

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
« ____ » _____ 2024 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Приложение для онлайн книжного клуба

Руководитель	_____	_____	_____	Л.В. Макуха
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		Д.А. Коровина
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____		Л.В. Макуха
	<i>подпись</i>	<i>дата</i>		

Красноярск 2024

СОДЕРЖАНИЕ

Введение	3
1 Анализ предметной области	4
1.1 Обзор аналогов	5
1.1.1 LiveLib	5
1.1.2 Букмейт	6
1.1.3 BookClubs: Book Club Organizer	8
1.2 Постановка задачи. Функции приложения, требования.....	9
1.3 Анализ требований	10
1.4 Вывод по первой главе	11
2 Этап разработки приложения	12
2.1 Проектирование вариантов использования	12
2.2 Разработка баз данных	24
2.3 Разработка интерфейса	26
2.4 Вывод по второй главе	28
3 Реализация разработанного приложения	29
3.1 Авторизация и регистрация	29
3.2 База данных	30
3.3 Главная страница.....	32
3.4 Клубы	34
3.5 Читательский дневник.....	38
3.6 Профиль пользователя	39
3.7 Добавление изображения.....	40
3.8 Ридер	42
3.8 Вывод по главе	43
Заключени.....	44
Список используемых источников.....	45

ВВЕДЕНИЕ

Чтение книг с годами набирает все большую популярность. И не важно, трактаты великих мыслителей ли это или бульварные романы. На каждую книгу найдется своя аудитория. Но все люди такие разные, и найти человека с такими же интересами большая проблема. Поэтому люди стали объединяться в литературные сообщества, книжные клубы.

Книжный клуб – место, где люди собираются для совместных чтений, обсуждают свои мысли и впечатления, помогают увидеть, то, что могло ускользнуть во время прочтения. Благодаря таким сообществам можно погрузиться в мир книги полностью, узнать новые жанры, познакомиться с новыми людьми и полюбить чтение еще больше. Но интересы бывают настолько узкими, что найти собеседника в привычной среде достаточно затруднительно. Однако мир настолько велик, что нужное сообщество наверняка существует. Что и объясняет нарастающую популярность виртуальных книжных клубов.

Онлайн книжный клуб предлагает невероятную возможность объединиться с людьми, невзирая на часовой пояс и географическое положение. Главное – доступ в интернет и встреча состоится.

После активного обсуждения книги, чтобы погрузиться в сюжет еще сильнее и оформить собственные мысли за пределами коллектива, следует обратиться к читательскому дневнику.

Читательский дневник помогает собрать мысли после прочтения, запомнить какие-то интересные цитаты, запечатлеть эмоции. Через некоторое время будет приятно возвращаться к тем самым эмоциям и погружаться в воспоминания.

1 Анализ предметной области

Предмет деятельности онлайн книжного клуба - предоставить пользователям платформу, которая дает возможность обсудить интересные книги, приобрести новых знакомых, научиться формировать свои мысли и уверенно их высказывать. Используя расширенный поиск и персонализированные рекомендации возможно облегчить поиск книг для дальнейшего прочтения. Так же это помогает структурировать свой опыт чтения, что позволяет анализировать и рефлексировать, улучшать запоминание.

Был проведен опрос среди студентов, который позволил узнать актуальность данной темы и необходимый функционал (Рисунки 1-3).

Хотели бы вы участвовать в книжном клубе?
39 ответов

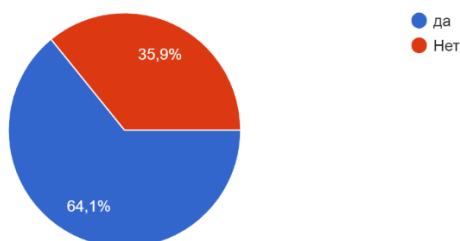


Рисунок 1 – Актуальность

Что для вас является наиболее привлекательным аспектом онлайн книжных клубов?
39 ответов

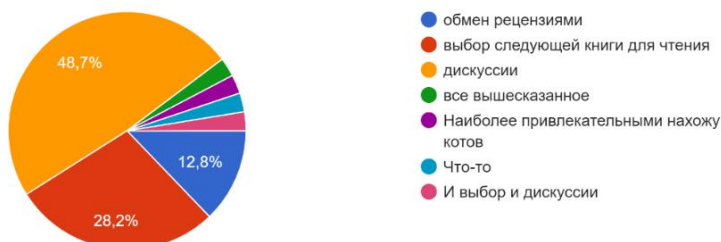


Рисунок 2 – Интересующие аспекты

Какая форма общения и участия в онлайн книжном клубе была бы для вас наиболее удобной
39 ответов

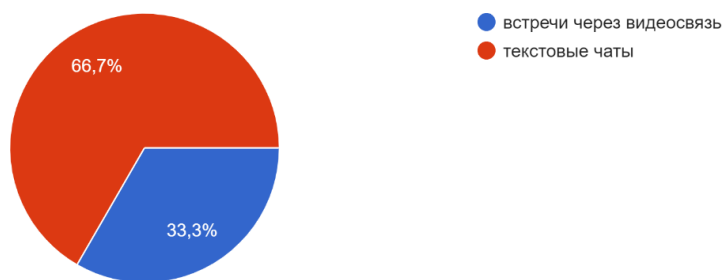


Рисунок 3 – Выявление удобного формата

Было выявлено насколько востребован онлайн книжный клуб. Студенты поделились наиболее важными для них аспектами и выбрали удобный формат. Также опрос показал, что притягивает и отталкивает пользователей от онлайн книжного клуба, например: лента рецензий и сложный интерфейс. Большинство опрошенных сошлись во мнении, что для них важно качественное общение и хороший коллектив, что невозможно без должной организации и определенного контроля в лице модератора.

1.1 Обзор аналогов

В главе (в этом разделе) представлено несколько аналогов с уже разработанными мобильными приложениями и разберем их плюсы и минусы, чтобы работа над приложением была более плодотворна.

1.1.1 LiveLib

«LiveLib» [1] представляет из себя многофункциональное приложение. Оно предоставляет рекомендации книг, подстроенных под интересы пользователя, различные категории поиска, блоги авторов, ленту рецензий и цитат, сообщества и много интерактивов и квестов.

Главная страница «LiveLib» представлена на рисунке 4.

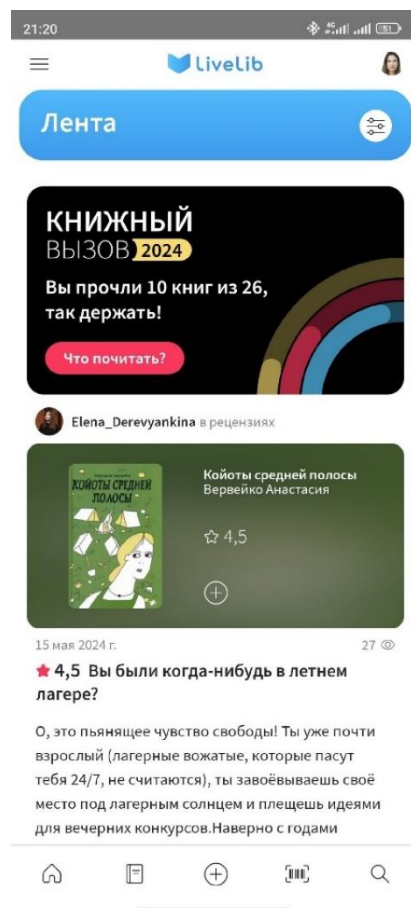


Рисунок 4 – Главная страница LiveLib

У приложения простой, понятный интерфейс, все интуитивно понятно. Можно составить свою книжную полку, вести свой блог, вести свою читательскую статистику, писать и читать отзывы, купить книгу. Можно найти друзей или связаться со своими единомышленниками.

Из минусов, есть проблемы с работой модераторов, не следящих за подлинностью рецензий, за их ненормативностью. Любой желающий может поставить книге оценку, даже если пользователь не читал книгу. Приложение иногда нестабильно и случаются сбои. Бывают проблемы со входом в аккаунт.

1.1.2 Букмейт

Букмейт – это приложение для чтения электронных книг и прослушивания аудиокниг. Букмейт предоставляет широкий спектр книг и предлагает обширный список рекомендаций. Приложение позволяет писать цитаты, оставлять отзывы

и впечатления. Также есть возможность загрузить книгу с устройства, при чтении текст и фон легко настраиваются.

Главная страница Букмейта показана на рисунке 5.

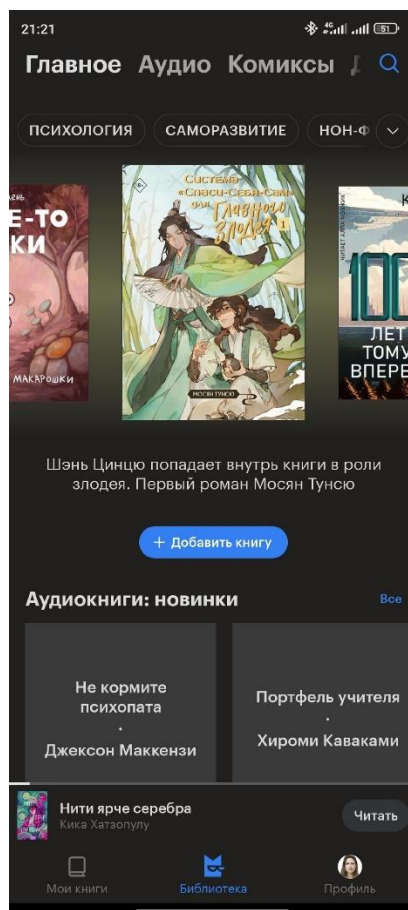


Рисунок 5 – Приложение Букмейт

Интерфейс приложения понятен и прост. Есть возможность поменять тему и иконку приложения. Можно составить свою подборку книг, подписаться на серию книг, при пополнении, придет оповещение. В Букмейте можно подписаться на пользователя и следить за его обновлениями. Большая лента рекомендаций, но по ней нет каталога. Многие функции доступны по платной подписке. Подписываясь на пользователя следить за его обновлениями неудобно, так как нигде не выводятся его новые цитаты или отзывы, необходимо заходить на его страницу.

1.1.3 BookClubs: Book Club Organizer

BookClubs: Book Club Organizer – англоязычное мобильное приложение помогающее в организации книжных клубов. Оно предоставляет возможность вступить в уже существующий клуб или создать свой.

Главная страница «BookClubs: Book Club Organizer» представлена на рисунке 6.



Рисунок 6 – Приложение «BookClubs: Book Club Organizer»

У приложения приятный интерфейс. Есть страничка с клубами, со списками книг, как прочитанных, так рекомендуемых, страничка с профилем, поиск книг и клубов, а также связь с разработчиками. BookClubs направлен как на офлайн встречи так и на онлайн. В нем можно поддерживать связь с участниками. Это приложение служит календарем для встреч, оповещая участников о встрече, и посредником для клуба и участников. Но при этом не все

функции бесплатные. Так как BookCluds англоязычное приложение, с плохим знанием языка будет сложно в нем ориентироваться и плодотворно проводить время.

1.2 Постановка задачи. Функции приложения, требования

Цель работы – разработать мобильное приложение для онлайн книжного клуба.

Разработанное приложение должно обладать следующими свойствами:

– наличие личного блога, в котором будут:

а) личные подборки;

б) рецензии и цитаты;

в) лента пользователя с подписками;

– возможность создавать и вступать в книжные сообщества;

– поиск по книгам, пользователям;

– наличие чатов в каждом из сообществ;

– наличие ленты с новостями;

– читательский дневник, в котором будут:

а) поля для основной информации о книге;

б) возможность вставить картинку;

в) поле для собственных интересов пользователя;

– возможность подписаться на пользователя и следить за обновлениями;

– возможность загрузить электронную книгу с устройства и читать в

приложении.

Анализируя вышеприведённые аналоги, можно сделать вывод о том, что человеческий фактор важен в наблюдении за работой клуба в целом. Поэтому обязательно должен присутствовать администратор, который будет следить за порядком, проверять рецензии, улаживать конфликты.

В рамках работы необходимо также решить следующие задачи – это провести анализ предметной области, осуществить проектирование мобильного приложения, включая разработку дизайна, создание диаграмм

последовательности и общей структуры приложения, а также определение структуры базы данных, разработать алгоритмы для обработки данных и преобразовать спроектированное приложение в рабочий код.

1.3 Анализ требований

Онлайн книжный клуб – портал, где любители чтения могут обмениваться мнениями, обсуждать книги и делиться литературными открытиями. В фокусе – создание дружелюбной и поддерживающей атмосферы, где каждый читатель чувствует, что его мнение ценно.

Сообщество поможет всем любителям чтения найти читательский уголок, который точно ему подойдет. Возраст, пол, род деятельности, ничего из этого не важно, главное, тяга к чтению.

Каждый пользователь имеет личную страничку, содержащую подборки, составленные пользователем, все написанные им рецензии и цитаты.

Предоставляется большая коллекция книг. Под каждой книгой можно поставить оценку, оставить рецензию и цитату.

Есть возможность добавить электронную книгу с устройства.

Читательский дневник – функция позволяющая углубиться в чтение, запомнить важные моменты и свое мнение о прочитанном. Он будет содержать поля: автор, название книги, жанры, цитаты, свои мысли, о чем книга, описание персонажей, глав и сцен, оценка книги, начало и конец прочтения.

Каждый пользователь может вступить в сообщества или создать его сам. В сообществе имеется чат, возможность оставлять записи на стене клуба для обозначения важной информации.

В приложении есть новостная лента, которая отображает важные события в книжном мире, статьи.

Приложение объединяет в себе многие полезные функции, что позволяет уменьшить количество приложений в телефоне и заменить бумажные варианты. Читая книгу в любом формате, можно будет ограничиться одним приложением.

Разработка мобильного приложения включает в себя несколько этапов: планирование и анализ, проектирование, разработка и тестирование.

Разработка структуры и макетов приложения включает в себя определение архитектуры интерфейса пользователя. Этот этап является важным для создания удобного и понятного интерфейса. Для этого используют различные программы, в этом проекте будет использоваться Figma.

Определив требования к продукту, начинается проектирование баз данных. Рассмотрев множество существующих баз данных и учитывая основные потребности в разработке, было решено выбрать Базу Данных Firebase.

Утвердив дизайн и требования начинается этап разработки. На этом этапе происходит создание приложения с использованием выбранных технологий и архитектурных решений. Для написания кода на языке Java используется Android Studio.

1.4 Вывод по первой главе

В первой главе были проведены исследования аналогов книжного клуба, а именно «LiveLib», «Букмейт» и «BookClubs: Book Club Organizer». В результате были выявлены их недостатки и преимущества. Также был проведен опрос, который позволил узнать актуальность данной темы и необходимый функционал.

После были сформулированы требования к разрабатываемому программному продукту. Они описывают основные преимущества каждого из рассмотренных продуктов, подчеркивая ключевые характеристики, представленные в описаниях. На основе требований были рассмотрены технологии необходимые для реализации продукта, а именно: среда разработки – Android Studio, язык программирования – Java, разработка дизайна в Figma, база данных – Firebase.

2 Этап разработки приложения

2.1 Проектирование вариантов использования

Исходя из возможностей системы была составлена диаграмма вариантов использования для всех типов пользователей. Диаграмма описывает какой функционал будет доступен пользователю, но не отражает того, как будет реализовано взаимодействия пользователя с системой. На рисунке 7 представлена диаграмма вариантов использования.

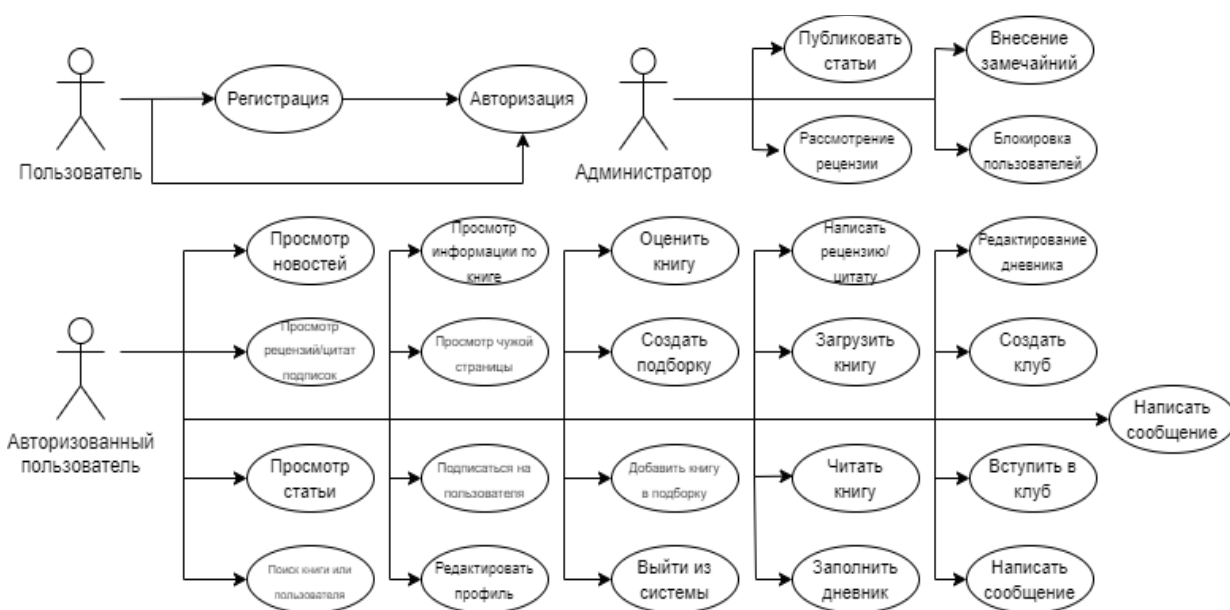


Рисунок 7 – Диаграмма вариантов использования

Название прецедента: регистрация.

Цель сценария: пройти регистрацию в приложении.

Предусловие: пользователь переходит в раздел регистрации.

Основной сценарий: после перехода на страницу регистрации пользователь заполняет форму и нажимает кнопку для ее отправки. Страница регистрации изображена на рисунке 8.



Рисунок 8 – Страница регистрации

Название прецедента: авторизация.

Цель сценария: пользователь входит в приложение и попадает на главную страницу.

Предусловие: пользователь должен быть зарегистрирован.

Основной сценарий: пользователь переходит на страницу авторизации, вводит номер телефона и пароль, после чего нажимает кнопку войти. Страница авторизации представлена на рисунке 9.

Постусловие: после успешного входа пользователю доступны функции личного кабинета.



Рисунок 9 – Страница входа

Название прецедента: посмотреть новостную ленту.

Цель сценария: пользователь просматривает статьи в новостной ленте, рецензии, цитаты.

Предусловие: пользователь находится на главной странице.

Основной сценарий: пользователь заходит на главный экран приложения (рисунок 10), в новостной вкладке отображаются статьи, рецензии, цитаты. Пользователь может открыть любую статью для прочтения.

Название прецедента: поиск книги или пользователя.

Цель сценария: поиск книги или страницы пользователя.

Предусловие: пользователь находится на «Главной» странице.

Основной сценарий: пользователь нажимает на иконку лупы в правом верхнем углу, открывается строка поиска, вводит текст.

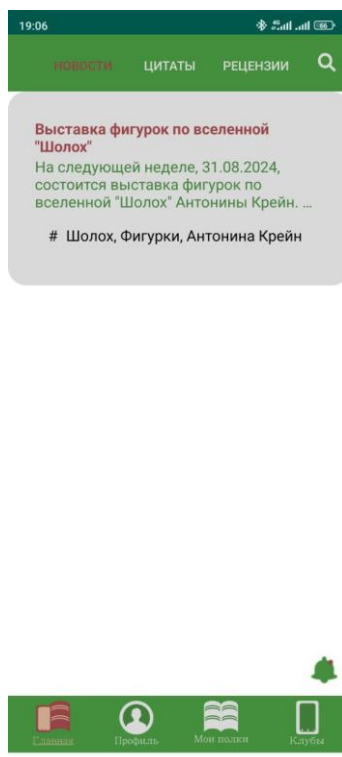


Рисунок 10 – Главная страница

Название прецедента: просмотреть статью.

Цель сценария: пользователь просматривает статью и оценивает ее.

Предусловие: пользователь находится на главной странице.

Основной сценарий: пользователь открывает статью (Рисунок 11) с главной страницы.

Название прецедента: публикация статей.

Цель сценария: опубликовать статью.

Предусловие: пользователь является администратором и находится на «Главной» странице.

Основной сценарий: администратор в правом нижнем углу нажимает на иконку плюса, пишет статью, нажимает кнопку опубликовать.

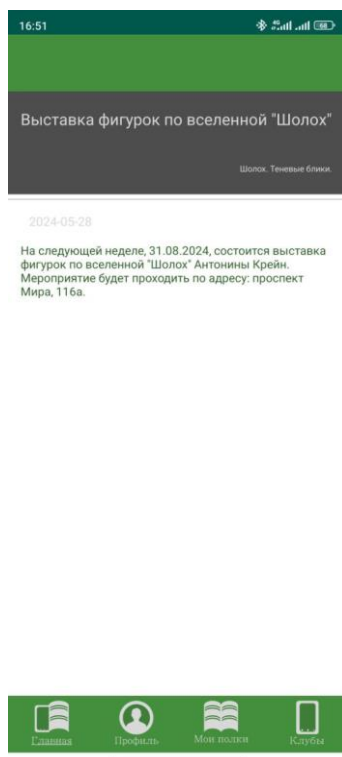


Рисунок 11 – Статья

Название прецедента: просмотр информации о книге.

Цель сценария: ознакомиться с книгой.

Предусловие: пользователь находится на странице книги (Рисунок 12).

Основной сценарий: пользователь ознакомливается с информацией о книге, смотрит в каких подборках она состоит, читает на нее рецензии, цитаты, пишет свои.

Название прецедента: добавление книги в подборку.

Цель сценария: добавить книгу в свою подборку.

Предусловие: пользователь находится на странице книги.

Основной сценарий: пользователь нажимает кнопку «Добавить» в левом верхнем углу и создает или выбирает в какую подборку добавить книгу.



Рисунок 12 – Информация о книге

Название прецедента: оценивание книги.

Цель сценария: оценка книги.

Предусловие: пользователь находится на странице книги.

Основной сценарий: пользователь нажимает то количество звезд, которую оценку хочет поставить.

Название прецедента: просмотр страницы другого пользователя.

Цель сценария: просмотр информации на странице другого пользователя.

Предусловие: пользователь зашел на страницу другого пользователя (Рисунок 13).

Основной сценарий: пользователь заходит на личную страницу другого пользователя, ознакомляется с информацией (его подборки, отзывы, цитаты), может подписаться или отписаться от него, нажав иконку человека в правом верхнем углу.

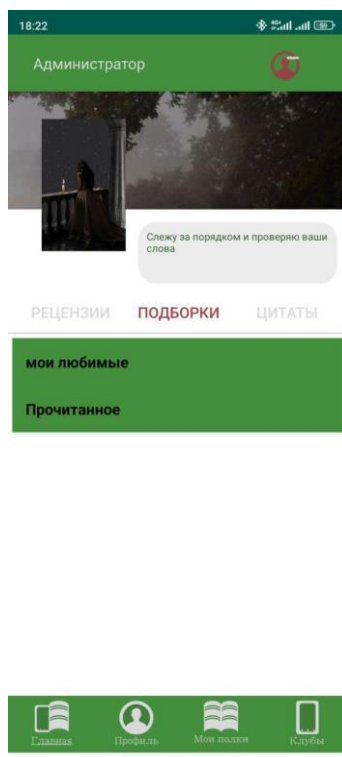


Рисунок 13 – Профиль другого пользователя

Название прецедента: редактирование своего профиля.

Цель сценария: пользователь изменяет информацию о себе.

Предусловие: пользователь находится на странице «Профиль» (Рисунок 14).

Основной сценарий: пользователь нажимает иконку карандаша в правом верхнем углу, переходит в редактирование профиля и меняет информацию.

Название прецедента: выход из системы.

Цель сценария: выйти из системы.

Предусловие: пользователь находится на странице «Профиль».

Основной сценарий: пользователь нажимает кнопку выход в правом верхнем углу.

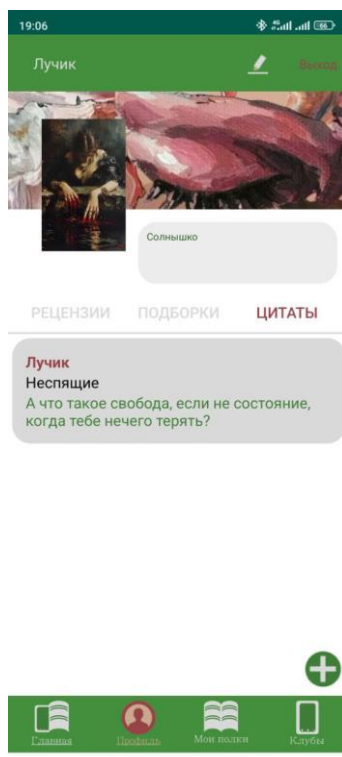


Рисунок 14 – Профиль пользователя

Название прецедента: написать рецензию/цитату.

Цель сценария: написать рецензию/цитату.

Предусловие: пользователь находится на странице книги или в своем профиле.

Основной сценарий: пользователь нажимает на карандаш в правом верхнем углу, на странице книги, или иконку плюса в правом нижнем углу, в своем профиле, выбирает что написать, рецензию или цитату, заполняет поля и отправляет.

Название прецедента: рассмотрение рецензии.

Цель сценария: одобрить или не одобрить рецензию.

Предусловие: пользователь является администратором и находится на «Главной» странице.

Основной сценарий: администратор в правом нижнем углу нажимает на иконку колокольчика, проверяет рецензию, отклоняет ее или одобряет, оставляя комментарий.

Название прецедента: редактировать отклоненные рецензии.

Цель сценария: отредактировать отклоненные отзывы.

Предусловие: пользователь зашел в список отклоненных отзывов через «Главную» страницу.

Основной сценарий: пользователь выбирает отклоненный отзыв, заходит в редактор и меняет текст опираясь на комментарий администратора.

Название прецедента: выгрузка электронной версии книги в приложение

Цель сценария: загрузить электронную книгу в приложение.

Предусловие: пользователь находится на странице «Мои полки» (Рисунок 15).

Основной сценарий: пользователь нажимает на иконку плюса в зеленом круге, расположенную в нижнем правом углу, выбирает электронную книгу с устройства, пишет название книги и добавляет в приложение.

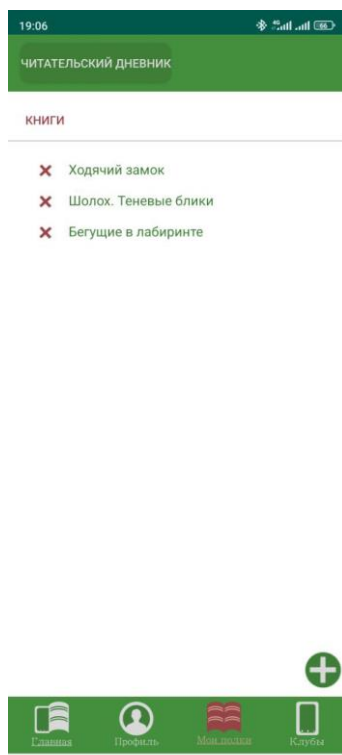


Рисунок 15 – Страница «Мои полки»

Название прецедента: чтение выгруженной книги.

Цель сценария: читать книги.

Предусловие: пользователь находится на странице «Мои полки».

Основной сценарий: пользователь выбирает книгу из списка на странице «Мои полки». На рисунке 16 показана выбранная книга.

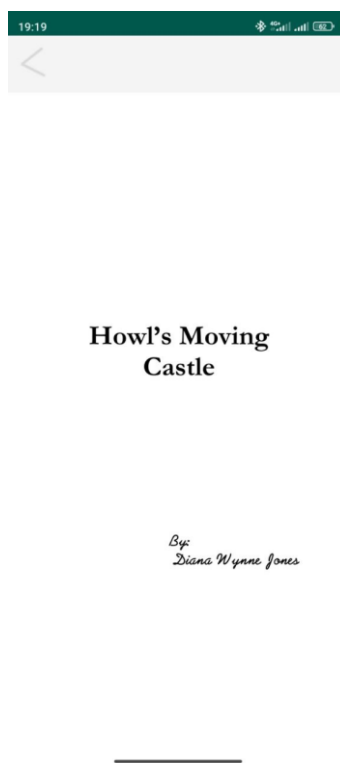


Рисунок 16 – Открытая электронная книга

Название прецедента: просмотр и редактирование уже существующих записей в читательском дневнике.

Цель сценария: посмотреть уже существующие записи.

Предусловие: пользователь находится на странице «Мои полки»

Основной сценарий: пользователь нажимает кнопку «Читательский дневник» в верхнем левом углу, переходит к списку читательских дневников, выбирает любой созданный им ранее читательский дневник.

Название прецедента: заполнить поля читательского дневника.

Цель сценария: ведение читательского дневника.

Предусловие: пользователь находится на странице со списком читательских дневников.

Основной сценарий: пользователь создает новый читательский дневник, нажимая на иконку плюса в правом верхнем углу, заходит на страницу

созданного читательского дневника (Рисунок 17) и записывает информацию в поля.

Рисунок 17 – Читательский дневник

Название прецедента: создание книжного клуба.

Цель сценария: создать книжный клуб.

Предусловие: пользователь находится на странице «Клубы» (Рисунок 18).

Основной сценарий: пользователь нажимает иконку карандаша в правом верхнем углу, заполняет необходимую информацию про клуб и создает его.

Название прецедента: вступление в книжный клуб.

Цель сценария: вступить в книжный клуб.

Предусловие: пользователь находится в нужном ему клубе.

Основной сценарий: пользователь нажимает иконку с человеком в правом верхнем углу и вступает в клуб

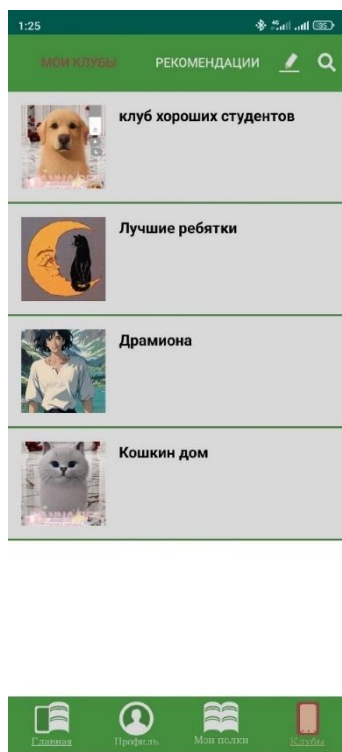


Рисунок 18 – Страница с книжными клубами

Название прецедента: писать сообщения.

Цель сценария: написать сообщение.

Предусловие: пользователь находится в нужном ему клубе (Рисунок 19).

Основной сценарий: пользователь нажимает иконку выноски в правом верхнем углу, пишет сообщение в чат клуба.

Название прецедента: писать записи на страницу клуба.

Цель сценария: написать записи на страницу клуба.

Предусловие: пользователь находится в нужном ему клубе.

Основной сценарий: пользователь нажимает иконку плюса в правом нижнем углу, пишет запись на страницу клуба.



Рисунок 19 – Страница клуба

2.2 Разработка баз данных

Для проекта была разработана схема базы данных (Рисунок 20).

Таблица «user» содержит следующую информацию о пользователе, такую как: пароль в зашифрованном виде, имя пользователя, адрес электронной почты, информация о пользователе, URL изображений для фото профиля и обложки.

Таблица «club» представляет собой клуб, в который может создать или вступить пользователь. В этой таблице хранится информация о клубе, имя клуба, идентификатор создателя, URL обложки клуба, время создания.

В таблице «book» есть название книги, идентификатор автора, написавший книгу, идентификатор жанра, URL обложки, информацию по книге, название книги.

Таблица «writing» содержит информацию о записях, оставленных на стене клуба. В ней есть текст записи, название записи, время создания, имя пользователя, написавшего ее, идентификатор пользователя, идентификатор клуба.

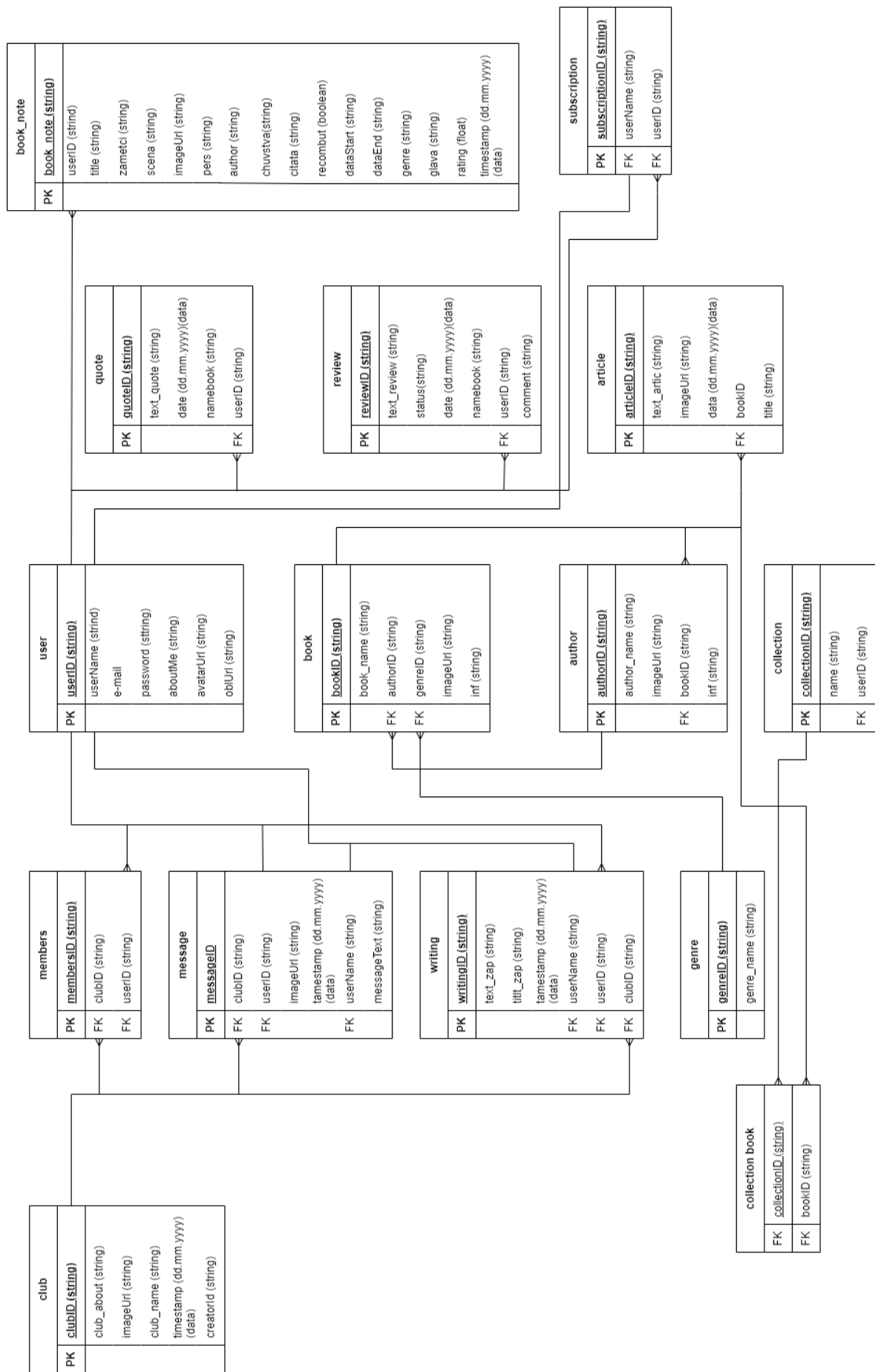


Рисунок 20 – Схема базы данных

Таблица «message» включает в себя идентификатор клуба, текст сообщения, URL изображения, идентификатор пользователя, который отправил сообщение, имя пользователя, дату и время отправки сообщения.

В таблице «members» хранятся идентификатор клуба и пользователя.

Таблица «collection» представляет собой подборку книг. В таблице есть идентификатор подборки и пользователя, создавшего его, название подборки.

Таблица «collection book» представляет собой книги вошедшие в подборку. В таблицу входят идентификаторы книги и подборки.

Таблица «article» представляет собой статью, в которой хранятся текст, URL изображения, дата публикации, название статьи и идентификатор книги.

Таблицы «quote» и «review» содержат в себе текст, дату публикации, название книги (необязательно книги, которая есть в приложении), к которой они относятся, статус (pending, approved, rejected) для рецензии и идентификатор пользователя, написавшего их.

Таблица «subscription» включает в себя имя и идентификатор подписчика.

В таблице «book_note» сохраняются данные полученные при заполнении читательского дневника, такие как: идентификатор создателя, название книги, заметки, описание любимой сцены, URL изображения обложки, описание любимого персонажа, автора, написавшего книгу, чувства при прочтении, цитаты, дату начала и конца прочтения, жанры, любимые главы, оценка пользователя, время создания.

В таблице «genre» хранятся названия жанров.

Таблица «author» содержит в себе имя автора, информацию об авторе, URL изображения автора, идентификатор написанных им книг.

2.3 Разработка интерфейса

Исходя из функционала приложения можно выделить список необходимых страниц:

- главная страница;
- страница авторизации;

- страница регистрации;
- страница пользователя;
- страница редактирования профиля;
- страница полок;
- страница ридера;
- страница всех читательских дневников;
- страница заполнения читательского дневника;
- страница всех книжных клубов;
- страница создания книжного клуба;
- страница книжного клуба;
- страница редактирования клуба;
- страница сообщений в книжном клубе;
- страница создания записи для клуба;
- страница информации о книге;
- страница со списком отклоненных рецензий;
- страница с не просмотренными рецензиями;
- страница подборки;
- страница написания рецензии/цитаты;
- страница редактирования рецензий;
- страница создания статьи.

Страница авторизации содержит форму, которая включает в себя адрес электронной почты и пароль. Со страницы авторизации можно перейти на страницу регистрации или главную страницу.

Страница регистрации содержит форму, которая включает в себя адрес электронной почты, имя пользователя и пароль. С этой страницы можно попасть обратно на страницу авторизации.

Главная страница – страница, на которую попадает пользователь после входа в свой аккаунт/приложение. На этой странице отображаются новости пользователя, рецензии и цитаты людей, на которых он подписан, кнопка для перехода на страницу со списком рецензий, не прошедших одобрение, для

администратора это кнопка перехода на страницу со списком не проверенных рецензий. Отдельно для администратора на этой странице также отображается кнопка добавления статьи.

На нижней панели расположено 4 кнопки, ведущие на главную страницу, в профиль пользователя, на полки пользователя и книжные клубы.

В профиле пользователя можно увидеть его фотографию, основную информацию, подборки, рецензии и цитаты пользователя.

Страница полки включает в себя кнопки для добавления электронной книги в приложение, для перехода к читательскому дневнику, для перехода к читаемой книге.

Страница ридера включает в себя текст книги.

Страница читательского дневника позволяет заполнить основную информацию о книге, поделиться впечатлениями, выбрать любимого персонажа и выписать заметки.

На странице всех книжных клубов можно посмотреть/вступить/создать книжные клубы.

После вступления можно перейти на страницу клуба общаться там, закреплять важную информацию.

На странице книги можно увидеть всю основную информацию, автора, дату выпуска, описание. Оценить книгу, оставить рецензию или цитату. Есть возможность добавить книгу в свою подборку. Посмотреть в каких подборках она состоит.

2.4 Вывод по второй главе

В этой главе были разработаны диаграммы прецедентов, которые показывают функциональные возможности пользователя и администратора в системе. Были рассмотрены сценарии для всех типов пользователей. Был разработан макет интерфейса для приложения. Была спроектирована структура базы данных.

3 Реализация разработанного приложения

После разработки макета приложения в Figma переносим его в Android Studio. Был использован плагин Export Kit. После переноса импортированных файлов в Android Studio, макеты были настроены под среду разработки. После этого необходимо добавить проект в Firebase, настроить сам проект, внести необходимые библиотеки.

3.1 Авторизация и регистрация

Для реализации авторизации и регистрации были разработаны отдельные классы, обращающиеся к базе данных.

На рисунке 21 представлен код для отправления запроса в базу данных для создания пользователя. Проверяется что строки с адресом и пароль не пустые, при помощи метода `createUserWithEmailAndPassword` создаем пользователя, сохраняем имя и переходим на страницу авторизации.

```
button.setOnClickListener(new View.OnClickListener() {
    @DKorovina
    @Override
    public void onClick(View v) {

        email = emailedt.getText().toString().trim();
        password = passwordedt.getText().toString().trim();
        if (email.isEmpty()){
            emailedt.setError("Введите e-mail");
        }
        if (password.isEmpty()){
            passwordedt.setError("Введите пароль");
        }else{
            @DKorovina
            auth.createUserWithEmailAndPassword(email,password).addOnCompleteListener (new OnCompleteListener<AuthResult>(){
                @DKorovina
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        Toast.makeText( context: _____activity.this, text: "SignUp Successful", Toast.LENGTH_SHORT).show();
                        String username = usernameedt.getText().toString().trim();
                        saveUsernameToFirestore(username);
                        startActivity(new Intent( packageContext: _____activity.this, _____activity.class));
                    }
                }
            });
        }
    }
});
```

Рисунок 21 – Регистрация пользователя

На рисунке 22 – код аутентификации для проверки наличия профиля пользователя в базе данных. При помощи метода `signInWithEmailAndPassword` проверяется почта и пароль, после переход на главную страницу.

```

button.setOnClickListener(new View.OnClickListener() {
    ± DKorovina
    @Override
    public void onClick(View v) {

        String email = emailedt.getText().toString();
        String password = passwordedt.getText().toString();

        if (!email.isEmpty() && Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            if (!password.isEmpty()) {
                auth.signInWithEmailAndPassword(email, password)
                ± DKorovina
                .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                    ± DKorovina
                    @Override
                    public void onSuccess(AuthResult authResult) {
                        Toast.makeText(context: _____activity.this, text: "Login Successful", Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(packageContext: _____activity.this, main_activity.class));
                        finish();
                    }
                })
                ± DKorovina
                .addOnFailureListener(new OnFailureListener() {
                    ± DKorovina
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        Toast.makeText(context: _____activity.this, text: "Login Failed", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        }
    }
});

```

Рисунок 22 – Аутентификация пользователя

На рисунке 23 Firebase Authentication, хранящая почту и пароль.

Identifier	Providers	Created ↓	Signed In	User UID
darya.korovina.02@mai...	✉	May 5, 2024	May 14, 2024	ribY5ouexChE9BzWafqGedoq...
cot@gmail.com	✉	May 2, 2024	May 2, 2024	8JJiABPFqAMYbWoSYdkYhtR...
daryagl02@gmail.com	✉	May 2, 2024	May 16, 2024	9X1MuvQ7mxZFEhfKQSNJ2C...

Rows per page: 50 1 - 3 of 3

Рисунок 23 – Firebase Authentication база данных

3.2 База данных

Для реализации проекта была создана база данных [8], содержащая как справочные материалы, такие как информация о книге или авторе, так и информацию о клубах, читательских дневниках и тд.

На рисунке 24 представлен способ хранения данных книги, пользователь не может вносить в нее изменения.

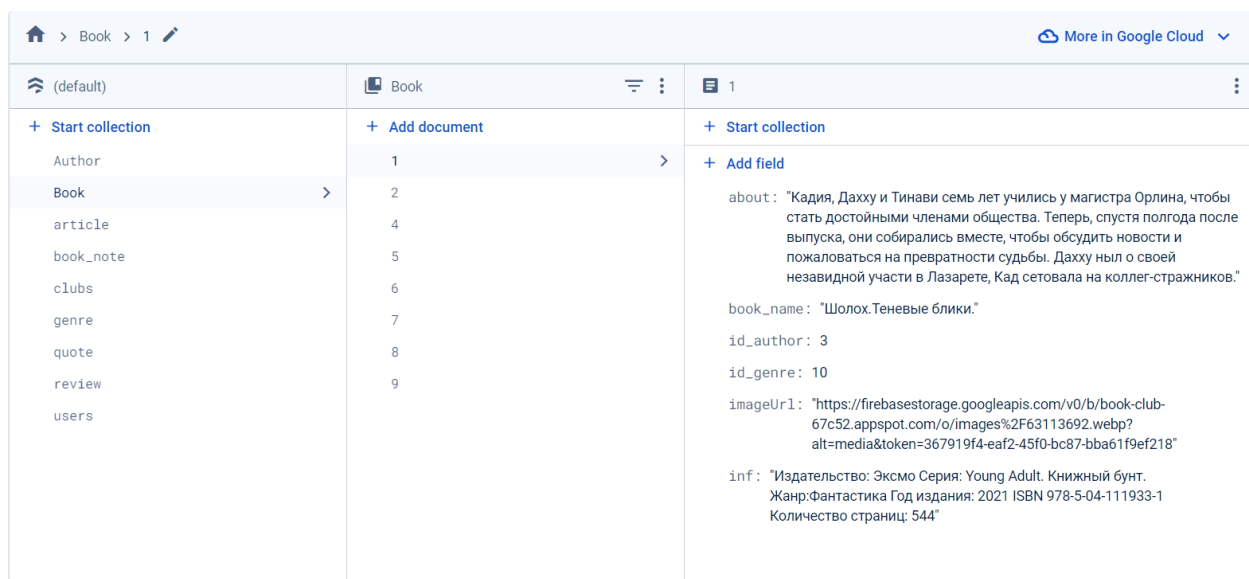


Рисунок 24 – Коллекция книг

На рисунке 25 представлен способ хранения информации о клубе, изменения сюда заносятся как пользователем, так и администратором. Здесь содержится вся основная информация о клубе, подколлекция сообщений, участников и записей.

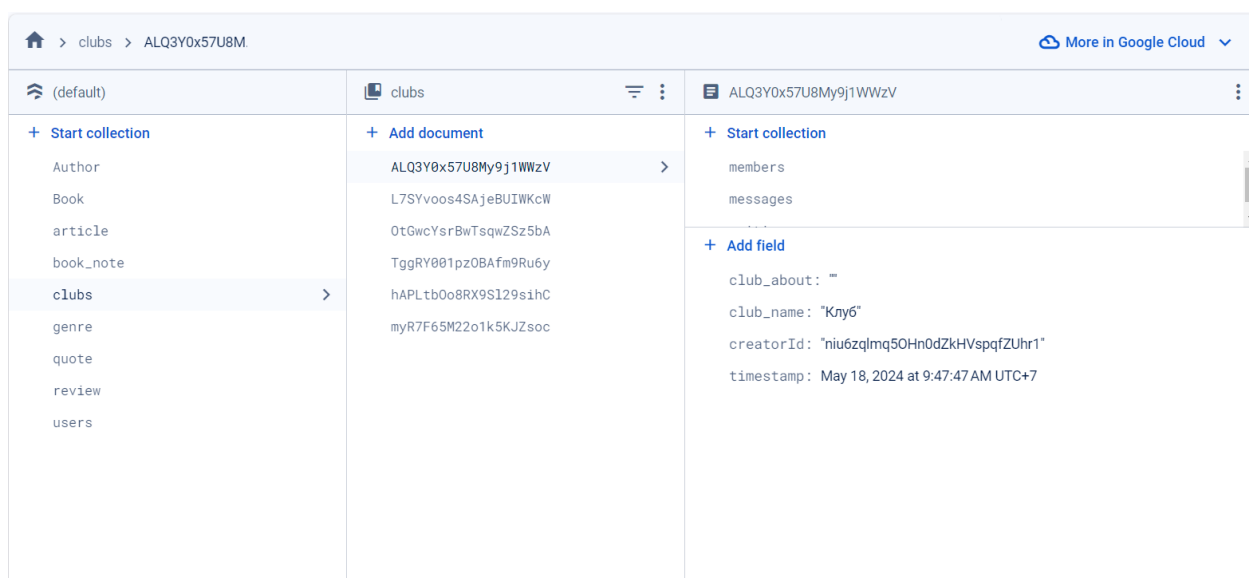


Рисунок 25 – Коллекция клубов

3.3 Главная страница

На главной странице отображаются: новости, которые выставляет система; цитаты и отзывы пользователей, на которых подписан пользователь. Также на главной странице возможно провести поиск книги или пользователя.

Для удобного поиска было использовано уменьшение регистра написанного в поле поиска текста и названия книг. На рисунке 26 видно, что это было реализовано благодаря методу `toLowerCase`. Для поиска по коллекции пользователей был реализован такой же метод.

```
db.collection( collectionPath: "Book") CollectionReference
    .get() Task<QuerySnapshot>
    new *
    .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
        new *
        @Override
        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                List<Map<String, String>> foundBooks = new ArrayList<>();
                for (QueryDocumentSnapshot document : task.getResult()) {
                    String bookName = document.getString( field: "book_name");

                    if (bookName.toLowerCase().contains(searchText.toLowerCase())) {
                        String bookId = document.getId();
                        Map<String, String> book = new HashMap<>();
                        book.put("name", bookName);
                        book.put("id", bookId);
                        foundBooks.add(book);
                    }
                }
            }
            updateBookList(foundBooks);
            updateVisibility();
        }
    });
```

Рисунок 26 – Поиск книги в базе данных

На рисунке 27 показан строка поиска и результат.

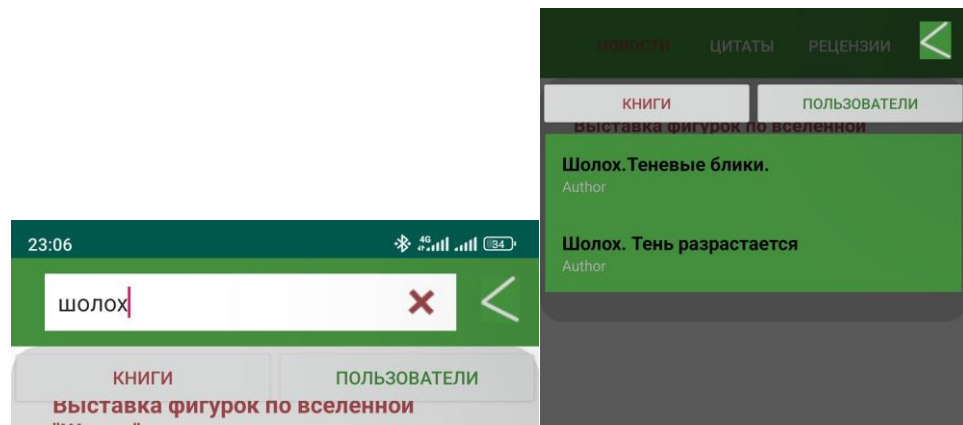


Рисунок 27 – Строка поиска и результат

Так же на главной странице у пользователя и администратора есть иконка колокольчика, которая меняется в зависимости от того есть ли у пользователя отклоненные рецензии, а у администратора не проверенные рецензии. Были установлены слушатели. На рисунке 28 «слушатель» добавляется к коллекции «review». Если среди документов, в которых задействован пользователь есть документ, где поле status равен rejected, то значек меняется (Рисунок 29). Для администратора значек меняется если в коллекции рецензий появляется документ со статусом pending.

```

reviewListener = db.collection( collectionPath: "review") CollectionReference
    .whereEqualTo( field: "userId", userId) Query
    new *
    .addSnapshotListener(new EventListener<QuerySnapshot>() {
        new *
        @Override
        public void onEvent(@Nullable QuerySnapshot snapshots, @Nullable FirebaseFirestoreException e) {
            if (e != null) {
                Log.w( tag: "Firestore", msg: "listen:error", e);
                return;
            }

            boolean hasPendingReviews = false;
            if (snapshots != null) {
                for (DocumentSnapshot doc : snapshots.getDocuments()) {
                    String status = doc.getString( field: "status");
                    if ("rejected".equals(status)) {
                        hasPendingReviews = true;
                        break;
                    }
                }
            }

            showNotificationDot(hasPendingReviews);
        }
    });

```

Рисунок 28 – Слушатель для коллекции «review»



Рисунок 29 – Метод для смены иконки

Если передается true ставится иконка с красной точкой, иначе обычная (Рисунок 30).



Рисунок 30 – Иконка без уведомления и с уведомлением

3.4 Клубы

На странице клубов отображаются списки клубов, в которых состоит/не состоит пользователь. Так же есть поиск по клубам. Пользователь может создать клуб (Рисунок 31), в последствии и редактировать.

При создании клуба в его документ, в коллекции «clubs», добавляется информация, вводимая пользователем и его идентификатор для того, чтобы в будущем он имел возможность вносить изменения в клуб, что не доступно обычному участнику. Так же сам создатель вносится в подколлекцию «members» для того, чтобы в дальнейшем было легче с поиском.

```

private void createNewClub() {
    String club_name = change_name.getText().toString();
    String club_about = change_about.getText().toString();
    String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();
    CollectionReference clubRef = db.collection( collectionPath: "clubs");
    if (club_name.isEmpty()) {
        Toast.makeText( context: creat_club_activity.this, text: "Введите название клуба", Toast.LENGTH_SHORT).show();
        return;
    } else {
        Map<String, Object> clubData = new HashMap<>();
        clubData.put("creatorId", uid);
        clubData.put("club_name", club_name);
        clubData.put("club_about", club_about);
        Timestamp timestamp = Timestamp.now();
        clubData.put("timestamp", timestamp);

        clubRef.add(clubData)
            .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                new *
                @Override
                public void onSuccess(DocumentReference documentReference) {
                    String docId = documentReference.getId();
                    Map<String, Object> userData = new HashMap<>();
                    userData.put("userId", uid);
                    db.collection( collectionPath: "clubs").document(docId) DocumentReference
                        .collection( collectionPath: "members") CollectionReference
                        .document(uid) DocumentReference
                        .set(userData) Task<Void>
                }
            })
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                new *
                @Override
                public void onSuccess(Void aVoid) {
                    newDocRef = db.collection( collectionPath: "clubs").document(docId);
                    uploadAvatar(filePath, newDocRef);
                    startActivity(new Intent( packageContext: creat_club_activity.this, club_main_activity.class));
                }
            });
    }
}

```

Рисунок 31 – Создание клуба

Заходя в клуб у пользователя, есть возможность вступить в него (Рисунок 32). При вступлении происходит поиск идентификатора пользователя по подколлекции «members», чтобы уточнить точно ли пользователь не является членом клуба. Если пользователь не является участником его идентификатор и дата вступления записываются в документ в подколлекции. Так же пользователю выводится табличка, уведомляющая об успешном вступлении.

```

public void onClickDobaw(View View) {
    DocumentReference clubMemberRef = db.collection( collectionPath: "clubs").document(clubId)
        .collection( collectionPath: "members").document(userId);
    Map<String, Object> memberData = new HashMap<>();
    memberData.put("userId", userId);
    Timestamp timestamp = Timestamp.now();
    memberData.put("timestamp", timestamp);

    new *
    clubMemberRef.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
        new *
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
            if (task.isSuccessful()) {
                DocumentSnapshot document = task.getResult();
                if (document.exists()) {
                    Toast.makeText( context: club_activity.this, text: "Пользователь уже добавлен в клуб", Toast.LENGTH_SHORT).show();
                } else {
                    new *
                    clubMemberRef.set(memberData).addOnCompleteListener(new OnCompleteListener<Void>() {
                        new *
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()) {
                                Toast.makeText( context: club_activity.this, text: "Успешно добавлено в клуб", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                }
            }
        }
    });
}

```

Рисунок 32 – Вступление в клуб

После вступления появляются кнопки для перехода в чат и для создания записи на стене сообщества, методы для отправки сообщений и публикаций записей на стене представлены на рисунках 33 и 34 соответственно.

Пользователь набирает сообщение и при отправке текст сообщения, идентификатор и имя пользователя, время отправки записываются в подколлекцию «message», которая находится в документе этого клуба. Имя пользователя необходимо для отображения сообщения в чате.

```

private void sendMessage(String username) {
    String textMess = textMess.getText().toString();
    if (!textMess.isEmpty()) {
        Map<String, Object> messageData = new HashMap<>();
        messageData.put("userId", uid);
        messageData.put("userName", username);
        messageData.put("messageText", textMess);
        messageData.put("timestamp", FieldValue.serverTimestamp());
        db.collection( collectionPath: "clubs").document(clubId) DocumentReference
            .collection( collectionPath: "messages") CollectionReference
            .add(messageData) Task<DocumentReference>
            new *
            .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                new *
                @Override
                public void onSuccess(DocumentReference documentReference) {
                    Log.d( tag: "Firestore", msg: "DocumentSnapshot written with ID: " + documentReference.getId());
                }
            })
            new *
            .addOnFailureListener(new OnFailureListener() {
                new *
                @Override
                public void onFailure(@NonNull Exception e) {
                    Log.w( tag: "Firestore", msg: "Error adding document", e);
                }
            })
    });
}

```

Рисунок 33 – Добавление сообщения в базу данных

Нажав на странице клуба на иконку добавления записи переход на страницу для создания записи. Перед сохранением записи в базе данных проверяются поля для записи заголовка и основного текста. Если поля пустые запись не будет сохранена. После создания данные записываются в подколлекции «writing».

```

private void makeZapis(String username) {
    String TitleZapis = title_zapis.getText().toString();
    String TextZapis = text_zapis.getText().toString();
    if (!TitleZapis.isEmpty() && !TextZapis.isEmpty()) {
        Map<String, Object> zapData = new HashMap<>();
        zapData.put("userId", uid);
        zapData.put("userName", username);
        zapData.put("title_zapis", TitleZapis);
        zapData.put("text_zapis", TextZapis);
        zapData.put("timestamp", FieldValue.serverTimestamp());

        db.collection( collectionPath: "clubs").document(clubId) DocumentReference
            .collection( collectionPath: "writing") CollectionReference
            .add(zapData) Task<DocumentReference>
            new *
            .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                new *
                @Override
                public void onSuccess(DocumentReference documentReference) {
                    Log.d( tag: "Firestore", msg: "DocumentSnapshot written with ID: " + documentReference.getId());
                }
            })
            new *
            .addOnFailureListener(new OnFailureListener() {
                new *
                @Override
                public void onFailure(@NonNull Exception e) {
                    Log.w( tag: "Firestore", msg: "Error adding document", e);
                }
            })
    });
}

```

Рисунок 34 – Создание записи на стену клуба

3.5 Читательский дневник

На странице «Полки» можно перейти к читательским дневникам. Создание читательского дневника мало чем отличается от создания записи на страницу клуба. Проверяется заполнение поля «title» и отсутствие такого названия в коллекции «book_note». Считываются данные из полей для ввода информации о книге и заносятся в документ.

После заполнения список читательских дневников отображается на странице, где можно выбрать ранее заполненный дневник и перейти в него.

Вывод данных читательского дневника изображен на рисунке 35. Поиск дневника происходит по его идентификатору. Дальше считываются данные из документа и добавляются в строки. В уже заполненном дневнике остается возможность заполнения и редактирования полей.

```
private void loadData() {
    db.collection( collectionPath: "book_note") CollectionReference
        .document(noteId) DocumentReference
        .get() Task<DocumentSnapshot>
        .addOnSuccessListener(documentSnapshot -> {
            if (documentSnapshot.exists()) {
                String title = documentSnapshot.getString( field: "title");
                titleTextView.setText(title);
                String author = documentSnapshot.getString( field: "author");
                authorTextView.setText(author);

                //...
                String imageUrl = documentSnapshot.getString( field: "imageUrl");
                if (imageUrl != null && !imageUrl.isEmpty()) {
                    Glide.with( activity: book_note_look_activity.this) RequestManager
                        .load(imageUrl) RequestBuilder<Drawable>
                        .into(imageAvatar);
                }
                boolean yesChecked = documentSnapshot.getBoolean( field: "yesRadioButton");
                boolean noChecked = documentSnapshot.getBoolean( field: "noRadioButton");
                yesRadioButton.setChecked(yesChecked);
                noRadioButton.setChecked(noChecked);
                Float rating = documentSnapshot.getDouble( field: "rating").floatValue();
                bookRatingBar.setRating(rating);
            }
        })
        .addOnFailureListener(e -> {
            Toast.makeText(getApplicationContext(), text: "Ошибка при загрузке данных", Toast.LENGTH_SHORT).show();
        });
}
```

Рисунок 35 – Вывод дневника

3.6 Профиль пользователя

В профиле пользователь может изменить свои данные (Рисунок 36, 37). При изменении пароля обязательно надо ввести старый пароль. Обращение здесь идет к сервису Firebase Authentication.

```
private void updateProfile(String new_username, String new_about, String new_pswd, String reauthent) {  
    auth = FirebaseAuth.getInstance();  
    user = auth.getCurrentUser();  
    if (user != null) {  
        String uid = user.getId();  
  
        if (!new_pswd.isEmpty()) {  
            AuthCredential credential = EmailAuthProvider.getCredential(user.getEmail(), reauthent);  
            user.reauthenticate(credential)  
                .addOnCompleteListener(task -> {  
                    if (task.isSuccessful()) {  
                        user.updatePassword(new_pswd).addOnCompleteListener(passwordUpdateTask -> {  
                            });  
                        }  
                    });  
                }  
        }  
    }  
}
```

Рисунок 36 – Изменение пароля

Здесь в Firestore сохраняются изменения в профиле. Используется метод обновления данных update. Остальные изменения данных (клуба, читательского дневника) в приложении происходят по такому же принципу.

```

DocumentReference userRef = db.collection( collectionPath: "users").document(uid);

Map<String, Object> updates = new HashMap<>();
if (!new_username.isEmpty()) {
    updates.put("username", new_username);
}
if (!new_about.isEmpty()) {
    updates.put("aboutMe", new_about);
}

userRef.update(updates)
    .addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            Log.d( tag: "ChangeActivity", msg: "Успешно сохранено");
        }
    })
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Log.e( tag: "ChangeActivity", msg: "Ошибка", e);
        }
    });
}

Toast.makeText( context: this, text: "Ваши изменения успешно сохранены", Toast.LENGTH_SHORT).show();
}

```

Рисунок 37 – Изменение имени пользователя и описания

Также в профиле пользователь может создать свою подборку книг, написать рецензию и цитату. Написание рецензии и цитаты почти не отличается от создания записи на стене клуба. В них необходимо сослаться на определенную книгу.

3.7 Добавление изображения

Для добавления изображения в базу данных используется сервис Firebase Storage. При нажатии на кнопку добавить изображение, открывается окно отображающее хранилище изображений на телефоне пользователя (Рисунок 38).


```

private void chooseImage() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "Select Picture"), PICK_IMAGE_REQUEST);
}

new *
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK) {
        if (requestCode == PICK_IMAGE_REQUEST && data != null && data.getData() != null) {
            filePath = data.getData();
        }
    }
}
}

```

Рисунок 38 – Открытие окна выбора

После выбора оно передается в Storage. Следом URL этого изображения записывается в тот документ откуда было совершено добавление фотографии (Рисунок 39). Во время выполнения этих методов выводится уведомление о загрузке изображения, которое не пропадает пока все изменения не будут сохранены.

```

private void uploadAvatar(Uri filePath, DocumentReference newDocRef) {
    if (filePath != null) {
        final ProgressDialog progressDialog = new ProgressDialog( context: this);
        progressDialog.setTitle("Uploading...");
        progressDialog.show();

        StorageReference ref = storageReference.child( pathString: "images/" + UUID.randomUUID().toString());
        ref.putFile(filePath) UploadTask
            new *
            .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                new *
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    progressDialog.dismiss();
                    new *
                    ref.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                        new *
                        @Override
                        public void onSuccess(Uri uri) {
                            String imageUrl = uri.toString();
                            newDocRef.update( field: "imageUrl", imageUrl)
                                new *
                                .addOnSuccessListener(new OnSuccessListener<Void>() {
                                    new *
                                    @Override
                                    public void onSuccess(Void aVoid) {
                                        imageUploaded = true;
                                        if (imageUploaded) {
                                            Toast.makeText( context: book_note_activity.this, text: "Uploaded", Toast.LENGTH_SHORT).show();
                                        }
                                    }
                                }
                            }
                    }
                }
            }
    }
}

```

Рисунок 39 – Сохранение в Storage и imageUrl

3.8 Ридер

На странице полки пользователь может загрузить предварительно скаченную книгу в формате pdf со своего устройства. Отображение ридера сделано с использованием библиотеки AndroidPdfViewer [9].

Пользователь в диалоговом окне (как при выборе изображения) выбирает книгу. После выбора сам пишет название книги и в метод на рисунке 40 передается путь к книге и имя, введённое пользователем. Здесь путь, имя и флаги доступа сохраняются в постоянное хранилище, чтобы они были доступны между запусками приложения.

```

private void addBookToList(Uri uri, String userInputName) {
    String path = uri.toString(); // Получить путь из URI
    SharedPreferences sharedPreferences = getSharedPreferences( name: "MyBookList", MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    // Добавляем имя и путь книги в SharedPreferences
    editor.putString("book_" + System.currentTimeMillis() + "_name", userInputName);
    editor.putString("book_" + System.currentTimeMillis() + "_path", path);
    editor.putInt("book_" + System.currentTimeMillis() + "_flags", Intent.FLAG_GRANT_READ_URI_PERMISSION);
    editor.apply();
}

```

Рисунок 40 – Метод для сохранения пути, имени и флага доступа

После добавления библиотеки для отображения PDF стал доступен метод Pdfview (Рисунок 41), который позволяет загрузить и отобразить файлы. Следом записывается цепочка методов для настройки параметров отображения, в их число входят: разрешение перелистывания по свайпу, горизонтальное отображение страниц, масштабирование двойным касанием, быстрое перелистывание страниц и т.д.

```
private void displayFromUri(Uri bookPath) {
    try {
        InputStream inputStream = getApplicationContext().getContentResolver().openInputStream(bookPath);
        Pdfview.fromStream(inputStream)
            .enableSwipe(true)
            .swipeHorizontal(true)
            .enableDoubletap(true)
            .pageSnap(true)
            .pageFling(true)
            .defaultPage(0)
            .pageFling(true)
    }
}
```

Рисунок 41 – Метод для отображения PDF файла

3.8 Вывод по главе

В третьей главе было реализовано приложение, которое соответствует поставленным требованиям. В ходе работы были разработаны база данных, активность для каждой страницы, что способствует корректной работе приложения.

ЗАКЛЮЧЕНИЕ

В результате работы было создано мобильное приложение для онлайн книжного клуба.

В данной работе был проведен анализ предметной области, что в результате помогло сформировать требования к приложению. Они описывают основные преимущества каждого из рассмотренных продуктов, подчеркивая ключевые характеристики, представленные в описаниях. На основе требований были рассмотрены технологии необходимые для реализации продукта и спроектирован макет приложения.

Для реализации проекта были использованы программная среда Android Studio, язык программирования Java и база данных Firebase. Разработанное приложение соответствует всем поставленным требованиям.

В результате разработано мобильное приложение для любителей книг и общения, которое помогает облегчить чтение и улучшить его качество, посредством онлайн книжных клубов.

В дальнейшем планируется внедрение дополнительных функций в книжные клубы таких как голосовые сообщения, видеоконференции, опросы и голосование, возможность презентовать книгу, планер встреч. Также добавить уведомление о сообщениях в чате клуба и о запланированных встречах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мобильное приложение книжного рекомендательного сервиса «LiveLib». – URL: https://play.google.com/store/apps/details?id=ru.livelib.client&hl=en_US (дата обращения 20.10.2023).
2. Мобильное приложение «Букмейт». – URL: https://play.google.com/store/apps/details?id=ru.plus.bookmate&hl=en_US (дата обращения 20.10.2023).
3. Мобильное приложение «BookClubs: Book Club Organizer». – URL: <https://play.google.com/store/apps/details?id=com.bookclubz&hl=en> (дата обращения 20.10.2023).
4. ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления // Консорциум кодексов: электронный фонд правовых и нормативно-технических документов : официальный сайт. – URL: <https://docs.cntd.ru/document/1200026224> (дата обращения: 24.04.2024).
5. ГОСТ 7.9-95 (ИСО 214-76). Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования // Научная периодика: проблемы и решения. – URL: <https://nppir.ru/wp-content/uploads/22-gost-7.9-95.pdf> (дата обращения: 14.04.2024).
6. ГОСТ 7.1-2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления // Российская государственная библиотека: официальный сайт. – URL: https://diss.rsl.ru/datadocs/doc_291wu.pdf (дата обращения: 10.04.2024).
7. СТУ 7.5-07-2021. Стандарт университета. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности // Сибирский федеральный университет : официальный сайт. – URL: <https://about.sfu-kras.ru/node/8127> (дата обращения: 07.04.2024).

8. Начните работу с Cloud Firestore. – URL: <https://firebase.google.com/docs/firestore?hl=ru> (дата обращения 12.03.2024)

9. Библиотека AndroidPdfViewer. – URL: <https://github.com/DImuthuUpe/AndroidPdfViewer/> (дата обращения: 17.05.2024).

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
«17» / 06 / 2024 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Приложение для онлайн книжного клуба

Руководитель	<u>Макуха</u> подпись	<u>17.06.2024</u> дата	Л.В. Макуха должность, учёная степень
Выпускник	<u>Корова</u> подпись	<u>17.06.2024</u> дата	Д.А. Коровина
Нормоконтролёр	<u>Макуха</u> подпись	<u>17.06.2024</u> дата	Л.В. Макуха