

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий
Кафедра Программной инженерии
Специальность 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем»
Направление специальности 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
КУРСОВОГО ПРОЕКТА:**

по дисциплине «Программирование мобильных систем»
Тема Мобильное приложение для чтения манги

Исполнитель
Студент(ка) 3 курса группы 8 Сокол Н.С.
(Ф.И.О.)

Руководитель работы ассистент Уласевич Н. А.
(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой _____
Председатель _____
(подпись)

Содержание

ВВЕДЕНИЕ.....	6
1 Постановка задачи.....	7
1.1 Функциональные требования.....	7
1.2 Технические требования.....	8
1.3 Цели и ожидаемые результаты.....	9
2 Аналитический обзор прототипов.....	10
2.1 Обзор аналогичных решений.....	10
2.2 Требования к проектируемому программному средству.....	11
3 Анализ требований к программному средству.....	13
3.1 Технические средства разработки.....	13
3.1.1 Язык программирования Dart и фреймворк Flutter.....	13
3.1.2 Облачная платформа Firebase.....	14
3.1.3 Архитектура системы.....	14
4 Проектирование приложения.....	16
4.1 Проектирования логической структуры базы данных.....	16
4.2 Проектирования структуры экранов.....	17
4.3 Проектирование архитектуры приложения.....	18
5 Реализация программного средства.....	19
5.1 Настройка инфраструктуры и интеграция с облачными сервисами.....	19
5.2 Разработка серверной и клиентской логики.....	20
6 Анализ информационной безопасности приложения.....	20
6.1 Аутентификация пользователей.....	20
6.2 Двухфакторная аутентификация.....	20
6.3 Безопасность API.....	20
7 Тестирование приложения.....	21
7.1 Модульное тестирование.....	21
7.2 Тестирование пользовательского интерфейса.....	21
8 Руководство пользователя.....	23
ЗАКЛЮЧЕНИЕ.....	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	27
Приложение.....	28

Введение

Тема курсового проекта «Мобильное приложение для чтения манги» является особенно актуальной в условиях растущего интереса к цифровому контенту и популярности манги среди аудитории разных возрастов. В современном мире, где мобильные устройства становятся основным средством доступа к информации, удобные решения для чтения играют ключевую роль в удовлетворении потребностей читателей.

Анализ существующих приложений для чтения манги выявляет ряд проблем, таких как ограниченный выбор контента, неудобный интерфейс или недостаточная поддержка мультимедийных форматов. Эти факторы могут снижать уровень удовлетворенности пользователей и ограничивать их опыт чтения. Запросы аудитории требуют разработки универсальных и адаптивных решений, которые обеспечивают качественный доступ к манге, интуитивно понятный интерфейс и высокую степень защиты авторских прав.

Цель проекта – создание удобного и функционального мобильного приложения для чтения манги, способного удовлетворить потребности как индивидуальных пользователей, так и поклонников жанра. При разработке будет уделено особое внимание удобству навигации, быстродействию приложения и интеграции с популярными платформами, что позволит обеспечить пользователям комфортное чтение и доступ к разнообразному контенту.

Таким образом, реализация данного проекта не только способствует освоению современных технологий мобильной разработки, но и создает практический инструмент для любителей манги, позволяя им наслаждаться любимыми произведениями в любое время и в любом месте.

1. Постановка задачи

Разработка мобильного приложения для чтения манги в рамках курсовой работы направлена на создание функционального программного продукта, соответствующего современным требованиям пользователей и тенденциям цифрового контента. Основной целью проекта является проектирование, реализация и тестирование приложения, которое обеспечивает удобный и безопасный доступ к манге, а также предоставляет пользователям инструменты для управления библиотекой. В процессе разработки необходимо учитывать исходные требования, которые определяют функциональные и технические аспекты приложения, а также обеспечить его производительность, надежность и безопасность. Постановка задачи включает детальное описание функциональных возможностей, технических требований и этапов реализации, что позволяет четко структурировать процесс разработки и обеспечить достижение поставленных целей.

1.1. Функциональные требования

Одной из ключевых задач проекта является реализация функциональности, связанной с управлением ролями пользователей. В приложении предусмотрены две основные роли: администратор и читатель. Роль администратора предполагает наличие расширенных прав, таких как управление контентом, пользователями и библиотеками. Читатель, в свою очередь, имеет доступ к основным функциям приложения, включая чтение манги, создание закладок и управление своей библиотекой.

Управление библиотекой является важной частью функциональности приложения. Пользователи должны иметь возможность добавлять новые манги, удалять существующие и редактировать информацию о них, такую как название, автор или жанр. Эта функциональность должна быть реализована с учетом удобства использования, чтобы процесс поиска и управления библиотекой был интуитивно понятным и не требовал дополнительных усилий. Для этого предполагается использование базы данных, которая будет хранить информацию о манге и обеспечивать быстрый доступ к ней.

Функциональность чтения манги представляет собой центральный аспект приложения. Пользователи должны иметь возможность открывать мангу, переключаться между страницами и использовать функции масштабирования. Для реализации этой функциональности необходимо обеспечить поддержку различных форматов файлов и оптимизацию загрузки изображений, чтобы минимизировать время ожидания и нагрузку на сеть.

Обмен рекомендациями между пользователями также является важным аспектом. Пользователи должны иметь возможность делиться своими мнениями о манге и оставлять отзывы. Эта функциональность требует разработки механизма комментариев, а также системы оценки, что позволит улучшить взаимодействие между пользователями и повысить уровень вовлеченности.

Обеспечение безопасности является критически важным аспектом разработки приложения. Для защиты данных пользователей необходимо внедрить шифрование данных, которое будет применяться как при передаче информации, так и при ее хранении. Предполагается использование современных алгоритмов шифрования для обеспечения конфиденциальности. Кроме того, для защиты учетных записей пользователей потребуется реализовать двухфакторную аутентификацию, что добавляет дополнительный уровень безопасности при входе в приложение. Эти меры направлены на предотвращение несанкционированного доступа и защиту пользовательских данных от утечек.

1.2. Технические требования

В соответствии с требованиями курсовой работы, приложение должно использовать базу данных для хранения всей информации, включая данные пользователей, мангу, отзывы и закладки. Выбор базы данных для приложения "MangaReader" был обусловлен необходимостью обеспечения высокой производительности, масштабируемости и простоты интеграции с мобильным приложением. Для этого в проекте используется облачная платформа Firebase, предоставляющая NoSQL базу данных Cloud Firestore и мощные инструменты для работы с данными. Cloud Firestore позволяет гибко организовать хранение информации благодаря документо-ориентированной модели, что упрощает адаптацию структуры данных под изменяющиеся требования приложения. Поддержка функций реального времени обеспечивает мгновенную синхронизацию данных, что критически важно для реализации обновления статусов манги, отзывов и уведомлений. Удобные SDK для Flutter и встроенные механизмы аутентификации и безопасности, такие как Firebase Security Rules, значительно упростили интеграцию и защиту данных, позволяя сосредоточиться на разработке функциональности приложения.

Авторизация и аутентификация являются обязательной частью приложения. Для реализации авторизации предполагается использование ролевых моделей, которые будут определять доступ пользователей к различным функциям в зависимости от их статуса (администратор, модератор или пользователь). Аутентификация будет осуществляться через ввод логина и пароля с дополнительной проверкой через двухфакторную аутентификацию.

Тестирование производительности приложения является важным этапом разработки, так как MangaReader должен стабильно работать при различной нагрузке. Для этого необходимо провести нагрузочное тестирование, чтобы оценить, как приложение справляется с большим количеством одновременно активных пользователей, отправкой отзывов и загрузкой манги. Также требуется протестировать скорость отклика интерфейса и стабильность соединения при использовании функций реального времени. Результаты тестирования позволят выявить узкие места в производительности и оптимизировать код приложения, чтобы обеспечить комфортное использование даже на устройствах с ограниченными ресурсами.

1.3.Цели и ожидаемые результаты

Основной целью разработки является создание мобильного приложения для чтения манги, которое полностью соответствует заявленным требованиям и предоставляет пользователям удобный и безопасный способ доступа к манге. Приложение должно обеспечивать стабильную работу всех функций, включая управление библиотекой манги, отзывами и закладками, а также гарантировать защиту данных за счет шифрования и двухфакторной аутентификации. Ожидается, что использование облачной платформы Firebase позволит минимизировать затраты на разработку инфраструктуры и обеспечит масштабируемость системы. Это обеспечит гибкость в адаптации приложения под изменяющиеся требования пользователей и позволит быстро внедрять новые функции. В результате выполнения курсовой работы будет представлено полноценное приложение, готовое к тестированию и дальнейшему развитию, а также подробная документация, описывающая процесс разработки и полученные результаты. Постановка задачи, описанная выше, служит основой для структурированного подхода к реализации проекта, обеспечивая четкое понимание целей и последовательности действий на всех этапах работы. Это позволит эффективно управлять ресурсами и временем, а также гарантирует высокое качество конечного продукта.

2. Анализ аналогичных решений

2.1. Обзор аналогичных решений

В данном разделе рассмотрены три наиболее популярных приложения для чтения манги, отличающихся по своим функциональным особенностям и подходам к обеспечению удобства пользователей и доступа к контенту.

Manga Rock является одним из самых известных приложений для чтения манги. Его основными преимуществами являются интуитивно понятный интерфейс и обширная библиотека манги, что делает чтение удобным и доступным. Приложение поддерживает чтение манги в различных форматах, добавление закладок и комментариев, персонализированные рекомендации. Однако у приложения есть ограничения, такие как отсутствие функции загрузки манги для оффлайн-чтения и ограниченные настройки для интеграции с внешними сервисами. Интерфейс приложения представлен на рисунке 1.1:

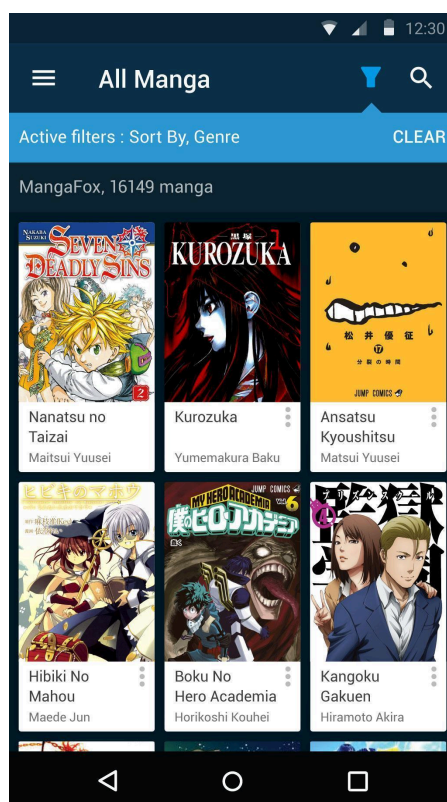


Рисунок 1.1 – Интерфейс приложения Manga Rock

Crunchyroll Manga выделяется своей интеграцией с платным сервисом потокового видео, предоставляя пользователям доступ к манге сразу после её выхода в Японии. Приложение поддерживает чтение манги в высоком качестве, возможность комментирования и обсуждения глав, персонализированные списки избранного. Crunchyroll Manga предлагает пользователям возможность чтения свежих глав, что делает его привлекательным для фанатов, следящих за новыми выпусками. Однако приложение требует подписки для полного доступа к контенту,

что может быть недостатком для некоторых пользователей. Интерфейс приложения представлен на рисунке 1.2:

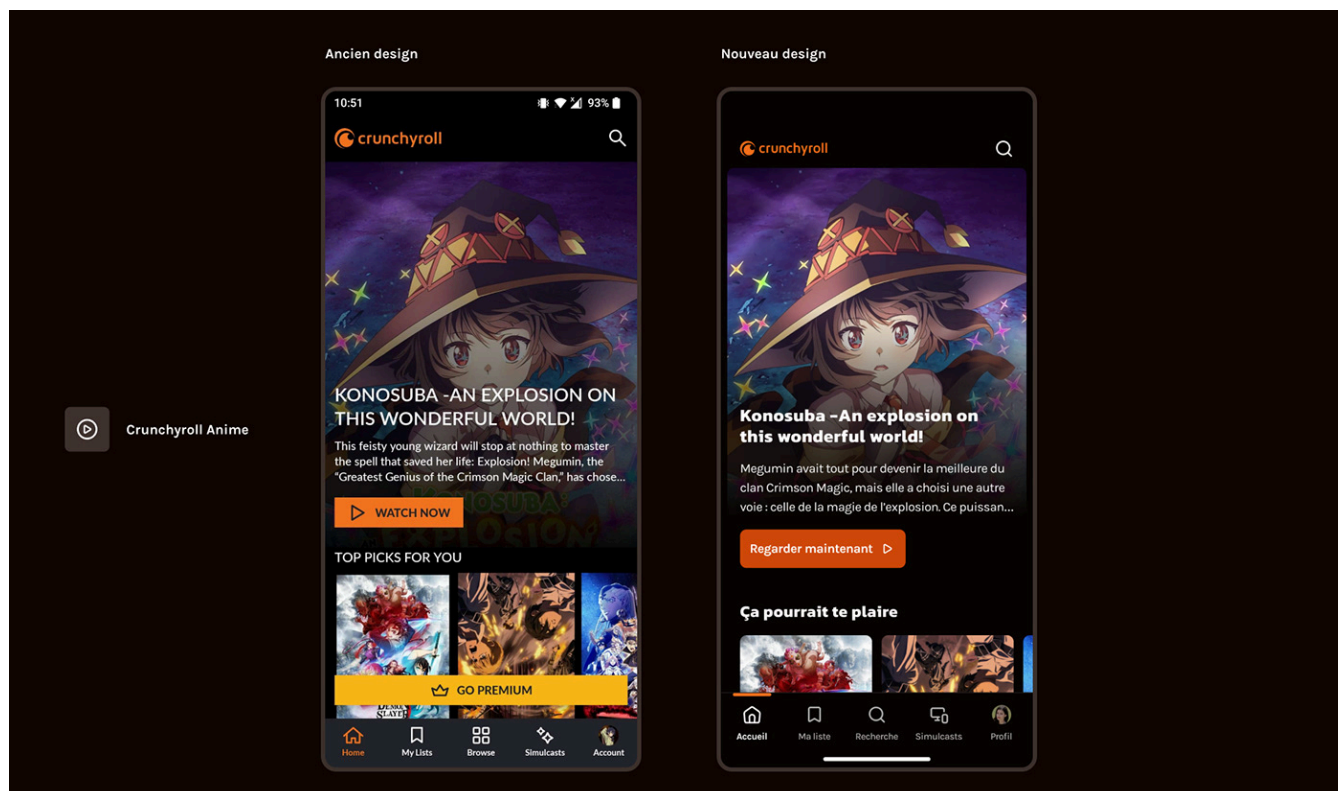


Рисунок 1.2 – Интерфейс приложения Crunchyroll Manga

VIZ Manga позиционируется как платформа, ориентированная на качественный контент и доступ к популярным сериям. Приложение с акцентом на лицензированную мангу завоевало доверие пользователей благодаря строгим стандартам качества. VIZ Manga поддерживает чтение манги с официальным переводом, возможность покупки отдельных томов и глав, Функцию загрузки для оффлайн-чтения. Интерфейс приложения остаётся простым и удобным, хотя функциональные возможности могут уступать более гибким конкурентам. Основное внимание разработчиков VIZ Manga уделено качеству и легальному доступу к контенту.

Таким образом, анализируя данные решения, можно отметить, что выбор приложения для чтения манги зависит от приоритетов пользователя: Manga Rock: удобство и обширная библиотека, Crunchyroll Manga: свежие главы и интеграция с видео, а VIZ Manga: качественный контент и легальный доступ.

2.2. Требования к проектируемому программному средству

Разработка проектируемого программного средства "MangaReader" проводится с учётом выявленных преимуществ и недостатков аналогичных решений, таких как Manga Rock, Crunchyroll Manga и VIZ Manga. Это позволяет сформировать требования, направленные на устранение имеющихся ограничений и

интеграцию лучших практик, что обеспечит конкурентоспособность и высокое качество конечного продукта.

Проектируемое приложение "MangaReader" должно обеспечивать базовый набор функций, включая управление мангой, отзывами, закладками и избранными. Пользователи смогут добавлять, удалять и изменять информацию о манге, а также оставлять и редактировать свои отзывы. Кроме того, приложение будет поддерживать возможность добавления манги в закладки и избранные списки для быстрого доступа.

Важной частью функционала станет регистрация пользователей с двухфакторной аутентификацией, что обеспечит высокий уровень безопасности. Дополнительно планируется внедрение расширенных возможностей, таких как интеграция ботов для автоматизации задач и поддержка уведомлений в режиме реального времени, чтобы мгновенно оповещать пользователей о новых выпусках и обновлениях манги. Особое внимание будет уделено адаптивности функционала, что позволит пользователям настраивать интерфейс и рабочие процессы под свои нужды. Это также обеспечит интеграцию приложения с внешними сервисами для повышения общей эффективности взаимодействия с контентом. Ключевыми аспектами приложения являются удобство использования, безопасность и высокая производительность. Интерфейс должен быть интуитивно понятным и обеспечивать стабильную работу всех функций, а данные пользователей должны быть надежно защищены.

Проектируемая система должна основываться на модульной архитектуре, что обеспечит гибкость при внедрении новых функций и масштабировании системы. Использование проверенных шаблонов проектирования и современных технологий позволит создать устойчивое решение, способное оперативно реагировать на изменения рыночных требований. Важно обеспечить возможность интеграции с внешними сервисами и API, что даст возможность расширять функционал и взаимодействовать с другими приложениями. Это позволит удовлетворить как повседневные, так и специализированные потребности пользователей.

Выработка требований к программному средству основывается на детальном анализе существующих аналогов, что позволяет сформировать универсальное решение с широким функционалом, высоким уровнем безопасности и адаптивностью интерфейса, способное удовлетворить запросы пользователей и обеспечить быстрое развитие в изменяющихся условиях рынка.

3. Анализ требований к программному средству

3.1. Технические средства разработки

В разработке приложения "MangaReader" используются современные технические средства, обеспечивающие высокую производительность и удобство эксплуатации. Приложение реализуется на языке программирования Dart с использованием фреймворка Flutter, что обеспечивает кроссплатформенность и возможность создания гибких, отзывчивых интерфейсов. Преимущества Flutter включают быстрое прототипирование, богатый набор виджетов и активное сообщество разработчиков, что способствует сокращению сроков разработки.. В качестве системы хранения данных и облачной платформы выбрана Firebase, предоставляющая эффективные инструменты для управления базами данных, аутентификации пользователей и обеспечения безопасности данных. Такая комбинация технологий позволяет создать надежное и масштабируемое приложение, отвечающее современным требованиям пользователей.

3.1.1 Язык программирования Dart и фреймворк Flutter

Dart — это современный объектно-ориентированный язык программирования, разработанный компанией Google. Он создан для обеспечения высокой производительности приложений на различных платформах, включая мобильные, веб и серверные решения. Dart отличается лаконичным синтаксисом и эффективными инструментами для разработки, что позволяет создавать масштабируемые и поддерживаемые проекты с минимальными накладными расходами.

Flutter — это мощный фреймворк от Google, построенный на основе языка Dart, предназначенный для разработки нативных интерфейсов для мобильных, веб- и настольных приложений. Одним из ключевых преимуществ Flutter является его кроссплатформенность: один и тот же код может быть использован для создания приложений на различных операционных системах, что значительно сокращает время и затраты на разработку.

Вместе Dart и Flutter формируют современную среду для создания инновационных мобильных приложений, позволяя разработчикам сосредоточиться на реализации уникального функционала, при этом обеспечивая высокую производительность и удобство использования конечными пользователями.

3.1.2 Облачная платформа Firebase

Firebase — это облачная платформа, разработанная компанией Google, которая предоставляет разработчикам все необходимые инструменты для создания современных мобильных и веб-приложений. Платформа охватывает широкий спектр сервисов, включая облачные базы данных (Firestore), системы

аутентификации, хранилище файлов, а также инструменты для аналитики и мониторинга.

Одним из главных преимуществ Firebase является возможность реализации серверной логики без необходимости в собственном серверном оборудовании, что существенно ускоряет процесс разработки и упрощает масштабирование приложения. Интеграция с другими сервисами Google позволяет обеспечить высокую производительность, надежность и безопасность проекта.

3.1.3 Архитектура системы

Для курсового проекта по разработке приложения "MangaReader" использован модульный подход, основанный на клиент-серверной архитектуре. Приложение состоит из двух основных компонентов: клиентская часть и серверная часть с базой данных.

Клиентская часть обеспечивает взаимодействие пользователя с приложением через удобный графический интерфейс, позволяя пользователям комфортно читать мангу, оставлять отзывы и управлять закладками. Серверная часть отвечает за обработку запросов от клиентских устройств, управление сессиями пользователей и организацию обмена данными в реальном времени, обеспечивая стабильную работу и безопасность данных.

Различают три уровня требований к проекту: бизнес-требования, пользовательские требования и функциональные требования. Эти требования отражают стратегические цели разработчиков приложения и определяются на высоком уровне.

Основные бизнес-требования включают обеспечение простоты и удобства интерфейса, а также использование надежной системы управления базами данных и серверной инфраструктуры.

Пользовательские требования описывают задачи и возможности, которые система должна предоставить конечным пользователям. Пользователь приложения должен иметь возможность регистрироваться и проходить аутентификацию, просматривать и читать мангу, оставлять отзывы и добавлять мангу в закладки. Администраторы, помимо этого, могут добавлять новые манги, редактировать информацию о них и удалять неподходящий контент.

Функциональные требования определяют конкретные функции системы, такие как централизованное хранение всей информации о манге в базе данных, возможность создания нового аккаунта через регистрационную форму, аутентификация пользователей по введенным учетным данным и обеспечение обмена данными о манге и отзывах в режиме реального времени.

Таким образом, тщательный анализ требований к приложению "MangaReader" позволил сформировать четкий перечень бизнес-, пользовательских и функциональных требований, что закладывает прочную основу для дальнейшей разработки системы в соответствии с современными стандартами и ожиданиями конечных пользователей.

4. Проектирование приложения

На этапе проектирования приложения "MangaReader" была разработана общая структура, включающая логическую модель базы данных, пользовательский интерфейс и программную архитектуру. Основной целью являлось создание масштабируемого, удобного для пользователя и поддерживаемого мобильного приложения для чтения манги. В процессе проектирования была создана логическая модель базы данных, которая включает все ключевые сущности, такие как манга, пользователи, отзывы и закладки. Это обеспечивает централизованное хранение данных и упрощает управление ими.

4.1. Проектирования логической структуры базы данных

Для хранения данных приложения "MangaReader" была выбрана облачная NoSQL база данных Cloud Firestore от Google Firebase. Этот выбор обусловлен её гибкостью, позволяющей легко адаптировать структуру данных по мере развития приложения, а также автоматической масштабируемостью для обработки больших объемов данных и трафика. Возможности работы в реальном времени, такие как мгновенная доставка сообщений и обновление статусов, идеально подходят для приложения, обеспечивая пользователям актуальную информацию и мгновенные обновления контента. Интеграция с Flutter и Firebase упрощает разработку благодаря удобным SDK, что позволяет быстро реализовать необходимые функции и сосредоточиться на создании качественного пользовательского опыта. Таким образом, выбор Cloud Firestore обеспечивает надежное и эффективное решение для хранения и обработки данных в приложении "MangaReader", соответствующее современным требованиям пользователей.

База данных проекта состоит из 4 коллекций, схема которой представлена на рисунке 4.1

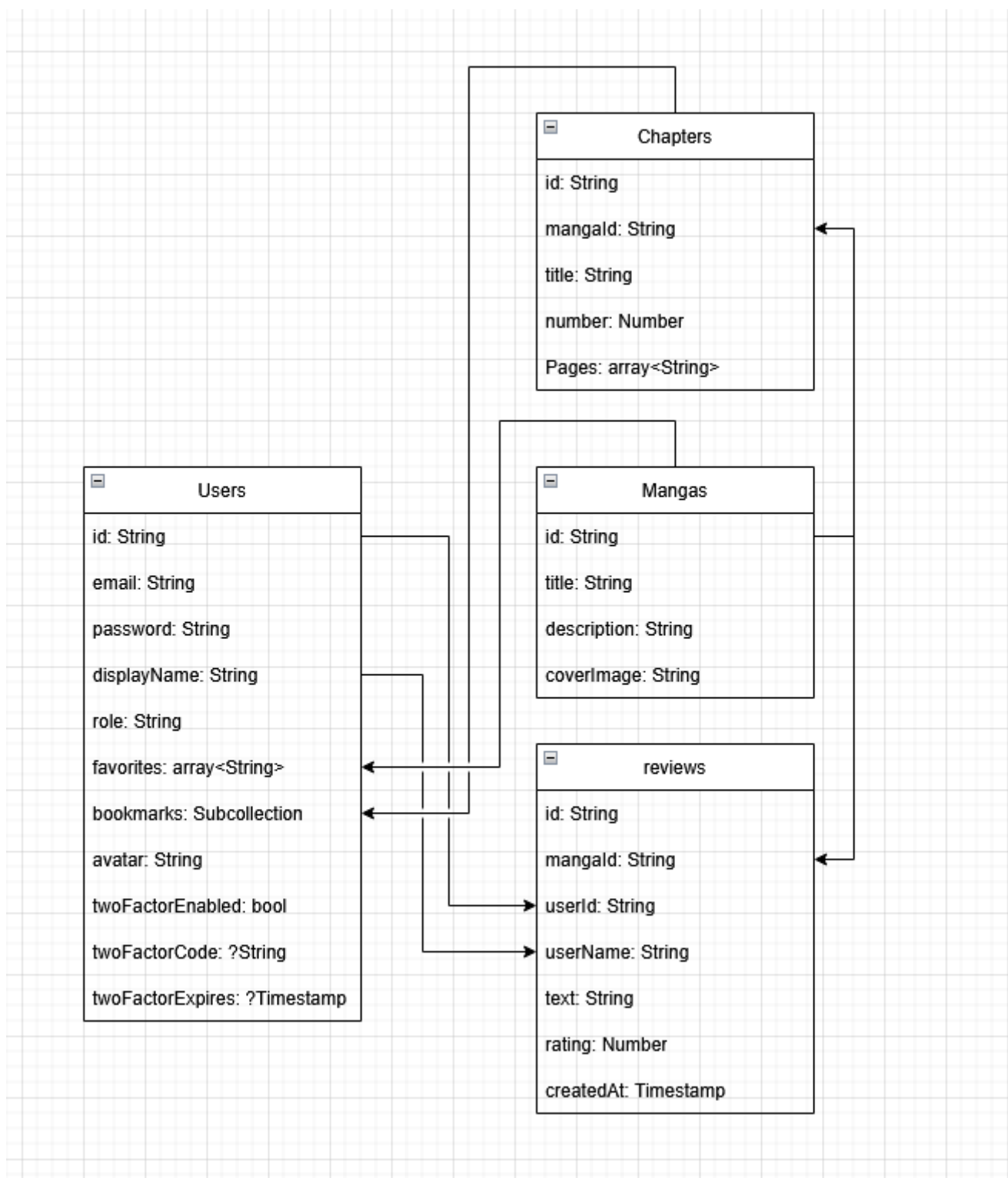


Рисунок 4.1 – Структура базы данных

Коллекция `users` предназначена для хранения информации о зарегистрированных пользователях. В ней содержатся данные, такие как уникальный идентификатор пользователя, электронная почта, имя пользователя, роль, закладки, избранные манги, дата регистрации и дата последней активности.

Эта коллекция позволяет управлять профилями пользователей и их состоянием в приложении.

Коллекция `manga` служит для управления всеми мангами в системе. Каждый документ представляет собой уникальную мангу и содержит уникальный идентификатор, название, описание, обложку в формате Base64, массив глав и дату создания. Эта коллекция обеспечивает удобное отображение и взаимодействие с мангами.

Коллекция `chapters` организована как подколлекция внутри каждой манги и используется для хранения глав. Каждая глава включает уникальный идентификатор, название, список страниц и номер главы. Это позволяет эффективно управлять содержимым манги.

Коллекция `reviews` отвечает за хранение комментариев пользователей к манге. Каждая запись содержит уникальный идентификатор, идентификатор манги, идентификатор пользователя, текст комментария и дату создания. Эта коллекция позволяет пользователям взаимодействовать с мангой, оставляя свои мнения и отзывы.

4.2. Проектирования структуры экранов

Приложение включает в себя 6 экранов. Структура экранов приложения была разработана для обеспечения интуитивного и эффективного взаимодействия с пользователем. Использование фреймворка Flutter и принципов Material Design позволило создать современный и удобный интерфейс.

В приложении "MangaReader" предусмотрены основные экраны, каждый из которых выполняет свою уникальную функцию. Экран аутентификации служит для входа и регистрации пользователей, предоставляя поля для ввода электронной почты и пароля, а также опцию восстановления пароля для удобства пользователей. Экран списка манги отображает все доступные манги, позволяя пользователям создавать новые закладки, осуществлять поиск и получать доступ к настройкам профиля. Экран манги предназначен для чтения, включая область отображения страниц, навигацию между главами и возможность оставлять комментарии к каждой манге. Экран комментариев предназначен для просмотра и добавления комментариев к манге, предлагая поле ввода для написания новых комментариев и отображение существующих. Экран поиска позволяет пользователям находить мангу по названию или жанру, с возможностью фильтрации результатов для более удобного поиска.

Дополнительные элементы интерфейса, такие как модальные окна, индикаторы загрузки и уведомления, используются для улучшения взаимодействия. Навигация между экранами реализована через механизм Navigator в Flutter, обеспечивая логичные переходы.

4.3.Проектирование архитектуры приложения

Архитектура приложения разработана с учетом гибкости, масштабируемости и тестируемости, для чего была выбрана многослойная структура с паттерном BLoC (Business Logic Component). Она включает слой представления, бизнес-логики, данных, сервисов и моделей. Диаграмма на рисунке 4.1 иллюстрирует структуру приложения в рамках этого шаблона.

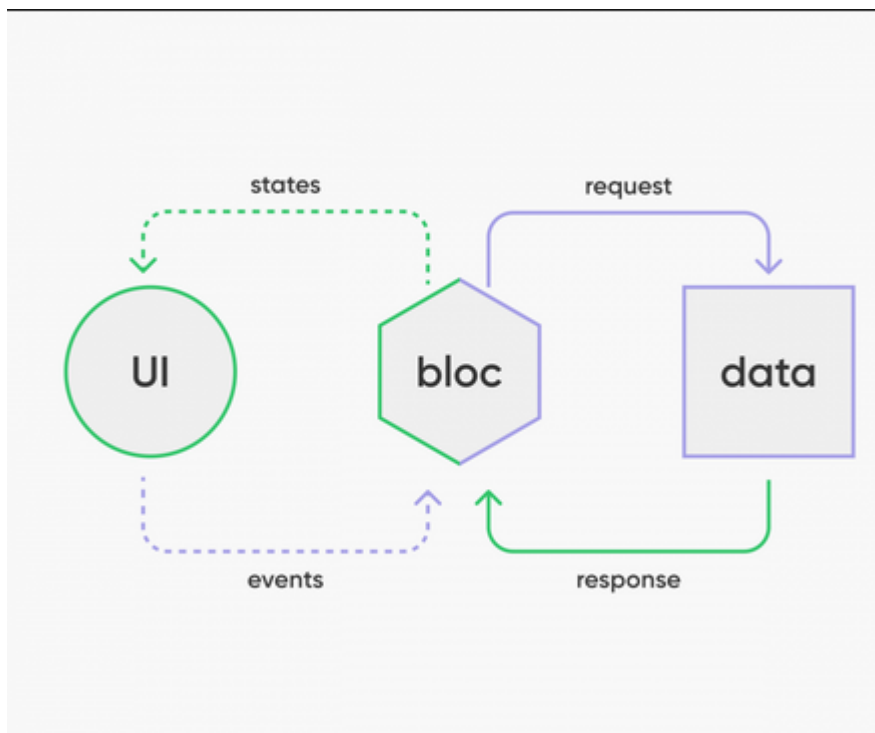


Рисунок 4.2 – Структура архитектурного паттерна BLoC

Слой представления, построенный на Flutter виджетах, отвечает за интерфейс и обработку ввода, взаимодействуя с бизнес-логикой через события и состояния. Слой бизнес-логики, реализованный с помощью flutter_bloc, управляет состоянием экранов, обрабатывает события и координирует работу с данными. Слой данных, основанный на паттерне Репозиторий, абстрагирует доступ к Firestore, скрывая детали реализации.

Слой моделей описывает сущности данных, такие как пользователи и сообщения, используемые всеми слоями. Использовались технологии Flutter, Dart, Cloud Firestore, а также библиотеки flutter_bloc и image_picker. Поток данных начинается с интерфейса, который передает события в BLoC, взаимодействует с репозиториями, обновляет состояние и перестраивает интерфейс. Такая архитектура обеспечивает четкое разделение ответственностей и надежную основу для развития приложения "MangaReader".

5. Реализация приложения

5.1. Настройка инфраструктуры и интеграция с облачными сервисами

Настройка инфраструктуры и интеграция с облачными сервисами для разработки приложения "MangaReader" включала использование Firebase для аутентификации пользователей и управления базой данных Cloud Firestore. Процесс настройки подключения к Firebase включал создание проекта в консоли Firebase, добавление приложения в проект и загрузку конфигурационного файла `firebase_options.dart`. Firebase обеспечивал надежное хранение данных и управление пользователями, что критично для функциональности приложения. Настройка включала также интеграцию с Firebase Authentication для упрощения процессов регистрации и входа пользователей. Слой сервисов инкапсулировал взаимодействие с внешними сервисами, включая использование Firebase Storage для хранения файлов, таких как аватарки пользователей и изображения в чатах. Настройка Firebase Storage включала создание правил доступа, чтобы ограничить операции загрузки и скачивания файлов, обеспечивая их безопасность. Такой подход к настройке инфраструктуры и интеграции облачных сервисов гарантирует надежность, безопасность и масштабируемость приложения "MangaReader".

5.2. Разработка серверной и клиентской логики

Разработка серверной и клиентской логики приложения "MangaReader" была организована для обеспечения полной функциональности мессенджера. Структура Flutter-проекта была разделена на модели данных (`models`), сервисы (`services`) для взаимодействия с облачными платформами и реализации бизнес-логики, а также экраны (`screens`) и виджеты (`widjets`) для пользовательского интерфейса.

Серверная логика обеспечивала функции аутентификации и уведомлений, взаимодействуя с облачными сервисами, такими как Firebase Authentication. Это позволяло пользователям безопасно входить в приложение и получать уведомления.

Проектирование и реализация пользовательского интерфейса осуществлялись с использованием Flutter, что позволило создать современный и отзывчивый дизайн. Были разработаны различные экраны, включая аутентификацию, список манги, чтение манги, настройки профиля и избранное, а также переиспользуемые компоненты, такие как кнопки, текстовые поля и карточки манги. Эти элементы, размещенные в директориях `lib/screens` и `lib/widjets`, обеспечивали модульность и повторное использование кода.

6. Анализ информационной безопасности приложения

Анализ информационной безопасности приложения "MangaReader" направлен на оценку мер, реализованных для защиты пользовательских данных, обеспечения конфиденциальности и предотвращения несанкционированного доступа. В процессе разработки были внедрены современные механизмы аутентификации и авторизации, включая использование функций шифрования пароля, что обеспечивает дополнительную защиту пользователей.

6.1. Аутентификация пользователей

Для аутентификации пользователей в приложении используется Firestore. Эта платформа обеспечивает безопасный вход и регистрацию, включая встроенные механизмы защиты от атак типа брутфорс через правила доступа к бд. Firestore поддерживает надежное хранение данных и их шифрование, что минимизирует риски компрометации информации пользователей.

6.2. Двухфакторная аутентификация

Двухфакторная аутентификация (2FA) реализована для дополнительной защиты учетных записей. Она требует ввода одноразового кода, отправляемого на электронную почту пользователя, что значительно повышает безопасность, так как доступ к аккаунту возможен только при наличии пароля и доступа к почтовому ящику.

7. Тестирование приложения.

Тестирование приложения "MangaReader" проводилось с целью проверки его функциональности, производительности, безопасности и удобства использования, чтобы гарантировать стабильную работу и соответствие заявленным требованиям. Процесс тестирования включал несколько этапов, охватывающих различные аспекты приложения.

7.1. Модульное тестирование

На первом этапе тестирования приложения **"MangaReader"** проводилось модульное тестирование, направленное на проверку отдельных компонентов, таких как функции аутентификации, загрузки манги, управления избранным и добавления комментариев. Для этого использовались библиотеки тестирования, такие как `flutter_test`, для написания юнит-тестов, которые проверяли корректность работы моделей данных, сервисов и логики BLoC. Модульное тестирование обеспечивало уверенность в правильности обработки данных и выполнении операций на уровне кода, минимизируя вероятность ошибок при интеграции. Этот этап был критически важен для выявления проблем на ранних стадиях разработки, что способствовало повышению качества конечного продукта. Результаты модульного тестирования можно наблюдать на рисунке 7.1.

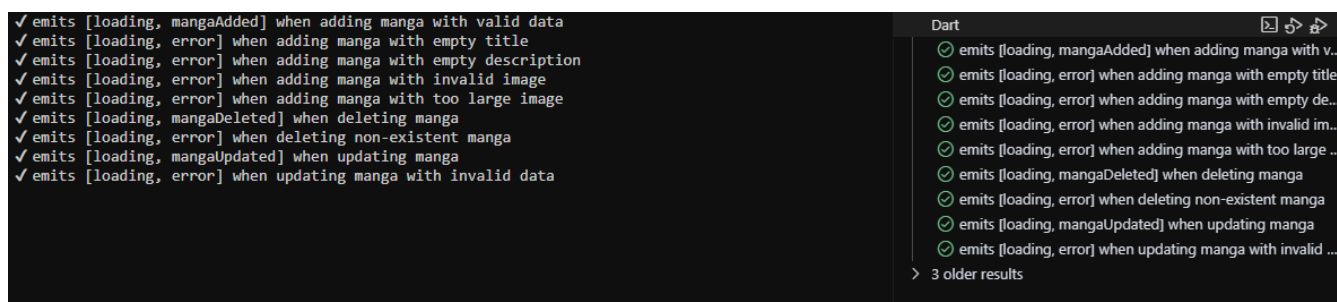


Рисунок 7.1 – Результаты модульного тестирования

Результаты модульного тестирования, представленные на рисунке 7.1, демонстрируют высокий уровень покрытия кода тестами и успешное выполнение всех проверок, что подтверждает надежность отдельных модулей приложения.

7.2. Тестирование пользовательского интерфейса

Тестирование пользовательского интерфейса приложения "MangaReader" проводилось преимущественно с использованием ручных методов, что позволило детально оценить удобство и корректность работы интерфейса. Ручное тестирование охватывало проверку навигации между экранами, включая переходы от списка манги к чтению, настройкам профиля и избранному, чтобы убедиться в

плавности и логичности взаимодействия. Также тестировалась корректность отображения всех экранов, таких как аутентификация, список манги и комментарии, на устройствах с различными размерами и разрешениями экранов, включая смартфоны с маленькими дисплеями. Особое внимание уделялось взаимодействию с элементами интерфейса, такими как кнопки для добавления в избранное, поля ввода комментариев и элементы управления навигацией, чтобы гарантировать их доступность и отзывчивость.

8. Руководство пользователя

MangaReader — это современное приложение для чтения манги, разработанное для удобного и безопасного использования пользователями. Оно предлагает широкий набор функций, включая просмотр манги, добавление в избранное, создание списков чтения и возможность оставлять комментарии..

Для установки MangaReader необходимо устройство на базе Android 5.0 или выше либо iOS 11 или выше и доступом к интернету. Процесс установки прост и не требует специальных навыков, а приложение автоматически адаптируется к настройкам вашего устройства.

Регистрация нового аккаунта начинается с открытия приложения и выбора опции "Зарегистрироваться" на экране входа, после чего вы будете перенаправлены на экран регистрации, представленный на рисунке 8.1

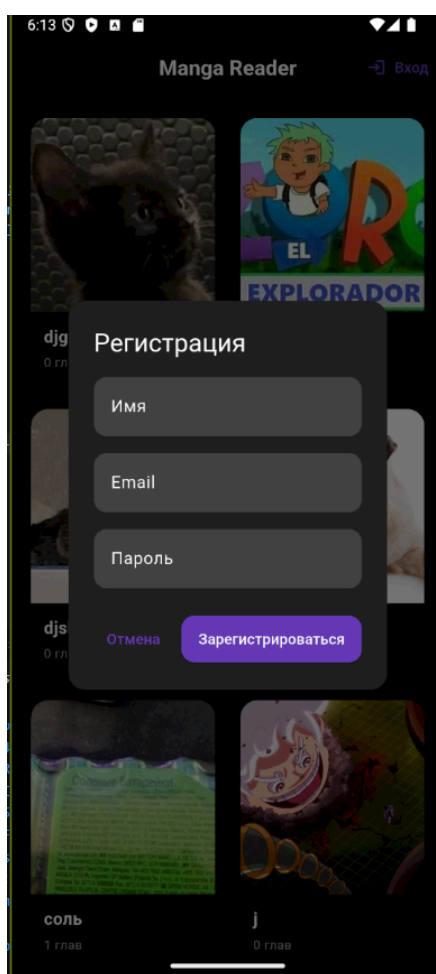


Рисунок 8.1 – Экран регистрации

Заполните обязательные поля при регистрации, включая имя пользователя, адрес электронной почты и пароль (не менее 6 символов). Для входа в существующий аккаунт достаточно ввести email и пароль, а при включенной двухфакторной аутентификации — дополнительно код из письма. Если вы забыли

пароль, на экране входа выберите "Забыли пароль?", укажите email и следуйте инструкциям из полученного письма для установки нового пароля.

Чтение манги в MangaReader поддерживает различные форматы контента. Для начала чтения нажмите на мангу на главной странице приложения. Главная страница представлена на рисунке 8.2

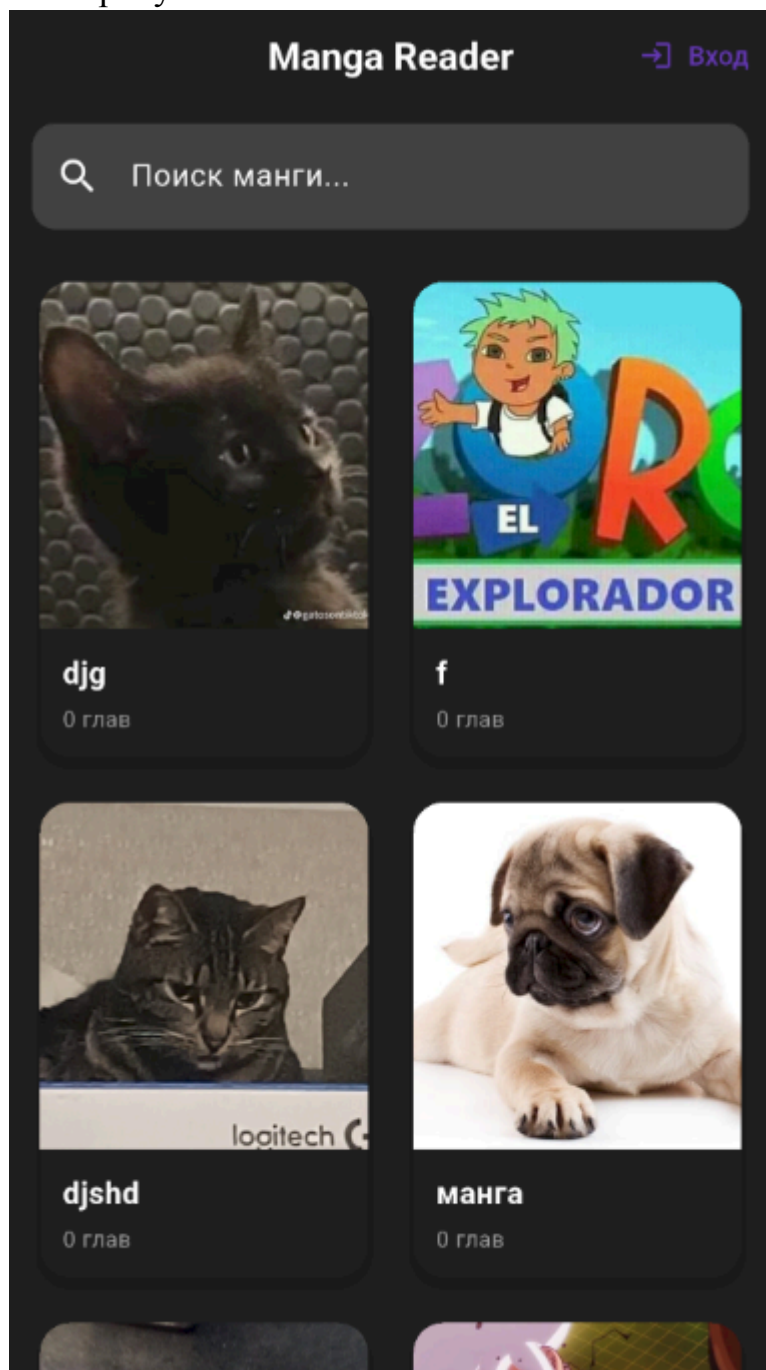


Рисунок 8.2 – Главная страница

После выберите главы и наслаждайтесь чтением. Вы можете добавить закладки, чтобы быстро возвращаться к нужным главам.

Управление избранным осуществляется в разделе "Избранное". Для добавления новой манги в избранное нажмите кнопку в правом верхнем углу на

странице манги, после чего она появится в списке. Чтобы отредактировать или удалить мангу из избранного, нажмите на ее название, после чего вы попадете на страницу с описанием, откуда можно удалить мангу из списка или добавить комментарий. Раздел избранного представлен на рисунке 8.3

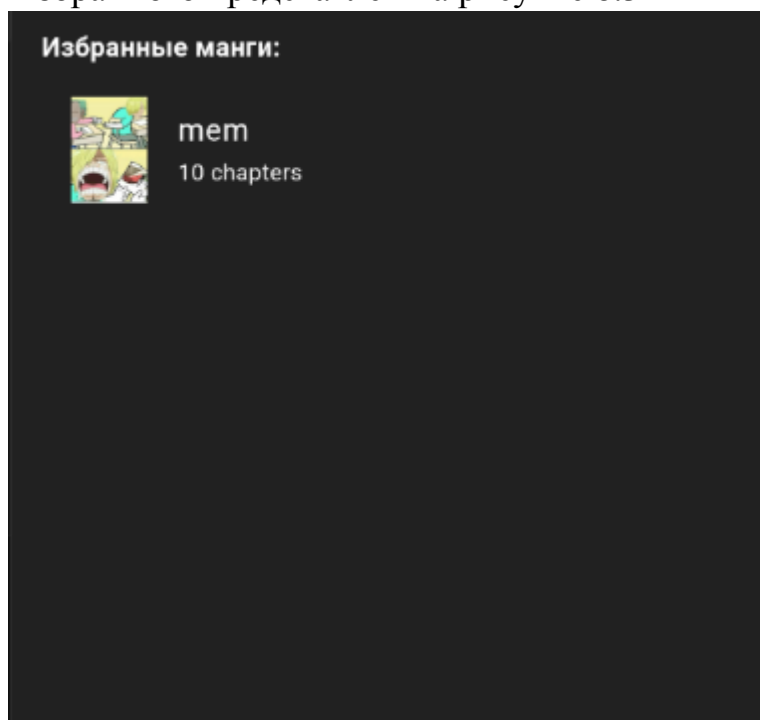


Рисунок 8.3 – Экран избранного

Профиль пользователя доступен в разделе "Профиль", где отображаются имя, фото и избранное. Изменение фото профиля выполняется через выбор изображения из галереи или съемку нового фото с последующей обрезкой и сохранением. Эти функции позволяют персонализировать приложение и повысить его безопасность.

Двухфакторная аутентификация (2FA) повышает безопасность аккаунта. Для её включения перейдите в настройки профиля, Включите тумблер "Двойная аутентификация" и подтвердив кодом из email. При входе с включенной 2FA после ввода email и пароля потребуется ввести код подтверждения, отправленный на почту, что защищает аккаунт от несанкционированного доступа.

Заключение

Разработка мобильного приложения "MangaReader" в рамках курсовой работы успешно достигла поставленных целей, обеспечив создание функционального, безопасного и удобного инструмента для чтения манги. Приложение реализовано с использованием современных технологий, таких как Flutter для кроссплатформенной разработки и встроенные средства для аутентификации и хранения данных.

В процессе проектирования была тщательно проработана логическая структура базы данных, архитектура приложения и пользовательский интерфейс, что позволило создать масштабируемую и поддерживаемую систему. Ключевые функциональные возможности, включая управление библиотекой манги, добавление в избранное, комментарии и настройки профиля, были реализованы в полном соответствии с требованиями.

Тестирование приложения, включающее модульное тестирование и тестирование пользовательского интерфейса, подтвердило его стабильность, производительность и удобство использования. Выявленные в процессе тестирования недочеты были устранены, что обеспечило высокое качество конечного продукта. Руководство пользователя предоставляет исчерпывающую информацию для комфортного освоения приложения, делая его доступным даже для начинающих пользователей.

В результате "MangaReader" представляет собой полноценное приложение, готовое к использованию и дальнейшему развитию. Проект демонстрирует успешное применение современных подходов к разработке мобильных приложений, включая многослойную архитектуру и облачные технологии. Дальнейшее развитие может включать добавление новых функций, таких как улучшенные настройки чтения или расширенные возможности для совместного использования, а также оптимизацию производительности для поддержки большего числа пользователей. Данная работа подтверждает возможность создания надежных и функциональных приложений для чтения манги с учетом строгих требований к безопасности и удобству.

Список использованных источников

1. Официальная документация Flutter [Электронный ресурс] – <https://flutter.dev/docs> – Дата доступа 02.05.2025
2. Официальная документация Firebase Authentication [Электронный ресурс] – <https://firebase.google.com/docs/auth> – Дата доступа 02.05.2025
3. Официальная документация Cloud Firestore [Электронный ресурс] – <https://firebase.google.com/docs/firestore> – Дата доступа 02.05.2025