Оглавление

ВВЕДЕНИЕ 5

1 АНАЛИЗ И ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛА ПРИЛОЖЕНИЯ (ПЛАНИРОВЩИК ФАЗА 1, РЕГИСТРАЦИЯ/ВХОД, АНКЕТИРОВАНИЕ) 8

1.1 Исходные данные для отчета 8

1.2 Анализ предметной области и существующих решений 9

1.3 Анализ требований пользователей к разрабатываемой системе 10

1.3.1 Требования к Фазе 1 Планировщика 10

1.3.2 Требования к системе регистрации/входа 11

1.3.3 Требования к анкете после регистрации 11

1.4 Проектирование Фазы 1 Планировщика 12

1.4.1 Описание этапов Фазы 1 Планировщика 12

1.4.2 Логика работы и переходы между состояниями в Фазе 1 Планировщика 13

1.4.3 Детализация шагов на каждом этапе Фазы 1 Планировщика 14

1.5 Проектирование системы регистрации/входа с использованием Firebase 15

1.5.1 Обзор Firebase Authentication 15

1.5.2 Настройка Firebase в проекте 16

1.5.3 Реализация функций регистрации и входа 17

1.6 Проектирование анкеты пользователя после регистрации 18

1.6.1 Цели и задачи анкетирования 18

1.6.2 Структура и типы вопросов в анкете 19

1.6.3 Сохранение и обработка данных анкеты 19

1.7 Техническое задание на разработку компонентов системы 20

1.7.1 Общие сведения 21

1.7.2 Цели и назначение создания системы 22

1.7.3 Характеристика объектов автоматизации 23

1.7.4 Требования к системе 23

1.7.5 Состав и содержание работ по созданию системы 24

1.7.6 Порядок разработки системы 24

1.7.7 Порядок контроля и приемки системы 25

1.7.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 25

1.7.9 Требования к документированию 26

1.7.10 Источники разработки 26

1.8 Выводы по разделу 27

ЗАКЛЮЧЕНИЕ 30

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 32

ВВЕДЕНИЕ

Основная цель преддипломной практики состоит в сборе фактического материала для выполнения выпускной квалификационной работы. В рамках данной работы рассматривается проектирование и разработка ключевых компонентов мобильного приложения, включая первую фазу планировщика задач, систему аутентификации пользователей с использованием Firebase и механизм анкетирования после регистрации.

Современные мобильные приложения часто требуют наличия удобного инструмента для планирования задач, а также безопасной и надежной системы управления учетными записями пользователей. Интеграция с Firebase позволяет упростить разработку аутентификации и обеспечить масштабируемость. Анкетирование новых пользователей помогает собрать важную информацию для персонализации опыта использования приложения и адаптации функционала под нужды целевой аудитории.

Данный отчет детализирует анализ требований, проектирование архитектуры и логики указанных компонентов. Описываются этапы работы планировщика, процесс настройки и интеграции Firebase Authentication, а также структура и назначение анкеты пользователя. Результаты этой работы лягут в основу дальнейшей разработки и реализации полнофункционального мобильного приложения.

Цель практики: получение практического опыта в анализе, проектировании и документировании разработки компонентов современного мобильного приложения, включая планировщик, систему аутентификации и анкетирование.

Задачи практики:

- Изучение предметной области: планировщики задач, системы аутентификации, анкетирование в мобильных приложениях.

- Анализ требований пользователей к разрабатываемым компонентам.

- Проектирование архитектуры и логики Фазы 1 планировщика.

- Исследование и описание процесса интеграции Firebase Authentication для регистрации и входа пользователей.

- Проектирование структуры анкеты для сбора данных о пользователях после регистрации.

- Составление технического задания на разработку описанных компонентов.

- Оформление отчета о проделанной работе в соответствии с установленными требованиями.

Источниками информации явились техническая документация Firebase, статьи и публикации по разработке мобильных приложений, а также открытые источники, описывающие лучшие практики проектирования пользовательских интерфейсов и систем управления задачами.

1 АНАЛИЗ И ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛА ПРИЛОЖЕНИЯ (ПЛАНИРОВЩИК ФАЗА 1, РЕГИСТРАЦИЯ/ВХОД, АНКЕТИРОВАНИЕ)

1.1 Исходные данные для отчета

Тема ВКР: «Разработка мобильного приложения с интегрированным планировщиком задач, системой аутентификации Firebase и анкетированием пользователей»

Процесс, рассматриваемый в рамках практики: «Анализ, проектирование и документирование Фазы 1 планировщика, системы регистрации/входа пользователей через Firebase и анкеты для сбора пользовательских данных в мобильном приложении.»

Подразделение, отвечающее за реализацию данного процесса: «Отдел разработки мобильных приложений (условное название)»

Ссылка на git-репозиторий: https://github.com/user/project-name (пример)

Учетные данные пользователя (тестовые, если применимо):

Логин: testuser@example.com, Пароль: password123

1.2 Анализ предметной области и существующих решений

В данном разделе проводится обзор существующих мобильных приложений с функциями планировщика, анализируются их сильные и слабые стороны. Рассматриваются различные подходы к реализации систем аутентификации, с акцентом на преимущества использования облачных сервисов, таких как Firebase. Также исследуются практики проведения анкетирования пользователей в мобильных приложениях для сбора первичной информации и улучшения пользовательского опыта.

Особое внимание уделяется изучению популярных приложений-планировщиков (например, Todoist, Microsoft To Do, Any.do) для выявления ключевых функций, востребованных пользователями. Анализируются механизмы регистрации и входа, используемые в этих приложениях, и оценивается их удобство и безопасность.

1.3 Анализ требований пользователей к разрабатываемой системе

На основе анализа предметной области и потенциальных сценариев использования приложения формулируются требования пользователей к основным компонентам.

1.3.1 Требования к Фазе 1 Планировщика

Фаза 1 планировщика должна предоставлять базовый, но достаточный функционал для управления задачами. Пользователи ожидают:

- Возможность создавать задачи с указанием названия, описания и срока выполнения.

- Просмотр списка созданных задач.

- Возможность помечать задачи как выполненные.

- Простой и интуитивно понятный интерфейс для взаимодействия с задачами.

- Возможность удалять задачи.

- Сортировка задач (например, по дате создания или сроку выполнения).

1.3.2 Требования к системе регистрации/входа

Система регистрации и входа должна быть безопасной и удобной:

- Регистрация новых пользователей с использованием адреса электронной почты и пароля.

- Вход существующих пользователей с использованием их учетных данных.

- Возможность восстановления пароля (может быть рассмотрена для последующих фаз).

- Надежное хранение учетных данных с использованием Firebase Authentication.

- Понятные сообщения об ошибках при регистрации или входе (например, "пользователь с таким email уже существует", "неверный пароль").

1.3.3 Требования к анкете после регистрации

Анкета после регистрации предназначена для сбора первоначальной информации о пользователе:

- Анкета должна появляться однократно после первой успешной регистрации.

- Вопросы должны быть понятными и не занимать много времени для ответа.

- Данные анкеты должны безопасно сохраняться для дальнейшего анализа (например, в Firebase Firestore или Realtime Database).

- Возможность пропустить анкету (если это предусмотрено логикой приложения).

- Цель анкеты – собрать информацию для персонализации контента или функционала приложения (например, интересы пользователя, цели использования планировщика).

1.4 Проектирование Фазы 1 Планировщика

Фаза 1 планировщика является ядром функционала управления задачами. Проектирование включает определение основных этапов, логики работы и пользовательских сценариев.

1.4.1 Описание этапов Фазы 1 Планировщика

Фаза 1 планировщика включает следующие ключевые этапы (состояния или экраны):

1. \*\*Экран списка задач (главный экран планировщика)\*\*:

\* Отображение всех активных задач пользователя.

\* Возможность перехода к созданию новой задачи.

\* Индикация выполненных/невыполненных задач.

2. \*\*Экран создания/редактирования задачи\*\*:

\* Форма для ввода названия задачи.

\* Поле для ввода детального описания задачи (опционально для Фазы 1).

\* Выбор даты и времени срока выполнения (дедлайна).

\* Кнопки для сохранения или отмены создания/редактирования.

3. \*\*Экран просмотра задачи (детальный экран)\*\*:

\* Отображение полной информации о задаче (название, описание, срок).

\* Возможность пометить задачу как выполненную/невыполненную.

\* Возможность редактирования или удаления задачи.

1.4.2 Логика работы и переходы между состояниями в Фазе 1 Планировщика

Логика работы планировщика строится на взаимодействии пользователя с задачами и переходах между экранами:

- \*\*При запуске приложения (после входа)\*\*: Пользователь попадает на экран списка задач. Если задач нет, отображается соответствующее сообщение или подсказка о создании первой задачи.

- \*\*Создание задачи\*\*:

\* Пользователь нажимает кнопку "Добавить задачу" (или аналогичную).

\* Переход на экран создания задачи.

\* Пользователь заполняет поля (минимум – название).

\* При сохранении: задача добавляется в общий список и сохраняется в базе данных (например, Firebase Firestore). Пользователь возвращается на экран списка задач, где видит новую задачу.

\* При отмене: пользователь возвращается на экран списка задач без сохранения.

- \*\*Взаимодействие с задачей из списка\*\*:

\* \*\*Отметка о выполнении\*\*: Пользователь может быстро пометить задачу как выполненную прямо из списка (например, с помощью чек-бокса). Статус задачи обновляется в базе данных и в интерфейсе.

\* \*\*Просмотр/Редактирование\*\*: При выборе задачи из списка пользователь переходит на экран просмотра задачи. Оттуда он может перейти в режим редактирования (аналогично экрану создания) или выполнить другие действия (удалить, изменить статус).

- \*\*Удаление задачи\*\*:

\* Пользователь может удалить задачу (например, с экрана просмотра задачи или через контекстное меню в списке).

\* Задача удаляется из базы данных и из списка в интерфейсе. Требуется подтверждение удаления.

1.4.3 Детализация шагов на каждом этапе Фазы 1 Планировщика

\*\*Этап 1: Создание новой задачи\*\*

1. Пользователь нажимает на кнопку «+» (или «Добавить задачу») на экране списка задач.

2. Система отображает экран создания задачи с полями: «Название», «Описание» (опционально), «Срок выполнения».

3. Пользователь вводит название задачи.

4. Пользователь (опционально) вводит описание задачи.

5. Пользователь (опционально) выбирает дату и время срока выполнения.

6. Пользователь нажимает кнопку «Сохранить».

7. Система валидирует введенные данные (например, название не должно быть пустым).

8. Система сохраняет задачу в базе данных.

9. Система перенаправляет пользователя на экран списка задач, где отображается новая задача.

\*\*Этап 2: Просмотр списка задач\*\*

1. Пользователь открывает приложение и успешно входит в систему.

2. Система отображает главный экран планировщика со списком ранее созданных задач.

3. Для каждой задачи в списке отображается: название, срок выполнения (если указан), статус (выполнена/не выполнена).

4. Пользователь может прокручивать список, если задач много.

\*\*Этап 3: Взаимодействие с задачей (выполнение, редактирование, удаление)\*\*

\* \*\*Пометка о выполнении:\*\*

1. Пользователь находит нужную задачу в списке.

2. Пользователь нажимает на чек-бокс (или аналогичный элемент) рядом с задачей.

3. Система обновляет статус задачи на «выполнена» в интерфейсе (например, зачеркивает текст задачи) и в базе данных.

\* \*\*Редактирование задачи:\*\*

1. Пользователь выбирает задачу из списка для детального просмотра.

2. Система отображает экран просмотра задачи.

3. Пользователь нажимает кнопку «Редактировать».

4. Система отображает экран редактирования задачи с предзаполненными данными.

5. Пользователь вносит необходимые изменения.

6. Пользователь нажимает кнопку «Сохранить».

7. Система обновляет данные задачи в базе данных и возвращает пользователя на экран просмотра задачи или список задач.

\* \*\*Удаление задачи:\*\*

1. Пользователь выбирает задачу из списка для детального просмотра или использует опцию удаления из списка (например, свайп или долгое нажатие).

2. Система запрашивает подтверждение удаления.

3. Пользователь подтверждает удаление.

4. Система удаляет задачу из базы данных и обновляет список задач в интерфейсе.

1.5 Проектирование системы регистрации/входа с использованием Firebase

Firebase Authentication предоставляет готовые решения для управления пользователями, что значительно упрощает разработку.

1.5.1 Обзор Firebase Authentication

Firebase Authentication – это сервис от Google, который позволяет легко добавлять аутентификацию пользователей в веб- и мобильные приложения. Он поддерживает различные методы аутентификации, включая:

- Email и пароль

- Номера телефонов

- Аккаунты Google, Facebook, Twitter и др.

Для данного проекта в Фазе 1 будет использоваться аутентификация по email и паролю. Firebase берет на себя обработку и хранение учетных данных, обеспечивая безопасность.

Ключевые преимущества:

- \*\*Безопасность\*\*: Firebase использует стандартные отраслевые практики для защиты данных пользователей.

- \*\*Простота интеграции\*\*: Firebase предоставляет SDK для различных платформ (Android, iOS, Web), что упрощает интеграцию.

- \*\*Масштабируемость\*\*: Сервис рассчитан на большие нагрузки.

- \*\*Бесплатный стартовый план\*\*: Для большинства небольших и средних проектов бесплатного плана достаточно.

1.5.2 Настройка Firebase в проекте

Настройка Firebase включает следующие шаги:

1. \*\*Создание проекта в Firebase Console\*\*:

\* Перейти на сайт Firebase (firebase.google.com).

\* Войти с помощью Google аккаунта.

\* Создать новый проект, указав его название и регион.

2. \*\*Подключение приложения к Firebase\*\*:

\* В Firebase Console выбрать созданный проект.

\* Добавить приложение (Android или iOS), следуя инструкциям мастера подключения. Это включает указание имени пакета (для Android) или Bundle ID (для iOS).

\* Скачать конфигурационный файл (google-services.json для Android или GoogleService-Info.plist для iOS) и добавить его в проект мобильного приложения согласно инструкциям.

3. \*\*Добавление Firebase SDK в приложение\*\*:

\* В файлах сборки проекта (например, build.gradle для Android или Podfile для iOS) добавить необходимые зависимости Firebase Authentication.

\* Синхронизировать проект для загрузки SDK.

4. \*\*Включение метода аутентификации\*\*:

\* В Firebase Console перейти в раздел "Authentication".

\* На вкладке "Sign-in method" (Методы входа) включить провайдер "Email/Password".

1.5.3 Реализация функций регистрации и входа

\*\*Регистрация нового пользователя (Sign Up)\*\*:

1. Пользователь вводит свой email и пароль на экране регистрации.

2. Приложение вызывает метод Firebase SDK для создания нового пользователя, например, `createUserWithEmailAndPassword(email, password)`.

3. Firebase Authentication проверяет корректность email, сложность пароля (если настроено) и создает нового пользователя в системе.

4. В случае успеха, пользователь считается аутентифицированным, и приложение может перенаправить его на следующий экран (например, анкета или главный экран).

5. В случае ошибки (например, email уже используется, слабый пароль), Firebase возвращает ошибку, которую приложение должно обработать и показать пользователю.

\*\*Вход существующего пользователя (Sign In/Login)\*\*:

1. Пользователь вводит свой email и пароль на экране входа.

2. Приложение вызывает метод Firebase SDK для входа, например, `signInWithEmailAndPassword(email, password)`.

3. Firebase Authentication проверяет предоставленные учетные данные.

4. В случае успеха, пользователь считается аутентифицированным. Приложение получает информацию о пользователе (например, UID) и может сохранить сессию.

5. В случае ошибки (например, неверный пароль, пользователь не найден), Firebase возвращает ошибку для обработки.

\*\*Отслеживание состояния аутентификации\*\*:

Firebase SDK позволяет отслеживать текущее состояние аутентификации пользователя (вошел он в систему или нет). Это используется для автоматического входа при повторном запуске приложения или для перенаправления на экран входа, если пользователь не аутентифицирован.

1.6 Проектирование анкеты пользователя после регистрации

Анкета после регистрации служит для сбора дополнительной информации о пользователе, которая может быть использована для персонализации приложения или аналитики.

1.6.1 Цели и задачи анкетирования

- \*\*Персонализация\*\*: Сбор данных об интересах, предпочтениях или целях пользователя для адаптации контента или функционала приложения (например, предлагать релевантные категории задач в планировщике).

- \*\*Сегментация пользователей\*\*: Группировка пользователей по определенным признакам для маркетинговых активностей или анализа поведения различных групп.

- \*\*Улучшение продукта\*\*: Получение обратной связи о том, что пользователи ожидают от приложения.

Анкета должна быть краткой и показываться один раз – сразу после первой успешной регистрации и входа пользователя.

1.6.2 Структура и типы вопросов в анкете

Примерные вопросы для анкеты:

1. \*\*Основная цель использования планировщика\*\*:

\* Тип: Одиночный выбор (Radio button)

\* Варианты: "Работа/Проекты", "Учеба", "Личные дела/Хобби", "Все вместе", "Другое"

2. \*\*Как вы обычно организуете свои задачи?\*\*:

\* Тип: Множественный выбор (Checkboxes)

\* Варианты: "Списки дел", "Календарь", "Заметки", "Стикеры", "В уме"

3. \*\*Какие функции для вас наиболее важны в планировщике? (выберите до 3-х)\*\*:

\* Тип: Множественный выбор с ограничением

\* Варианты: "Напоминания", "Повторяющиеся задачи", "Совместная работа", "Приоритеты задач", "Теги/Категории", "Аналитика продуктивности"

4. \*\*Ваш возраст (опционально)\*\*:

\* Тип: Выпадающий список или ввод числа

\* Варианты: "<18", "18-24", "25-34", "35-44", "45+"

5. \*\*Ваш род деятельности (опционально)\*\*:

\* Тип: Текстовое поле (короткий ответ)

Интерфейс анкеты должен быть простым и понятным, с четкими инструкциями. Должна быть предусмотрена кнопка "Завершить" или "Продолжить" для отправки данных.

1.6.3 Сохранение и обработка данных анкеты

Данные, собранные через анкету, должны быть сохранены. Для этого можно использовать:

- \*\*Firebase Firestore\*\*: NoSQL документо-ориентированная база данных, хорошо подходит для хранения структурированных данных, таких как ответы пользователей. Для каждого пользователя (по его UID из Firebase Authentication) можно создать документ, в котором будут храниться его ответы.

- \*\*Firebase Realtime Database\*\*: Другая NoSQL база данных от Firebase, работающая с данными в формате JSON и синхронизирующая их в реальном времени.

При сохранении важно обеспечить связь ответов с конкретным пользователем. После отправки анкеты пользователь перенаправляется на главный экран приложения.

1.7 Техническое задание на разработку компонентов системы

ТЗ СТРОГО ПО ГОСТ 34.602-2020 (адаптировано для учебного проекта).

1.7.1 Общие сведения

- \*\*Полное наименование системы\*\*: Мобильное приложение "Персональный Планировщик" (условное название). Условное обозначение: ПП-МП-Ф1.

- \*\*Шифр темы\*\*: ПП\_ВКР\_2023 (пример).

- \*\*Наименование организации-заказчика\*\*: Учебный проект кафедры ИС ЧОУ ВО «Московский университет им. С.Ю. Витте».

- \*\*Наименование организации-разработчика\*\*: Студент ФИО, группа ХХХ.

- \*\*Основание для создания системы\*\*: Задание на преддипломную практику, тема ВКР.

- \*\*Плановые сроки начала и окончания работ\*\*: В соответствии с графиком практики и ВКР.

- \*\*Источники и порядок финансирования\*\*: Учебный проект, без внешнего финансирования.

1.7.2 Цели и назначение создания системы

- \*\*Цели создания системы\*\*:

\* Разработать Фазу 1 мобильного приложения-планировщика с базовым функционалом управления задачами.

\* Реализовать безопасную систему регистрации и аутентификации пользователей с использованием Firebase Authentication.

\* Внедрить механизм анкетирования новых пользователей для сбора данных и последующей персонализации.

\* Обеспечить интуитивно понятный пользовательский интерфейс.

- \*\*Назначение системы\*\*:

\* Предоставление пользователям инструмента для создания, отслеживания и управления личными или рабочими задачами.

\* Обеспечение безопасного доступа к данным пользователя через систему учетных записей.

\* Сбор информации о предпочтениях пользователей для улучшения качества сервиса.

Автоматизируемая деятельность: управление задачами, аутентификация пользователей, сбор данных через анкетирование.

1.7.3 Характеристика объектов автоматизации

Объектами автоматизации являются процессы взаимодействия пользователя с мобильным приложением:

- Процесс создания и управления задачами в планировщике (добавление, просмотр, редактирование, удаление, изменение статуса).

- Процесс регистрации нового пользователя и входа существующего пользователя в систему.

- Процесс заполнения анкеты пользователем после первой регистрации.

Система предназначена для индивидуальных пользователей на платформах Android и/или iOS.

1.7.4 Требования к системе

- \*\*Требования к структуре системы в целом\*\*:

\* Модульная архитектура, включающая модуль планировщика, модуль аутентификации, модуль анкетирования.

\* Клиент-серверное взаимодействие с Firebase для аутентификации и хранения данных.

- \*\*Требования к функциям, выполняемым системой\*\*:

\* \*\*Планировщик (Фаза 1)\*\*: создание задач (название, описание, срок), просмотр списка задач, отметка о выполнении, удаление задач.

\* \*\*Аутентификация\*\*: регистрация по email/паролю, вход по email/паролю, управление сессией пользователя.

\* \*\*Анкетирование\*\*: отображение анкеты после первой регистрации, сбор ответов, сохранение ответов в привязке к пользователю.

- \*\*Требования к видам обеспечения\*\*:

\* \*\*Программное обеспечение\*\*: разработка на нативном языке (Java/Kotlin для Android, Swift/Objective-C для iOS) или кроссплатформенном фреймворке (например, Flutter, React Native). Использование Firebase SDK.

\* \*\*Информационное обеспечение\*\*: данные пользователя (задачи, учетные данные, ответы на анкету) хранятся в Firebase.

\* \*\*Лингвистическое обеспечение\*\*: интерфейс на русском языке.

- \*\*Общие технические требования\*\*:

\* Работа на мобильных устройствах под управлением ОС Android (версии X.X и выше) и/или iOS (версии Y.Y и выше).

\* Обеспечение безопасности пользовательских данных при передаче и хранении.

\* Стабильная работа основных функций.

1.7.5 Состав и содержание работ по созданию системы

1. Анализ требований и проектирование (текущий этап, описанный в отчете).

2. Разработка пользовательского интерфейса для экранов планировщика, регистрации/входа, анкеты.

3. Реализация логики Фазы 1 планировщика (создание, отображение, управление задачами).

4. Интеграция Firebase Authentication (настройка, реализация функций регистрации и входа).

5. Разработка модуля анкетирования (отображение, сбор и сохранение данных).

6. Тестирование компонентов и системы в целом.

7. Подготовка документации.

1.7.6 Порядок разработки системы

Разработка ведется итерационно.

1. Утверждение ТЗ (данного документа).

2. Разработка и тестирование модуля аутентификации.

3. Разработка и тестирование Фазы 1 планировщика.

4. Разработка и тестирование модуля анкетирования.

5. Интеграционное тестирование.

Документирование ведется параллельно с разработкой.

1.7.7 Порядок контроля и приемки системы

- \*\*Виды испытаний\*\*: модульное тестирование, интеграционное тестирование, пользовательское тестирование (приемка).

- \*\*Приемка работ\*\*: демонстрация работоспособности функционала, соответствия требованиям ТЗ.

- \*\*Статус приемочной комиссии\*\*: руководитель практики/научный руководитель ВКР.

1.7.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

- Для учебного проекта не требуется сложных мероприятий. Основное – наличие тестового окружения (эмулятор/реальное устройство) с доступом в интернет для работы с Firebase.

- Обучение пользователя не требуется (интерфейс должен быть интуитивно понятен).

1.7.9 Требования к документированию

- Отчет о преддипломной практике (данный документ).

- Пояснительная записка к ВКР (включающая описание разработанной системы).

- Комментарии в коде.

Использование стандартов документирования кода, принятых для выбранного языка/платформы.

1.7.10 Источники разработки

- Документация Firebase (firebase.google.com/docs).

- Официальная документация по разработке для Android/iOS или выбранного кроссплатформенного фреймворка.

- Статьи и руководства по проектированию UI/UX для мобильных приложений.

- Примеры и шаблоны ТЗ (ГОСТ 34.602-2020).

1.8 Выводы по разделу

В данном разделе был проведен анализ и проектирование ключевых компонентов мобильного приложения: Фазы 1 планировщика, системы регистрации/входа с использованием Firebase и анкеты пользователя.

Основные результаты:

1. \*\*Сформулированы требования пользователей\*\* к каждому из компонентов, что позволило определить основной функционал и ожидания от системы.

2. \*\*Разработана архитектура и логика Фазы 1 планировщика\*\*, включая описание его этапов, переходов между состояниями и детализацию шагов пользователя. Это закладывает основу для последующей UI/UX разработки и программирования.

3. \*\*Описан процесс интеграции Firebase Authentication\*\*, включая этапы настройки и принципы реализации функций регистрации и входа. Выбор Firebase позволяет обеспечить безопасность и сократить время разработки модуля аутентификации.

4. \*\*Спроектирована анкета пользователя\*\*, определены ее цели, структура вопросов и способы сохранения данных. Это позволит собирать важную информацию для персонализации приложения.

5. \*\*Составлено техническое задание\*\* на разработку указанных компонентов, что формализует требования и порядок работ.

Проделанная работа позволяет перейти к следующим этапам разработки мобильного приложения. Полученные знания и навыки соответствуют следующим компетенциям (примеры):

- \*\*ПК-1\*\*: Способность разрабатывать прикладное программное обеспечение, проектировать его архитектуру и бизнес-логику. (Проектирование планировщика, логики Firebase).

- \*\*ПК-X\*\*: Знание современных технологий и инструментов разработки мобильных приложений (Использование Firebase).

- \*\*ПК-Y\*\*: Умение анализировать требования пользователей и преобразовывать их в технические спецификации. (Анализ требований, составление ТЗ).

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы, описанной в данном отчете, был проведен комплексный анализ и проектирование начальной фазы мобильного приложения, включающего планировщик задач, систему аутентификации на базе Firebase и механизм анкетирования пользователей.

Были детально рассмотрены требования пользователей к каждому из модулей, что позволило заложить функциональную основу для дальнейшей разработки. Проектирование Фазы 1 планировщика определило его ключевые экраны, логику взаимодействия и основные пользовательские сценарии. Это создало четкое видение того, как пользователи будут взаимодействовать с основной функцией приложения.

Исследование и описание процесса интеграции Firebase Authentication показало преимущества использования готовых облачных решений для задач аутентификации, таких как повышение безопасности и ускорение разработки. Были описаны конкретные шаги по настройке Firebase и реализации функций регистрации и входа.

Проектирование анкеты после регистрации направлено на сбор ценной информации о пользователях, которая в дальнейшем может быть использована для персонализации контента и улучшения пользовательского опыта.

Составленное техническое задание служит формальным документом, который определяет требования к системе и этапы ее создания, что важно для структурированного подхода к разработке.

Результаты данной работы являются важным этапом в цикле создания мобильного приложения и предоставляют необходимую базу для перехода к непосредственной разработке пользовательского интерфейса, программированию логики и тестированию. Полученный опыт в анализе, проектировании и документировании является ценным для формирования профессиональных компетенций в области разработки программного обеспечения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Официальная документация Firebase. [Электронный ресурс] URL: https://firebase.google.com/docs (дата обращения: ДД.ММ.ГГГГ).

2. Документация для разработчиков Android. [Электронный ресурс] URL: https://developer.android.com/docs (дата обращения: ДД.ММ.ГГГГ).

3. Документация для разработчиков Apple iOS. [Электронный ресурс] URL: https://developer.apple.com/documentation/ (дата обращения: ДД.ММ.ГГГГ).

4. ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

5. Ноурем, Д. Firebase для Android. Разработка приложений. / Д. Ноурем. – СПб.: Питер, 20XX. – XXX с. (Пример книжного источника, заменить на актуальный, если использовался)

6. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – М.: Русская Редакция, 20XX. – XXX с.

7. Статьи и блоги по разработке мобильных приложений и UI/UX дизайну (например, Smashing Magazine, Habr – разделы по мобильной разработке).