

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет **Прикладной информатики**

Направление подготовки **09.03.03 Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

КУРСОВОЙ ПРОЕКТ

Тема: «Мобильное приложение для отслеживания поставленной цели»

Обучающийся: Гашимов Ильхам Фаррух оглы, К3139

Санкт-Петербург 2024

ВВЕДЕНИЕ

Актуальность рассматриваемой темы

Ведение отчетов на листе бумаги не слишком удобно и давно потеряла свою актуальность. В наше время без отчетов не обойтись, особенно крупным предприятиям. Одним из наиболее эффективных способов контроля за выполнением задач и личных достижений является практика о ведении ежедневных отчетов о выполненных задачах, делах и достижениях в течение дня и ее субъективное оценивание. Такие отчеты помогают пользователям приложения контролировать свою деятельность и отслеживать как свой прогресс, так и других пользователей.

Кому нужен полученный результат?

1. Людям, работающим над саморазвитием

Приложение будет полезно тем, кто хочет следить за своим прогрессом в процессе личного роста, например, при развитии новых навыков, формировании полезных привычек или достижении личных целей. Регулярные отчеты о сделанных делах помогают поддерживать мотивацию и уверенность в своих силах. Кроме того, в процессе отслеживания целей других пользователей у нас может возникнуть желание поставить перед собой подобные цели или принять участие в похожих начинаниях. Видя, как другие достигают своих целей, можно вдохновиться и попробовать применить их подходы к своему собственному развитию. Это создает эффект взаимной мотивации, когда успехи других людей становятся стимулом для личных достижений и стремления к новым вершинам.

2. Командам и коллективам

Приложение также будет полезно для командной работы, где сотрудники могут не только отслеживать свои личные отчеты, но и делиться результатами с коллегами. Это помогает повысить уровень ответственности и улучшить взаимодействие внутри команды. Например, в стартапах небольшого размера это может стать удобным инструментом для контроля выполнения задач, а в крупных компаниях — эффективным способом организовать ежедневные отчеты сотрудников и поддерживать прозрачность рабочих процессов. Возможность видеть, что делает команда в целом, способствует лучшему распределению задач, а также позволяет быстрее реагировать на возникающие проблемы.

Цель проекта

Разработать мобильное приложение для Android, позволяющее пользователям создавать ежедневные отчеты о прогрессе, просматривать отчеты других пользователей, а также визуализировать свой прогресс по достижению цели.

Задачи проекта

Задача: Реализовать пользовательский интерфейс (UI) для просмотра отчетов других пользователей.

Описание: Задача включает настройку списка отчетов с возможностью прокрутки, а также добавление обработки ошибок на случай отсутствия отчетов. Результатом выполнения будет полностью функционирующий модуль для просмотра отчетов пользователей.

Ответственный: Гашимов Ильхам

Задача: Подключить API для просмотра отчетов пользователей

Описание: Реализовать модуль для просмотра отчетов пользователей, получаемых через API. Модуль должен отображать описание отчета до 500 символов, оценку по 10-балльной шкале и дату в формате UNIX (которую нужно преобразовать в читаемый вид). Необходимо предусмотреть обработку ошибок, если отчеты отсутствуют или произошла ошибка при загрузке данных, и отображать индикатор загрузки. Результатом будет полностью функционирующий модуль с возможностью прокрутки и корректным отображением данных.

Ответственный: Гашимов Ильхам

Задача: Исправить найденные ошибки в ходе тестирования

Описание: Устранить ошибки, которые влияют на функциональность приложения. После внесения исправлений провести повторное тестирование для проверки стабильности. В процессе работы были исправлены ошибки по моей фиче, а также проконтролировано исправление ошибок другими членами команды по их фичам.

Ответственный: Гашимов Ильхам

Задача: Спроектировать структуру базы данных

Описание: Создать таблицу для пользователей с полями: ID (уникальный идентификатор) и имя. Для отчетов — таблицу с полями: ID (уникальный идентификатор), описание (текст отчета), дата (в формате UNIX), оценка (по 10-балльной шкале) и цель (описание цели). В результате будет разработана схема базы данных с таблицами для пользователей и отчетов, а также настроены связи между ними, что обеспечит правильное хранение и извлечение данных.

Ответственная: Петрова Мария

Задача: Реализовать API для просмотра отчетов

Описание: Необходимо создать API-эндпоинт, который будет возвращать список отчетов пользователей с данными о каждом отчете, включая описание, дату, оценку и цель. В результате будет рабочий API-эндпоинт для получения отчетов.

Ответственная: Петрова Мария

Задача: Реализовать API для создания и удаления отчетов

Описание: Необходимо настроить POST-эндпоинт для создания нового отчета с валидацией данных на сервере, чтобы длина текста описания не превышала 500 символов. Также требуется настроить DELETE-эндпоинт для удаления отчета по его ID. В результате будут рабочие API-эндпоинты для создания и удаления отчетов.

Ответственная: Петрова Мария

Задача: Реализовать прототип экрана просмотра отчетов

Описание: Необходимо создать макет интерфейса, который будет отображать список отчетов с датой, описанием и оценкой. Также предусмотреть кнопку для перехода к экрану создания нового отчета. В результате получится прототип экрана с полным функционалом для просмотра и добавления отчетов.

Ответственная: Коновалова Кира

Задача: Реализовать прототип экрана создания отчетов

Описание: Создать макет с текстовым полем для описания, ползунком для выбора оценки (1–10) и отображением текущей даты. Кнопка сохранения должна быть активна только при заполнении всех полей. Результатом будет прототип экрана создания отчетов с удобным вводом данных.

Ответственная: Коновалова Кира

Задача: Реализовать прототип экрана визуализации прогресса

Описание: Необходимо создать макет интерфейса, который будет отображать прогресс пользователя в виде линейного графика, показывающего оценки по дням.

Ответственная: Коновалова Кира

Задача: Реализовать UI создания отчетов

Описание: Реализовать ввод описания (текст, до 500 символов), выбор оценки (по 10-балльной шкале) и автоматическую установку текущей даты (в формате UNIX)).

Ответственная: Блинова Полина

Задача: Подключить API для создания отчетов пользователей

Описание: Настроить API для отправки данных, включая описание (текст до 500 символов), оценку (по 10-балльной шкале) и дату (в формате UNIX).

Ответственная: Блинова Полина

Задача: Провести функциональное тестирование

Описание: Необходимо протестировать приложение, включая создание отчетов, их просмотр и визуализацию прогресса. Тестирование должно проводиться на устройствах с Android 9 и выше. В результате нужно оформить список найденных ошибок и недочетов.

Ответственная: Блинова Полина

Задача: Реализовать UI визуализации прогресса

Описание: Интегрировать библиотеку графиков (fl_chart или аналогичную) для отображения прогресса пользователя по 10-балльной шкале. График должен поддерживать прокрутку для удобного просмотра данных.

Ответственная: Сусликова Вероника

Задача: Подключить API для визуализации прогресса

Описание: Настроить отображение графиков, используя данные, полученные из API. Графики должны строиться на основе оценок пользователей по 10-балльной шкале и отображаться в соответствующем интерфейсе.

Ответственная: Сусликова Вероника

ОСНОВНАЯ ЧАСТЬ

1 Описание предметной области

Предметная область проекта заключается в разработке мобильного приложения для отслеживания целей с помощью ведения ежедневных отчетов. Приложение должно предоставлять пользователю возможность добавлять отчеты о выполнении задач, а также просматривать отчеты других пользователей. Важной частью системы является возможность удаления и создания отчетов, что важно для обеспечения удобного взаимодействия с данными. Приложение использует API для взаимодействия с серверной частью, где хранятся все отчеты.

Была выявлена необходимость в удобном интерфейсе для ввода и отображения отчетов, а также в функции просмотра отчетов других пользователей. Похожие приложения уже существуют на рынке, однако, они не все предоставляют возможность оценивания личного отчета и просмотра отчетов других пользователей.

2 Методы, средства

2.1 Методы

2.1.1 API-интеграция

Для взаимодействия с сервером используется API, которое позволяет обмениваться данными между мобильным приложением и серверной частью. Это позволяет пользователю получать и отправлять отчеты, а также удалять их.

2.1.2 Аутентификация через Google

Для обеспечения безопасности и удобства входа в приложение используется OAuth-авторизация через Google.

2.1.3 Тестирование приложения

Проводилось тестирование мобильного приложения, включая поиск и исправление ошибок, а также улучшение пользовательского интерфейса на основе полученных данных.

2.2 Средства

2.2.1 Фреймворк Flutter

Используется для разработки кросс-платформенного мобильного приложения.

2.2.2 HTTP-запросы

Для взаимодействия с сервером используется библиотека http, которая позволяет выполнять GET, POST и DELETE запросы.

2.2.3 Flutter Secure Storage

Плагин для безопасного хранения токенов аутентификации на устройстве пользователя.

2.2.4 Google Sign-In

Безопасная система для аутентификации пользователей через учетную запись Google.

3 Суть проекта

Суть проекта заключается в разработке мобильного приложения для отслеживания поставленных целей с помощью ведения отчетов. Основная задача приложения – помочь пользователю в цифровом формате отслеживать свои цели, оценивать их, тем самым способствует саморазвитию. Второстепенная задача приложения – просмотр отчетов других пользователей, что может помочь вам в достижении новых целей. Приложение предлагает удобный интерфейс для записи ежедневных отчетов, оценки выполненных целей и анализа результатов.

4 Процесс работы над проектом

4.1 Планирование и распределение обязанностей

На начальном этапе разработки проекта команда провела обсуждение задач и определила, кто за что будет отвечать. Были выделены основные компоненты проекта, такие как разработка пользовательского интерфейса, интеграция с серверной частью, а также тестирование и исправление ошибок.

Я выбрал для себя задачу разработки пользовательского интерфейса (UI), используя Flutter. Я сосредоточился на создании экранов для отображения отчетов, интеграции с API и обеспечении удобства взаимодействия пользователя с приложением.

Кроме того, я активно взаимодействовал с другими членами команды, которые занимались серверной частью проекта и тестированием. Наше общение было важным для того, чтобы все части приложения правильно работали вместе.

4.2 Настройка окружения для разработки с использованием Flutter

Для начала разработки приложения на Flutter я прошел несколько этапов настройки окружения, которые подробно описаны в документации Flutter[1]. Этот процесс включал в себя установку необходимых инструментов для работы с Flutter на операционной системе Windows.

Первым шагом было скачивание и установка Flutter SDK. После этого я установил Android Studio — официальную среду разработки, которая включает в себя все необходимые компоненты для создания мобильных приложений. В процессе установки я также настроил Android Virtual Device (AVD), чтобы иметь возможность запускать и тестировать приложение на виртуальном устройстве Android.

Затем я убедился, что Flutter и Android Studio правильно интегрированы, используя команду `flutter doctor`, которая проверяет конфигурацию окружения и выявляет возможные проблемы. После выполнения этой команды, я установил все необходимые зависимости и плагины для работы с Dart и Flutter в Android Studio

Когда окружение было настроено, я создал новый проект с помощью команды `flutter create itmomind`. Эта команда автоматически сгенерировала базовую структуру проекта, в которой уже были настроены все необходимые зависимости и файлы для начала работы. Я проверил, что проект успешно компилируется и запускается на виртуальном устройстве.

4.3 Изучение документаций

4.3.1 Изучение документации о структуре Flutter

После того как я настроил окружение, я приступил к изучению структуры проекта Flutter. В документации[2] описывается, как организованы файлы и папки внутри проекта, что помогает лучше ориентироваться в его архитектуре и не запутаться.

Одним из важных этапов было знакомство с папкой `lib`. папка `lib` является основной для размещения кода приложения, в том числе для файла `main.dart`, который является точкой входа в приложение. Я также узнал, что в папках для различных платформ (Android, iOS и других) хранятся настройки и код, специфичные для каждой платформы. Это означает, что для каждой платформы можно изменить то, что вы хотите, что позволяет гибко изменять приложение.

В документации также объясняется, как можно ограничить поддержку проекта только определёнными платформами с помощью команды `flutter create --platforms=android`. Это полезно для проектов, которые ориентированы только на мобильные устройства, например, для Android.

4.3.2 Изучение языка Dart

Важной частью разработки на Flutter является знание языка Dart, на котором пишется весь код приложения. Я изучил основы Dart с помощью документации[3], где рассмотрены его основные особенности, такие как типизация, работа с коллекциями и асинхронное программирование. Dart обладает строгой типизацией, что позволяет минимизировать ошибки на этапе компиляции.

4.3.3 Изучение виджетов Flutter

Важным шагом в освоении Flutter было изучение принципов работы с виджетами[4]. Я познакомился с основными типами виджетов, такими как StatelessWidget и StatefulWidget, а также с их применением для создания интерфейсов. Эти виджеты играют ключевую роль в организации UI в Flutter: StatelessWidget используется для создания неизменяемых виджетов, а StatefulWidget — для тех элементов интерфейса, которые должны изменяться во время работы приложения (например, при взаимодействии с пользователем).

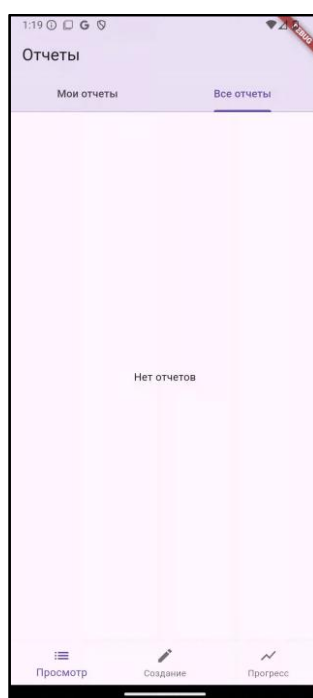
Документация[4] подробно объясняет, как создавать виджеты, как управлять их состоянием и как работать с композициями виджетов для создания сложных интерфейсов.

4.3.4 Изучение стандартных виджетов

Далее я изучил из документации[5] стандартные виджеты Flutter, которые позволяют легко создавать часто используемые элементы интерфейса, такие как Text, Container, Column, Row, ListView и другие. Эти виджеты предоставляют удобные и гибкие способы для создания интерфейсов, которые автоматически подстраиваются под различные размеры экранов и платформы. Я использовал стандартные виджеты для быстрого создания интерфейса моего приложения. Например, для реализации списка задач я применил ListView, который автоматически поддерживает прокрутку, а для ввода данных использовал виджет TextField. Все эти компоненты значительно упростили процесс разработки и позволили быстрее перейти к реализации функционала.

4.4 Реализация интерфейса (UI)

Я был ответственным за разработку интерфейса для просмотра отчетов других пользователей. Задача заключалась в создании списка отчетов с возможностью прокрутки, а также в добавлении обработки ошибок на случай отсутствия отчетов что можно увидеть на анимированном изображении 1. Было важно, чтобы интерфейс был простым и понятным для пользователя, с интуитивно понятной навигацией. Также я интегрировал обработку ошибок, чтобы при отсутствии отчетов или проблемах с загрузкой данных отображался индикатор загрузки.



Анимированный рисунок 1 - Рисунок, который иллюстрирует отображение при отсутствии отчетов

4.5 Интеграция с API

Следующим этапом было подключение API для получения отчетов пользователей. Это включало разработку модулей для отображения отчетов, получение данных с сервера, обработку ошибок, а также форматирование даты в читаемый вид (перевод Unix-времени в обычную дату). Я реализовал обработку ошибок для случаев, когда отчеты отсутствуют или произошла ошибка на сервере.

4.6 Тестирование

В процессе тестирования были выявлены некоторые ошибки в функциональности, которые нужно было исправить. Это касалось не только функционала, за который я был ответственен, но и других частей приложения, которые разработали мои коллеги. При тестировании функции удаления отчетов возникла ошибка, связанная с передачей неправильного идентификатора для удаления отчета.

Ошибка заключалась в том, что вместо корректного ID отчета в запросе использовался ID пользователя, который не существовал в контексте запроса на удаление отчета.

4.7 Исправление ошибок

В процессе работы над проектом я устранял баги, которые возникали в ходе тестирования. Эти ошибки касались как взаимодействия с API, так и отображения данных в интерфейсе. Одним из таких багов было неправильное удаление отчетов, как я описал выше. Ошибка заключалась в передаче ID пользователя вместо ID отчета, что приводило к ошибке 404 на сервере.

После всех исправлений я провел повторное тестирование, чтобы убедиться в стабильности работы приложения и устранении всех выявленных ошибок.

5 Раскрыть суть проблемы, поставленной передо мной

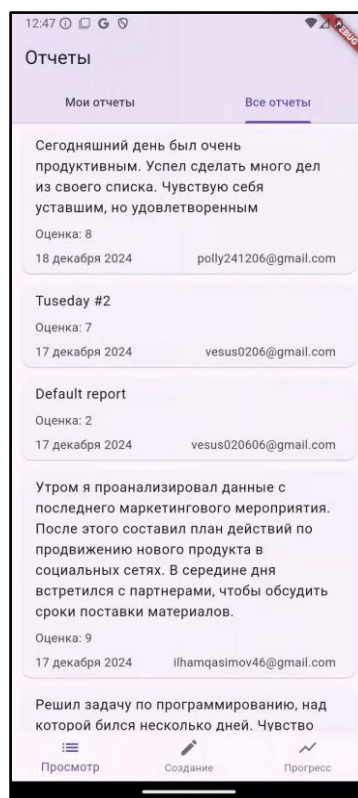
Моя основная задача заключалась в реализации пользовательского интерфейса (UI) для просмотра отчетов других пользователей и интеграции с API для получения отчетов. Проблема, которая была поставлена, заключалась в создании удобного и функционального интерфейса, который бы не только позволял пользователю просматривать отчеты других участников, но и обеспечивал бы правильную обработку ошибок, если отчеты не были доступны или произошла ошибка при их загрузке.

Также нужно было учитывать, что отчеты пользователей могут содержать данные в разных форматах, включая время в формате Unix, которое выглядит как длинное число. Эти данные необходимо было преобразовать в удобный для чтения формат, в обычную дату. Задача оказалась сложной, потому что приходилось интегрировать различные части приложения, например, обрабатывать ошибки и работать с данными, которые загружаются асинхронно, то есть не сразу, а по мере получения с сервера.

6 Как я решал поставленную задачу

6.1 Разработка интерфейса

Я создал экран с табами для переключения между моими отчетами и отчетами других пользователей. Для отображения отчетов использовал виджет ListView, что позволяло реализовать прокрутку, которую можно увидеть на анимированном изображении 1. Кроме того, я добавил обработку ошибок, чтобы при отсутствии отчетов или ошибках загрузки показывался индикатор.



Анимированный рисунок 1 - Анимация, показывающая наличие прокрутки в списке отчетов

6.2 Подключение API

Для работы с API я использовал сервис ApiService, который выполнял запросы на сервер, получая данные о отчетах. Я реализовал методы для получения отчетов пользователя и всех отчетов с сервера, а также обработку ошибок и отображение данных.

6.3 Работа с данными

Я научился работать с данными, поступающими с сервера, правильно их обрабатывать и отображать, что можно увидеть на рисунке 1. Например, для правильного отображения даты я использовал библиотеку intl для преобразования времени из Unix-формата в читаемый вид.

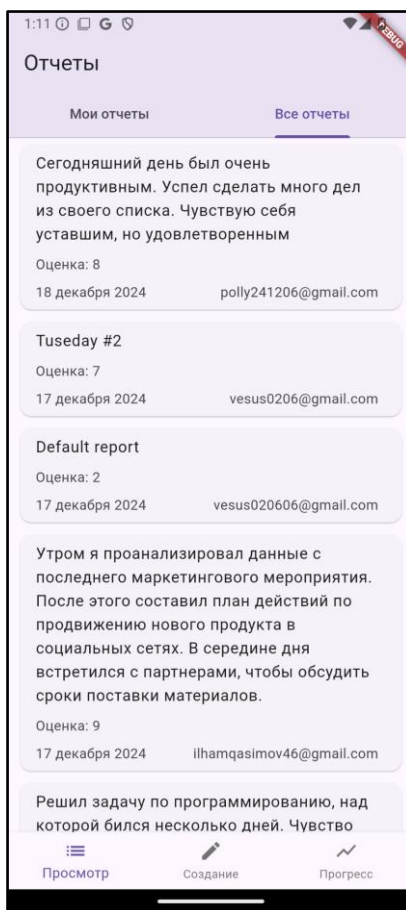


Рисунок 1 - Рисунок, показывающий правильную обработку и отображение данных, полученных с API

6.4 Исправление ошибок

В ходе тестирования приложения я столкнулся с ошибкой, когда при попытке удалить отчет отправлялся неправильный идентификатор. Вместо ID отчета в запросе использовался ID пользователя, что приводило к ошибке 404 — "Не найдено". Это происходило из-за того, что в коде запроса на удаление отчета я случайно передал ID пользователя. После того как я исправил этот момент и стал передавать правильный идентификатор отчета, ошибка была устранена. Затем я снова протестировал приложение, чтобы убедиться, что оно работает стабильно, и убедился, что проблема была решена.

7 Анализ работы над курсовым проектом

В процессе выполнения курсового проекта мне удалось успешно реализовать все основные задачи, которые ставились передо мной. Однако, как и в любой работе, возникли трудности, которые я решил со своим руководителем команды. Мне удалось работать планомерно.

При работе так же я освоил основные навыки Front-end части программирования, хотя до этого я этим не занимался, синтаксис был весьма современным, все было сделано для удобства.

Курсовой проект продемонстрировал значительный успех в разработке мобильного приложения для отслеживания целей, но в процессе работы были также выявлены трудности, связанные с корректной обработкой данных, исправлением ошибок. Важно отметить, что процесс разработки включал в себя ключевые этапы: проектирование, разработка, тестирование и исправление ошибок, что позволило достичь высокого качества конечного продукта. Работа над проектом позволила глубже понять особенности разработки мобильных приложений с использованием Flutter и интеграции с серверной частью, так же работы в команде.

8 Взаимодействие с командой

На начальном этапе работы над проектом наша команда провела несколько встреч для обсуждения целей и распределения задач. Важно было понимать, кто будет заниматься какой частью работы.

Я тесно работал с Петровой Марией, которая занималась созданием API для получения и отправки отчетов. Нам необходимо было синхронизировать наши работы, чтобы интерфейс приложения корректно взаимодействовал с сервером. Я предоставлял ей требования к API для отображения отчетов, а также информацию о форматах данных, таких как преобразование Unix-времени в обычную дату. Мария, в свою очередь, делала соответствующие изменения на серверной стороне, чтобы обеспечить правильную обработку запросов.

Я также взаимодействовал с Кирой Коноваловой, которая занималась прототипированием экранов приложения. Мы обсуждали, как именно будут выглядеть экраны для просмотра отчетов, создания новых отчетов и визуализации прогресса.

В процессе тестирования я активно взаимодействовал с Блиновой Полиной, которая занималась функциональным тестированием. Мы регулярно проводили тесты интерфейса и интеграции с API, чтобы выявить ошибки. После каждого этапа тестирования я получал обратную связь, исправлял выявленные проблемы и снова тестировал приложение. Например, одна из ошибок касалась неправильной передачи идентификатора при удалении отчета, которую мне пришлось исправить в процессе тестирования.

Мое взаимодействие с командой было продуктивным. Мы активно обменивались идеями, решали возникающие проблемы и помогали друг другу в решении задач. Благодаря тесной работе мы добились успеха.

9 Взаимодействие с руководителем проекта и его оценка

В процессе работы над курсовым проектом важным оказалось регулярное общение с руководителем проекта, Алибеков Олег Олегович. Взаимодействие с руководителем проекта помогало улучшать качество работы и повышать уровень знаний в Flutter.

Руководитель проекта участвовал в распределении обязанностей между членами команды. Это позволило оптимально организовать рабочие процессы и также сосредоточиться на разработке интерфейса (UI) и интеграции с сервером.

Руководитель всегда давал четкие рекомендации и наставления, что позволило мне улучшить качество работы над проектом. В итоге оцениваю руководителя на отлично. Он показал себя как опытного разработчика и грамотного лидера команды, что позволило участинкам команды достичь всех поставленных целей.

ЗАКЛЮЧЕНИЕ

Проект был успешным и достиг всех поставленных целей. Тем самым проект можно оценить на отлично. Мы разработали мобильное приложение для отслеживания целей для Android, который позволяет отслеживать поставленные цели и следить за прогрессом других пользователей, а главной фишкой является возможность оценивать свой отчет, и тем самым другие пользователи могут сделать выводы для себя.

Все задачи были выполнены. В ходе разработки и тестирования удалось устранить ошибки и обеспечить стабильную работу приложения.

Мой вклад в проект был не менее важным, я занимался разработкой пользовательского интерфейса, интеграцией API и тестированием продукта. Также я активно взаимодействовал с командой, что позволило почувствовать атмосферу командной работы.

Этот опыт стал важным шагом в мое будущее в сфере программирования, что является ценным вкладом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Настройка окружения Flutter // docs.flutter.dev URL:
<https://docs.flutter.dev/get-started/install>
2. Flutter: структура проекта // education.yandex.ru URL:
<https://education.yandex.ru/handbook/flutter/article/flutter-struktura-proekta>
3. Dart: особенности языка // education.yandex.ru URL:
<https://education.yandex.ru/handbook/flutter/article/dart-osobennosti-yazyka>
4. Flutter: виджеты // education.yandex.ru URL:
<https://education.yandex.ru/handbook/flutter/article/widgets-basics-stless-stful-inherited>
5. Flutter: стандартные виджеты // education.yandex.ru URL:
<https://education.yandex.ru/handbook/flutter/article/widgets-standard-widgets>