Промежуточный отчет по программному проекту

1. Основные планы и этапы проекта

1.1 Краткое описание проекта:

Приложение для создания и обмена пешеходными маршрутами предоставит пользователям возможности для планирования прогулок и исследования новых мест.

Основная функция приложения — создание персонализированных маршрутов, где пользователи могут добавлять важные точки и сохранять маршрут в черновик для последующего редактирования. Кроме того, в приложении будет реализован поиск маршрутов с использованием фильтров и сортировки, что позволит находить маршруты, соответствующие предпочтениям пользователя. При прохождении маршрутов будет предусмотрена возможность ставить их на паузу и возвращаться к прогулке позже. Дополнительно будет реализована функция сохранения маршрутов в избранное, чтобы пользователи могли быстро вернуться к понравившимся вариантам.

В отличие от стандартных картографических сервисов, предлагающих маршруты для транспорта или спортивных приложений, ориентированных на поиск маршрутов для бега, наше приложение фокусируется именно на пеших маршрутах, позволяя пользователям создавать, сохранять и делиться маршрутами, что делает продукт инструментом для планирования прогулок.

Название проекта: Приложение для создания пешеходных маршрутов

Цель проекта: разработать приложение для создания пешеходных маршрутов

Краткое описание задач:

- Выбор инструментов для написания фронтенд-части приложения
- Выбор архитектуры для фронтенд-части приложения
- Перенос дизайна приложения из фигмы, вёрстка экранов
- Разработка логики взаимодействия с экранами
- Разработка бизнес-логики на фронтенд-части
- Интеграция внешних API (бэкенд + Яндекс Карты API) с фронтенд-частью
- Доработка, исправление ошибок, тестирование

1.2 Планы и этапы выполнения проекта:

Этап проекта	**Описание работ**	**Ожидаемые результаты**	**Сроки выполнения**
Выбор инструментов и архитектуры для разработки фронтенд-части приложения	Выбор инструментов для написания фронтенд-части приложения	Определён стек технологий	15.10.24 – 10.11.24
	Выбор архитектуры для фронтенд-части приложения	Выбрана архитектура	10.11.24 – 30.11.24
Разработка проекта	Перенос дизайна приложения из фигмы, вёрстка экранов	Созданы экраны, соответствующие дизайну	04.12.24 – 20.12.24
	Разработка логики взаимодействия с экранами	Обеспечена навигация и взаимодействие между экранами	20.12.24 – 15.01.25
	Разработка бизнес-логики на фронтенд-части	Реализованы основные функции приложения	15.01.25 – 31.01.25
	Интеграция внешних API (бэкенд + Яндекс Карты API) с фронтенд-частью	Настроено взаимодействие с API для работы маршрутов и отображения карт	15.01.25 – 31.01.25
Тестирование, исправление ошибок	Доработка, исправление ошибок, тестирование	Обеспечена стабильная работа фронтенд-части, исправлены выявленные баги, подготовлен релиз.	10.02.25 – 02.03.25

2. Используемый технологический стек и его обоснование

2.1 Перечень используемых технологий:

Технология/Инструмент	**Описание**	**Причины выбора**
Kotlin	Язык программирования для разработки Android-приложений	Лаконичный, безопасный, официально поддерживается Google, широко используемый

		язык для разработки Aydroid-приложений
Android Studio	Интегрированная среда разработки для Android	Поддержка Jetpack Compose, встроенные инструменты для отладки и тестирования, высокая производительность, удобство
Jetpack Compose	Фреймворк для создания интерфейсов на Android	Современный подход к разработке UI, удобная интеграция с Android Studio
Яндекс Карты API	API для работы с картами и геолокацией	Богатый функционал для работы с маршрутами, актуальные Российские карты
Git, GitHub	Система контроля версий и платформа для совместной работы	Удобство командной работы, контроль изменений
Figma	Программа для проектирования интерфейсов	Удобство работы над дизайном

2.2 Обоснование выбранного технологического стека:

Этот стек был выбран за его современность, удобство и соответствие задачам проекта. Kotlin и Jetpack Compose упрощают разработку и поддержку UI, Android Studio обеспечивает эффективную среду работы, а Яндекс Карты API предоставляет необходимый функционал для маршрутов. Git и GitHub удобны для командной работы, а Figma упрощает передачу дизайна. Вместе они ускоряют разработку и повышают качество приложения.

3. Критерии оценивания проекта

Критерий	**Описание**
Использование адаптивного дизайна	Будут использованы/Не будут использованы
Функциональность - Процент выполнения функциональных требований	Выполненные требования в процентах от общего количества
Функциональность - Количество реализованных функций	Абсолютное количество функций, которые работают правильно
Качество кода - Количество обнаруженных ошибок (bugs/KLOC)	Количество ошибок на 1000 строк кода
Соблюдение сроков и плана - Процент выполнения работы в срок (%)	Процент задач, выполненных в срок

Соблюдение сроков и плана - Количество дней отклонения от плана	Общее число дней отклонения от плана
Использование технологического стека - Процент использования функциональности стека (%)	Процент использования функциональности выбранного стека технологий
Оценка командной работы - Среднее время коммуникации (в часах)	Среднее время, потраченное на обсуждение задач и решение вопросов
Оценка командной работы - Количество завершенных задач на каждого участника	Общее число задач, выполненных каждым членом команды