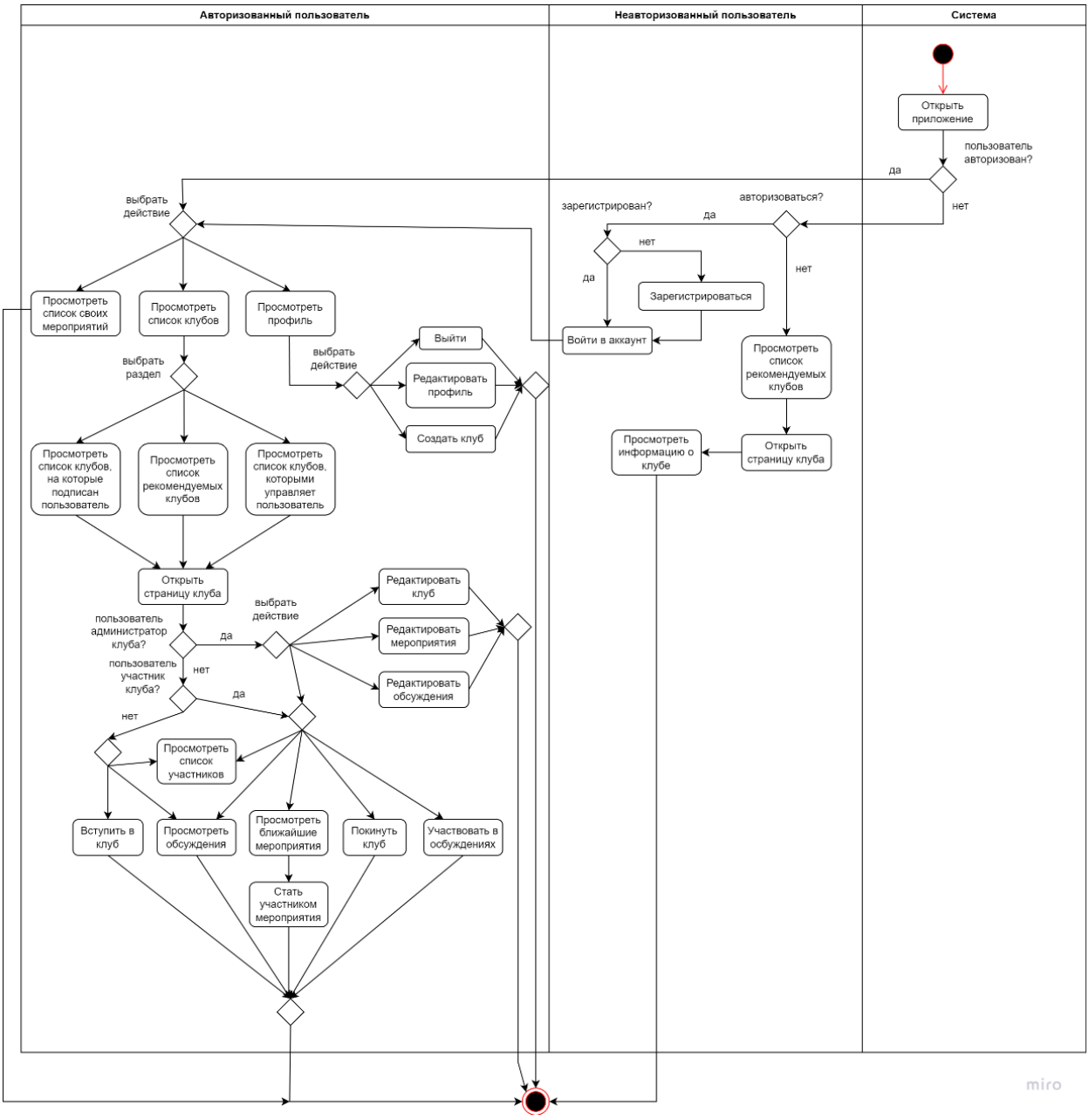

Рисунок 11 - Диаграмма активностей

3.5 Диаграмма последовательности (Sequence diagram)

Диаграмма последовательности – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) информационной системы в рамках прецедента.

На рисунках 12 и 13 представлены диаграммы последовательности для разрабатываемого приложения bookTalk для неавторизованного пользователя и авторизованного пользователя соответственно.

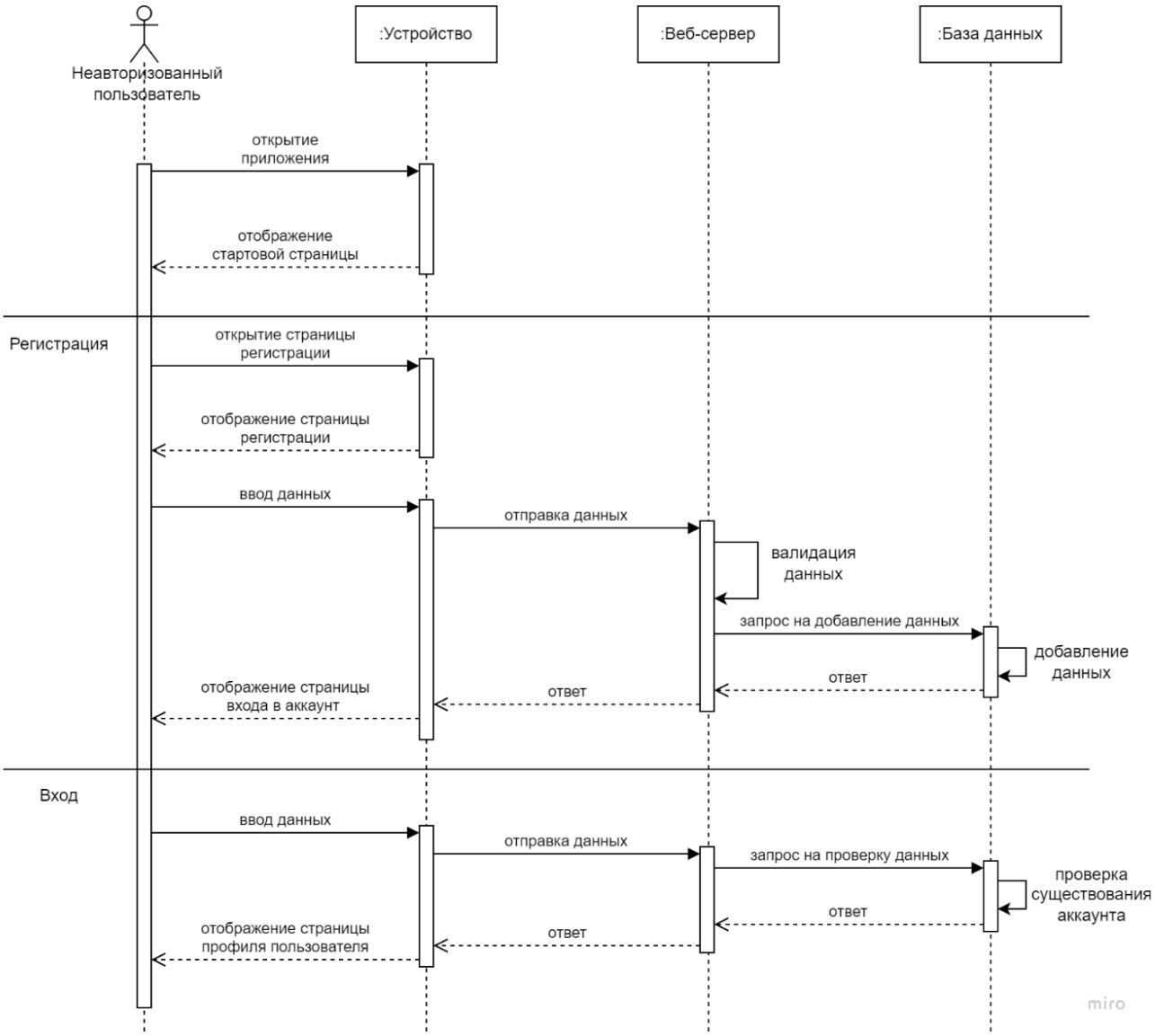

Рисунок 12 - Диаграмма последовательности для неавторизованного пользователя

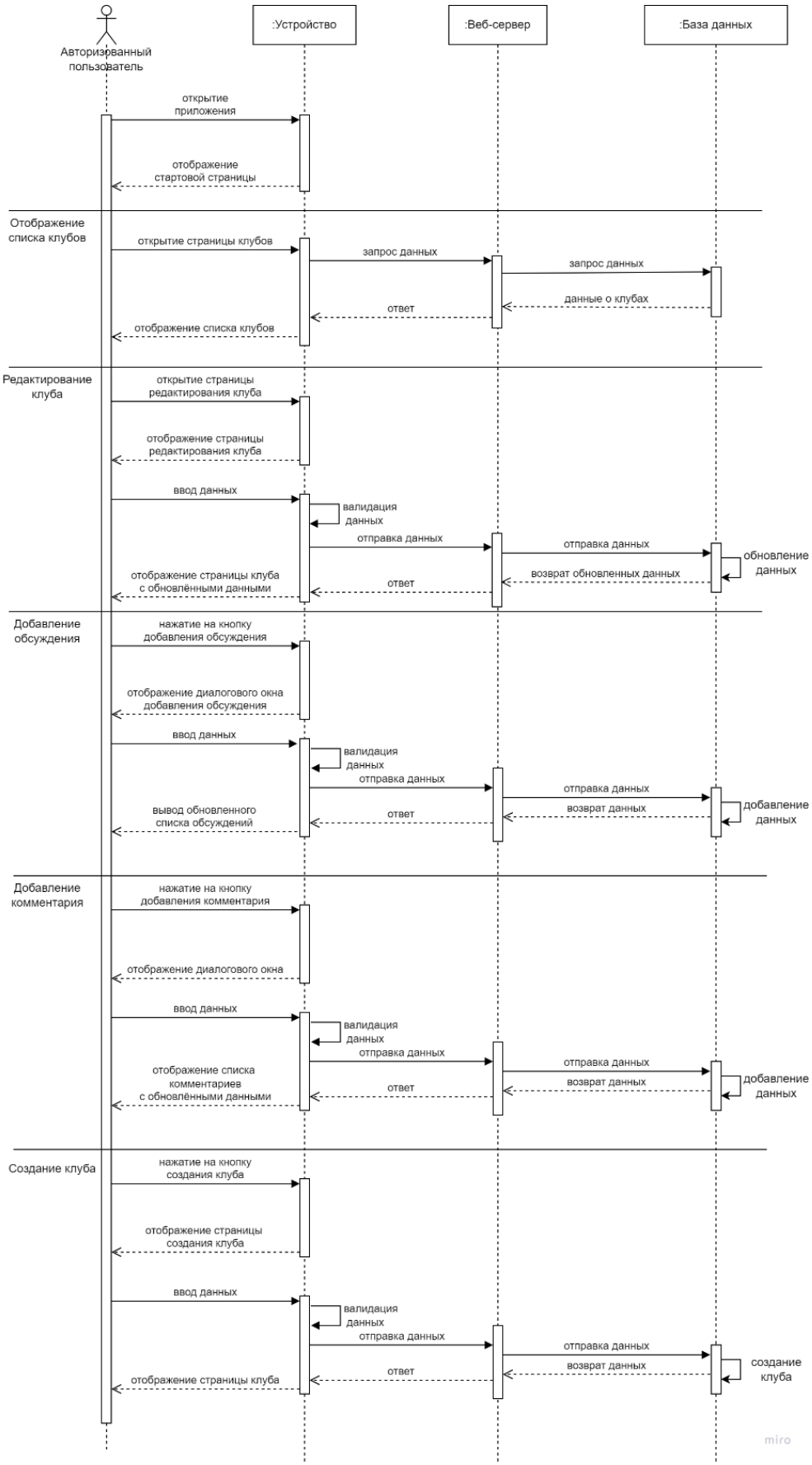

Рисунок 13 - Диаграмма последовательности для авторизованного пользователя

3.6 Диаграмма IDEF0

IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

На рисунках 14, 15 и 16 представлены диаграммы IDEF0 для разрабатываемого приложения bookTalk. На них отображена работа различных подсистем для создания клуба, участия в мероприятиях и добавления обсуждения.

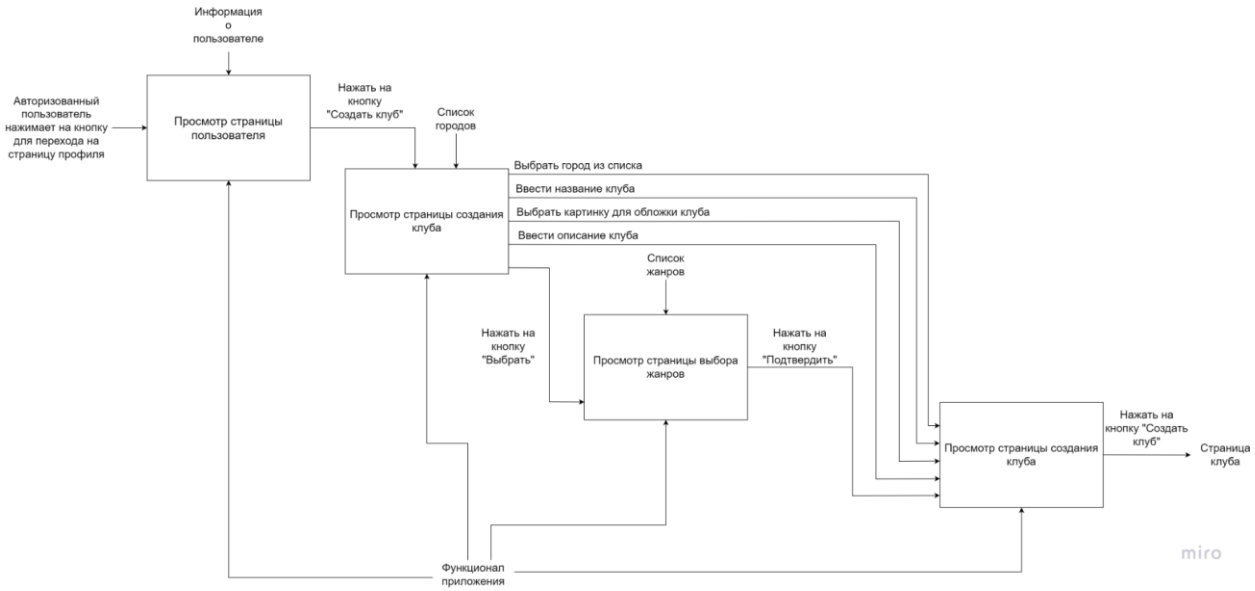

Рисунок 14 - Диаграмма IDEF0 (создание клуба)

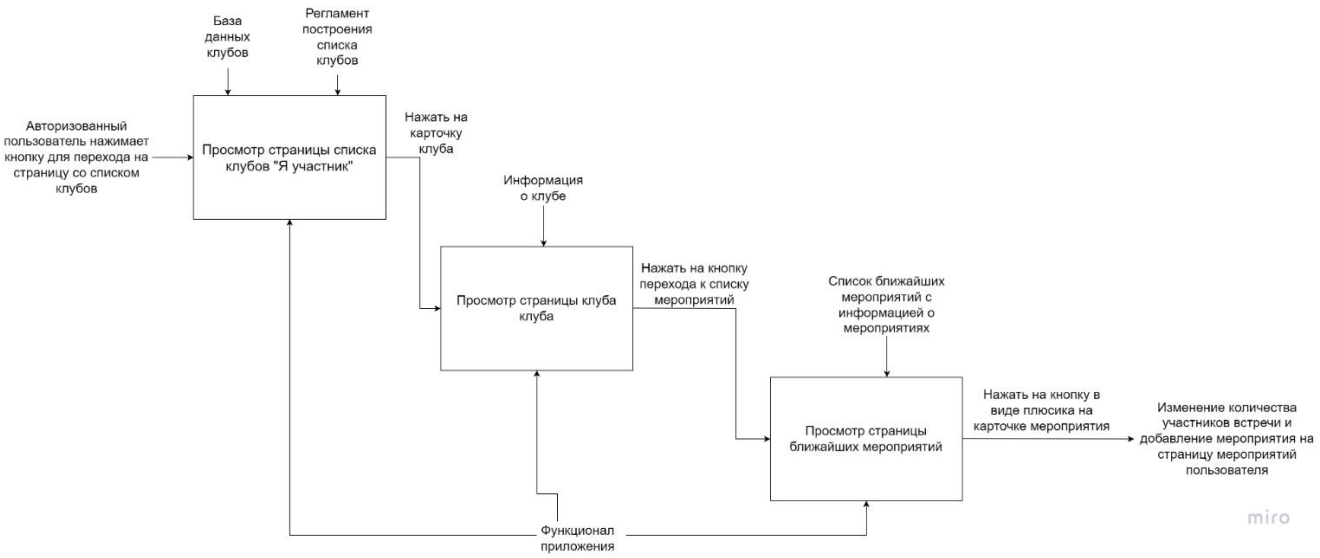

Рисунок 15 - Диаграмма IDEF0 (участие в мероприятии)

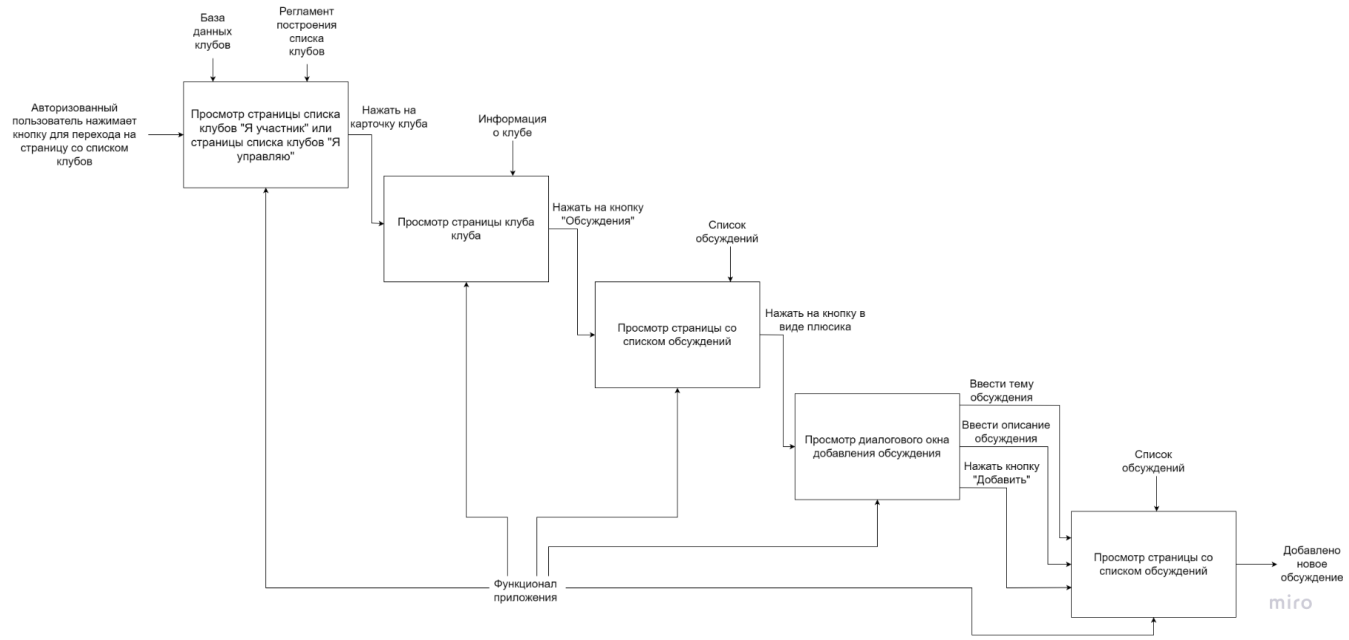

Рисунок 16 - Диаграмма IDEF0 (добавление обсуждения)

3.7 Диаграмма сотрудничества (Collaboration diagram)

Диаграмма сотрудничества – UML диаграмма, описывающая взаимодействие объектов, принимающих и отправляющих сообщения.

На рисунке 17 представлена диаграмма сотрудничества для разрабатываемого приложения bookTalk, описывающая взаимодействие объектов при создании клуба.

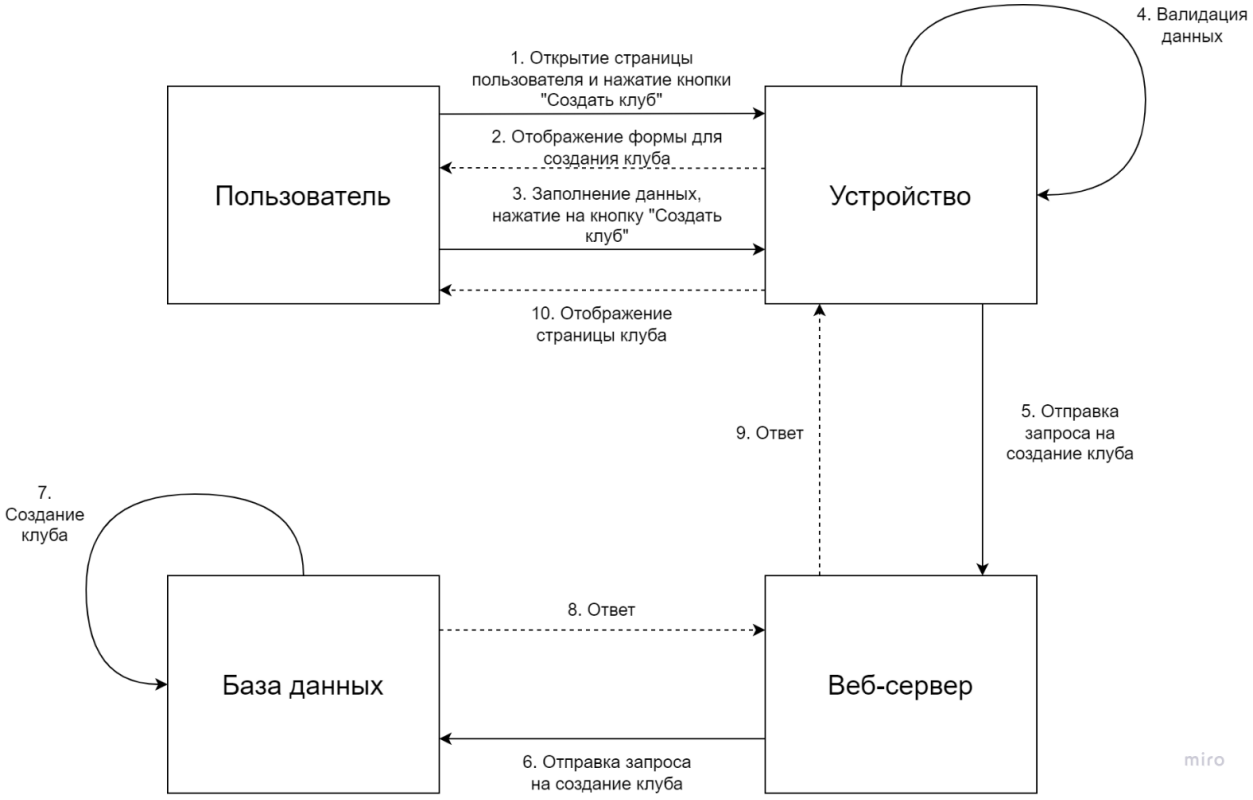

Рисунок 17 - Диаграмма сотрудничества

3.8 ER-диаграмма

ER-диаграмма – это модель данных, визуализирующая связь между «сущностями» внутри системы.

На рисунке 18 представлена ER-диаграмма для разрабатываемого приложения bookTalk.

3.9 Диаграмма классов (Class diagram)

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их

кооперацій, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

На рисунке 19 представлена диаграмма классов для разрабатываемого приложения bookTalk.

3.10 Диаграмма объектов (Object diagram)

Диаграммы объектов, иногда называемые диаграммами экземпляров, очень похожи на диаграммы классов. Как и диаграммы классов, они также показывают отношения между объектами, но в них используются реальные примеры. Они показывают, как будет выглядеть система в определенный момент времени. Поскольку в объектах имеются данные, они используются для объяснения сложных отношений между объектами.

На рисунке 20 представлена диаграмма объектов для разрабатываемого приложения bookTalk.

3.11 Диаграмма развертывания (Deployment diagram)

Диаграммы развертывания обычно используются для визуализации физического аппаратного и программного обеспечения системы. Она моделирует физическое развертывание артефактов на узлах.

На рисунке 21 представлена диаграмма развертывания для разрабатываемого приложения bookTalk.

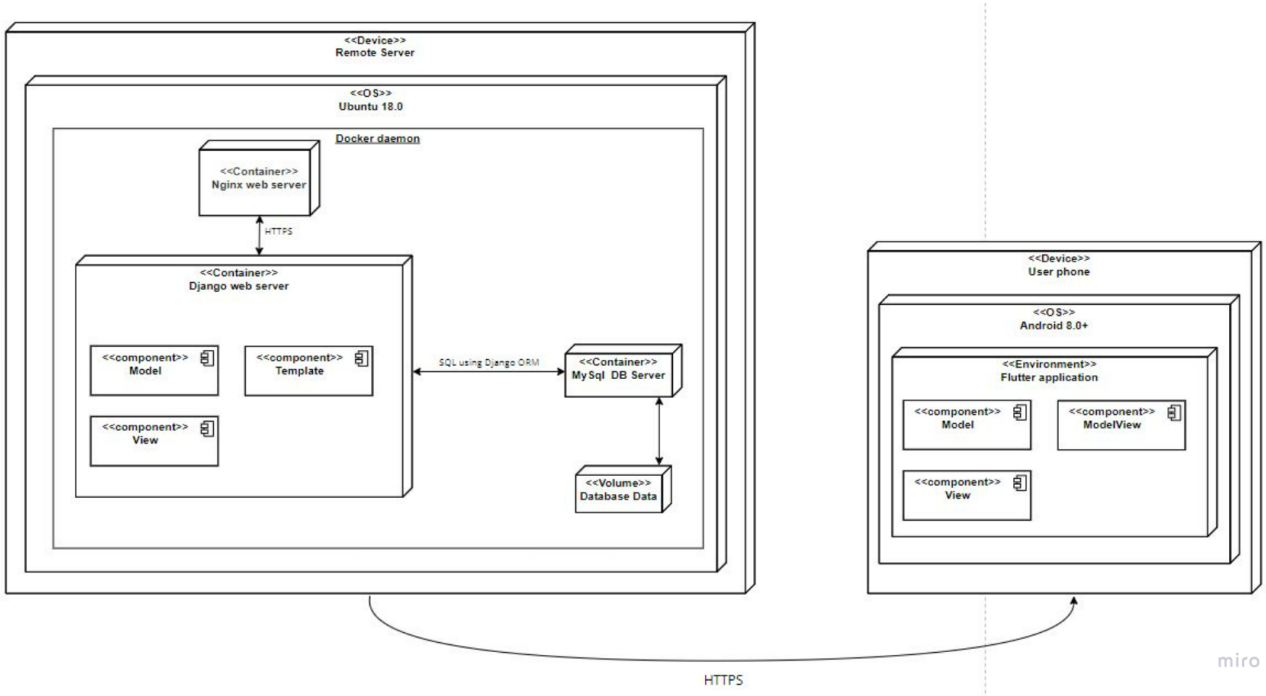

Рисунок 18 - Диаграмма развертывания

3.12 Архитектура клиентской части

Приложение реализовано в соответствии с подходом MVVM (Model – View – ViewModel). MVVM — это паттерн разработки, позволяющий разделить приложение на три функциональные части:

— Model – основная логика программы;

— View – вид или представление (пользовательский интерфейс);

— ViewModel – модель представления, которая служит прослойкой между View и Model.

Шаблон MVVM помогает четко отделять бизнес-логику приложения и логику презентации от пользовательского интерфейса. Поддержание четкого разделения между логикой приложения и пользовательским интерфейсом помогает решить многочисленные проблемы разработки и упрощает тестирование, обслуживание и развитие приложения.

3.13 Архитектура серверной части

Приложение реализовано в соответствии с подходом MVT (Model – View – Template) — паттерн разработки, разделяющий архитектуру приложения на три модуля: модель (Model), представление или вид (View), шаблон (Template).

— Model – этот компонент отвечает за бизнес-логику и данные приложения. Модели представляют собой структуры данных, которые определяют, как данные должны быть организованы и как они взаимодействуют друг с другом;

— View – отвечает за отображение данных пользователю. В контексте MVT, виды обрабатывают запросы от пользователя и возвращают ответы, используя данные, полученные от моделей. Виды могут быть представлены в виде функций или классов, которые определяют, какие данные должны быть отображены и как;

— Template – используется для определения структуры и формата представления данных. Шаблоны позволяют разделить логику представления данных от бизнес-логики и логики приложения, обеспечивая гибкость и повторное использование кода. Шаблоны могут включать в себя HTML, CSS и JavaScript для создания пользовательского интерфейса.