

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: «Разработка прототипа мобильного Android-приложения для планирования путешествий»

Обучающийся: Тихонов Роман Александрович, К3139

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Описание проекта.....	5
2 Работа над проектом	9
2.1 Знакомство и предварительная подготовка	9
2.2 Разработка дизайна макетов приложения	9
2.3 Разработка frontend части приложения.....	9
2.4 Разработка backend части приложения	9
3 Мой вклад в реализацию проекта.....	11
4 Анализ моей работы.....	12
5 Взаимодействие с командой	14
6 Взаимодействие с руководителем проекта.....	15
7 Оценка работы руководителя	16
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	18
ПРИЛОЖЕНИЕ А.....	19

ВВЕДЕНИЕ

Современный мир стремительно движется к тотальной цифровизации, что кардинально меняет взаимодействие человека с окружающим миром и открывает возможности для автоматизации самых разных процессов, в том числе планирования отдыха и путешествий.

Цифровые технологии кардинально изменили подход к организации путешествий, предоставляя пользователям возможность планировать свои поездки с помощью различных мобильных приложений. В условиях стремительного роста туристической индустрии и увеличения числа пользователей смартфонов, разработка эффективного инструмента для планирования путешествий становится особенно актуальной. Не смотря на то, что существует множество мобильных приложений для путешественников, таких как Google Maps [1], TripAdvisor [2], которые уже зарекомендовали себя на рынке, большинство из них предлагают лишь частичные решения, не охватывающие все аспекты планирования путешествий, а также не могут работать без подключения к интернету.

С учетом данных недостатков, необходимость разработки нового приложения заключается в создании единого сервиса, содержащего функции поиска достопримечательностей, построения маршрутов и управления деталями поездки в одном интерфейсе. Уникальность данного приложения заключается в наличии оффлайн-доступа к информации о путешествии, что особенно важно в условиях роуминга за границей или отсутствия стабильного подключения к интернету в удаленных районах. Целью данной работы является проектирование прототипа Android-приложения для помощи в планировании путешествий. Исходя из поставленной цели, можно выделить следующие задачи:

- 1) описать требования к прототипу и составить техническое задание,
- 2) спроектировать макеты прототипа приложения,
- 3) спроектировать архитектуру приложения,

- 4) реализовать прототип согласно техническому заданию и макетам,
- 5) провести автоматизированное и ручное тестирование прототипа,
- 6) проанализировать и исправить выявленные в ходе тестирования недочеты,
- 7) подготовить доклад о проделанной работе с презентацией.

Проектирование и разработка мобильного приложения для удовлетворения любой пользовательской потребности – это сложный процесс, который включает в себя различные этапы:

- 1) исследование рынка на предмет наличия аналогов,
- 2) проектирование интерфейсов,
- 3) реализации функционала,
- 4) тестирования.

В рамках данной работы рассматривается разработка прототипа Android-приложения, которое объединяет инструменты построения путешествий: подбор удовлетворяющих пользователя авиабилетов и отелей для проживания в выбранные даты, а также поиска интересных достопримечательностей.

1 Описание проекта

Разрабатываемое мобильное приложение является инструментом для полноценного планирования путешествия: от выбора места до составления точного маршрута и списка достопримечательностей. Оно содержит следующие функции:

- 1) покупка авиабилетов,
- 2) бронирование отелей,
- 3) выбор активностей на время путешествия,
- 4) редактирование деталей маршрута,
- 5) сохранение данных о текущем путешествии.

В качестве архитектурного паттерна приложения был выбран подход MVP (Model, View, Presenter).

Паттерн MVP состоит из трех основных компонентов:

- 1) `model` – отвечает за управление данными приложения. Он взаимодействует с источниками данных, такими как базы данных или API, и предоставляет `Presenter` необходимую информацию,
- 2) `view` – интерфейс пользователя, который отображает данные и отправляет пользовательские действия в `Presenter`. `View` не содержит бизнес-логики и лишь отвечает за визуализацию информации,
- 3) `presenter` – служит связующим звеном между `Model` и `View`. Он обрабатывает пользовательские действия, запрашивает данные у `Model` и обновляет `View` с полученной информацией.

Этот паттерн был выбран из-за своей способности разделять логику приложения, что обеспечивает лучшую тестируемость и поддержку кода. Выбор паттерна MVP обусловлен несколькими факторами:

- 1) разделение ответственности: каждый компонент имеет свою четко определенную роль, что упрощает поддержку и развитие приложения,
- 2) тестируемость: логика приложения сосредоточена в `Presenter`, что облегчает тестирование отдельных компонентов,

3) гибкость: изменения в одном компоненте (например, изменение интерфейса) не требуют значительных изменений в других компонентах.

Реализация паттерна MVP отображенная при помощи нотации UML, а также его применение на примере выбора авиабилетов представлены на рисунке 1.

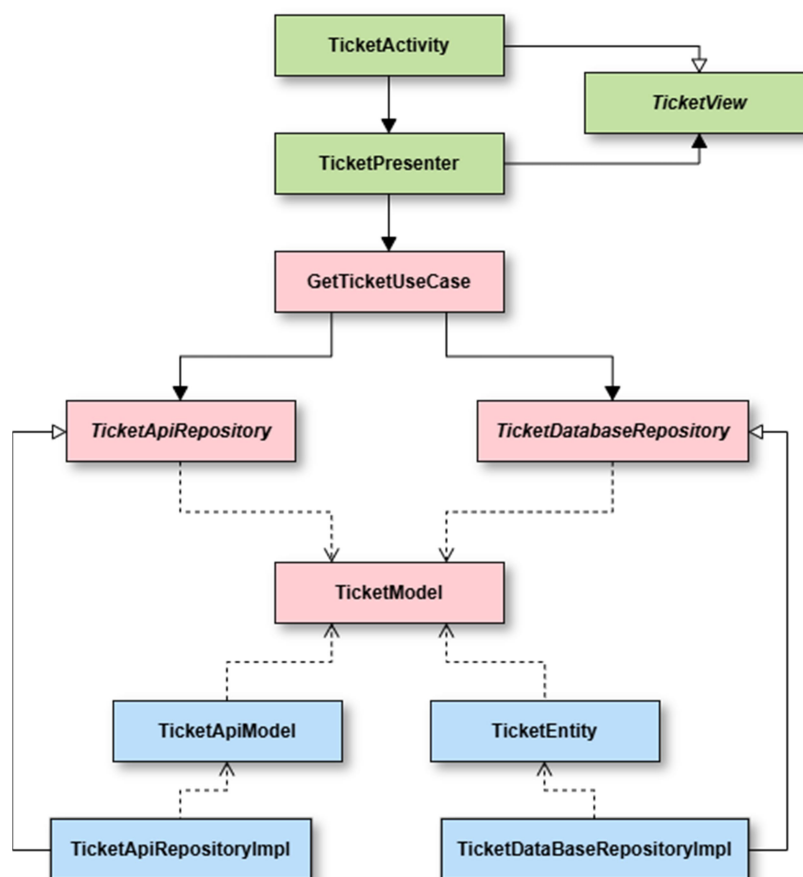


Рисунок 1 – Структура фрагмента выбора авиабилетов

Так как приложение разрабатывается для операционной системы android, то в качестве среды разработки было выбрано официальное решение от компании Google – Android Studio [3]. которая имеет большой набор инструментов для разработки, официальную документацию и множество обучающих материалов, помогающих улучшить свои навыки разработки в данной программе.

Android Studio поддерживает два языка программирования: Kotlin и Java. Для нашей задачи Kotlin является более предпочтительным из-за ряда преимуществ:

1) совместимость с Java: Kotlin полностью совместим с Java, что позволяет интегрировать существующий код и библиотеки без необходимости их переписывания,

2) лаконичность и читаемость кода: Код, написанный на Kotlin, как правило, короче и более понятен по сравнению с аналогичным кодом на Java. Это способствует снижению вероятности ошибок и упрощает дальнейшую поддержку и развитие проекта,

3) безопасность типов: Kotlin внедряет механизмы, направленные на предотвращение распространенной проблемы нулевых ссылок (NullPointerException). Система типов языка позволяет явно указывать, какие переменные могут принимать значение null, что повышает общую безопасность кода,

4) современные языковые конструкции: Kotlin поддерживает множество современных функций, таких как расширяющие функции, лямбда-выражения и корутины для асинхронного программирования. Это позволяет создавать более эффективный код, что положительно сказывается на оптимизации приложений,

5) официальная поддержка Google: Google признал Kotlin языком первого класса для разработки под Android, что означает, что новые инструменты и библиотеки будут разрабатываться с учетом этого языка. Это обеспечивает актуальность и поддержку Kotlin в операционной системе Android,

6) интеграция с Android Studio: Среда разработки Android Studio предлагает отличную поддержку Kotlin, включая функции автозаполнения кода, рефакторинга и отладки. Это делает процесс разработки эффективнее,

7) улучшенная производительность: Kotlin компилируется в байт-код JVM и может использоваться с существующими библиотеками Java, что обеспечивает высокую производительность приложений.

Таким образом, выбор языка программирования Kotlin обоснован его современными возможностями, безопасностью кода и поддержкой со

стороны Google, что делает его наиболее подходящим инструментом для разработки Android-приложений.

После изучения рекомендаций от компании Google, было принято решение о реализации архитектуры Clean Architecture [4], согласно которой проект разделен на три модуля:

- 1) `app` – основной модуль, который содержит реализацию интерфейса пользователя и связывает все компоненты,
- 2) `data` – модуль, отвечающий за работу с данными. Он включает реализацию репозитория и источники данных, обеспечивая доступ к необходимой информации,
- 3) `domain` – модуль, содержащий бизнес-логику приложения и юзкейсы. Этот модуль обеспечивает изоляцию бизнес-логики от деталей реализации.

Clean Architecture [4] помогает значительно улучшить качество кода и упростить процесс разработки, особенно в больших и сложных проектах. Схема данной архитектуры представлена на рисунке 2.

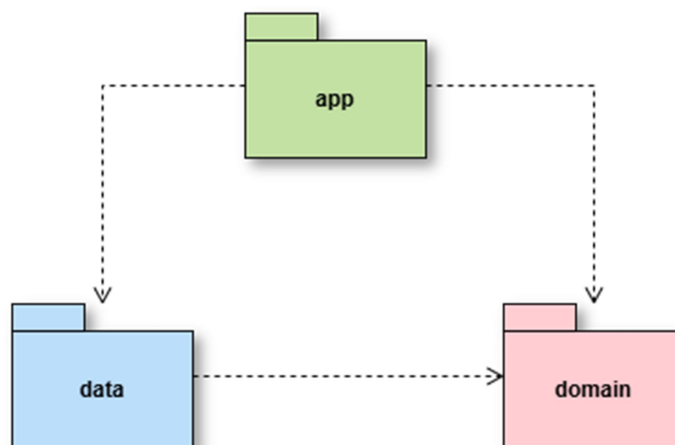


Рисунок 2 – Структура android-приложения

2 Работа над проектом

Наша команда состояла из 6 человек:

- 1) руководитель – Бердников Данил Игоревич,
- 2) три разработчика – Тихонов Роман, Артемов Илья и Говоров Павел,
- 3) UI/UX дизайнер – Мошкина Галина,
- 4) тестировщик – Левчук Софья.

2.1 Знакомство и предварительная подготовка

Работа над проектом началась со знакомства и составления плана разработки. Также руководителем были даны рекомендации по изучению определенного материала для увеличения эффективности и улучшения качества разработки.

2.2 Разработка дизайна макетов приложения

В первую очередь было необходимо создать дизайн макетов приложения. В сервисе Figma [5], который предоставляет онлайн доступ к макетам, дизайнер Галина создала удобный и интуитивно понятный интерфейс с учетом общепринятых стандартов.

2.3 Разработка frontend части приложения

После создания макетов началась frontend разработка, в ходе которой был сверстан пользовательский интерфейс согласно макетам. Затем была проведена работа над адаптивностью интерфейса с целью корректного отображения всех элементов на устройствах с различным разрешением и соотношением сторон экрана. Затем была написана логика для клиентской части (реализация календаря для выбора даты, переход между экранами, приветственный экран, обработка нажатий кнопки, проверка введенных значений и т.д.).

2.4 Разработка backend части приложения

Когда клиентская часть была в стадии завершения, наша команда приступила к backend разработке. В ходе данного этапа необходимо было решить следующие задачи:

- 1) подключение локальной базы данных для хранения файлов, которые могут располагаться непосредственно на устройстве пользователя с целью экономии трафика и ускорения загрузки приложения,

- 2) реализация сетевых запросов к API для получения данных о авиабилетах, бронировании отелей, поиска достопримечательностей и т.д.

Для работы с базами данных использовалась библиотека Room, а для запросов к сторонним API – Retrofit.

После завершения основного этапа разработки, необходимо было протестировать приложение. Для тестирования были использованы следующие библиотеки:

- 1) JUnit – библиотека для тестирования отдельно взятых частей кода (юнит-тестирования),

- 2) Espresso – библиотека для тестирования пользовательского интерфейса (UI-тестирования).

После того как наш тестировщик – Левчук Софья подготовила отчет с выявленными недочетами, команда приступила к их исправлению.

Последним этапом работы стала подготовка защиты проекта, в которую входят:

- 1) разработка плана защиты,
- 2) подготовка презентации, соответствующей плану защиты проекта,

- 3) итоговая репетиция, на которой должны быть выявлены и исправлены недочеты, отработаны ответы на возможный список вопросов.

3 Мой вклад в реализацию проекта

В нашей команде я являюсь одним из разработчиков. Мною были реализованы несколько экранов приложения и их связь с backend частью. Передо мной были поставлены следующие задачи:

- 1) изменить название и иконку приложения в соответствии с макетом,
- 2) реализовать загрузочный экран, который появляется во время запуска приложения,
- 3) реализовать проверку, является ли запуск приложения первым с момента его установки, и сохранять соответствующую информацию,
- 4) реализовать экран списка отелей, на котором пользователь сможет выбрать дату заезда и отъезда, используя описание подобрать наиболее подходящий отель из списка доступных на выбранные даты, выбрать необходимый для проживания номер из имеющихся,
- 5) реализовать экран с отображением календаря событий, на котором пользователю будет предоставляться возможность выбора новых событий и просмотра выбранных ранее активностей,
- 6) реализовать экран рекомендаций активностей к путешествию согласно спроектированным макетам. Данный экран помогает пользователю подобрать наиболее привлекательные мероприятия согласно его вкусам.

Кроме самого создания экранов, я занимался реализацией логической связи и сочетанием в единое приложение моих экранов и экранов, созданных другими разработчиками. Для экрана выбора авиабилетов и экрана главного меню также потребовалось реализовать обработку и информативное отображение в пользовательском интерфейсе данных, полученных с API запросов и баз данных соответственно.

В ходе выполнения поставленных задач, весь код я отправлял на удалённый сервер GitHub [6] для проверки руководителем.

4 Анализ моей работы

В результате работы были выполнены почти все задачи. Из-за ограниченных сроков нашей командой было принято решение отказаться от экрана рекомендаций активностей к путешествию. Причин для отказа от данной идеи было несколько:

- 1) отсутствие качественного API,
- 2) высокая загруженность члена команды, отвечающего за backend часть данного экрана,
- 3) временные затраты на устранение непредвиденных трудностей и реализацию небольших подзадач возникших в ходе разработки,
- 4) незначительная важность. Данный экран был одним из самых наименее значимых для реализации основной идеи приложения.

Из-за совокупного влияния данных факторов с целью сохранения качества остальной части разрабатываемого приложения было принято решение не реализовывать данный экран.

В ходе разработки пришлось изучить большое количество новой информации:

- 1) XML верстка,
- 2) программирование на языке Kotlin,
- 3) архитектурный паттерн MVP и его реализация в проекте,
- 4) циклы жизни различных классов Android-приложения,
- 5) новые библиотеки для языка Kotlin.

Также было непривычно работать в полноценной команде, но благодаря руководителю, коллектив работал слажено.

Во время разработки некоторых экранов не раз возникали трудности из-за недостатка опыта или незнания некоторых библиотек. В таких ситуациях мне помогал руководитель или же члены команды, к которым тоже всегда можно было обратиться за помощью.

Наличие четкого плана работы (с учетом времени на исправление непредвиденных трудностей) и разделение задач обеспечили равномерную

работу над проектом, избавив команду от доделывания проекта бессонными ночами перед защитой. Доска с задачами, разделенными на группы, в сервисе itmo-learning-projects позволяла грамотно распределять время. Несмотря на то, что периодически приходилось затрачивать больше времени на решение возникших ошибок и исправление указанных руководителем недочетов в выполненной работе, я всегда понимал объем сделанной и предстоящей работы, что способствовало грамотному управлению временными ресурсами.

За время работы над проектом мною было получено и отработано большое количество навыков:

- 1) разработка многомодульного Android проекта,
- 2) программирование на языке Kotlin в среде разработки Android Studio,
- 3) применение архитектурного паттерна MVP,
- 4) опыт командной разработки над проектом с использованием системы контроля версий Git [7] и удаленного сервера на Github [6],
- 5) создание адаптивного интерфейса с использованием XML разметки,
- 6) реализация экранов по готовым макетам.

Полученный опыт определенно пригодится в дальнейшей образовательной и профессиональной деятельности.

5 Взаимодействие с командой

Коммуникация с участниками и руководителем команды происходила в групповом чате мессенджера Telegram [8], в котором публиковались рекомендованные руководителем источники информации, обсуждалась разработка и тестирование, утверждались даты групповых звонков для обсуждения текущего прогресса.

Один раз в неделю проводились групповые звонки в сервисе Google Meet [9] для подведения итогов выполнения задачи каждого члена команды, решения возникших в ходе разработки проблем, а также для получения новых задач на неделю. Также руководитель отвечал на вопросы и давал комментарии для каждой задачи при необходимости.

Руководитель проекта дополнительно оставлял комментарии для дизайнера в сервисе Figma [5], а для разработчиков на странице GitHub [6]

6 Взаимодействие с руководителем проекта

Взаимодействие с руководителем происходило в основном в групповых чатах и звонках, однако при необходимости можно было отправить свой вопрос в личных сообщениях. Также нередко назначались личные звонки для более качественного решения возникших трудностей.

Кроме назначения задач и помощи в решении проблем руководитель всегда оставлял дополнительные комментарии или обучающие материалы по решению каждой задачи, комментировал написанный разработчиками код, давал советы по его оптимизации, так как имел значительно больше опыта в разработке приложений для операционной системы Android.

Несмотря на то, что у многих членов команды сильно отличалось расписание учебного процесса, руководитель всегда старался подобрать наиболее подходящее всем участникам проекта время с целью минимизации влияния работы над проектом на учебный процесс и создания комфортной рабочей атмосферы в коллективе. О назначенных датах различных событий от руководителя всегда приходило своевременное оповещение и последующее напоминание.

7 Оценка работы руководителя

Уже до начала разработки проекта Бердников Данил проявил лидерские качества, умение организовать команду и стратегически планировать все этапы работы, начиная от составления технического задания до планирования стратегии защиты проекта.

Даниил Игоревич всегда демонстрировал высокий уровень ответственности и профессионализма в сфере разработки Android-приложений. Его выстраивание эффективной коммуникации с командой позволила быстро решать возникающие вопросы. Также он на протяжении всей работы над проектом старался мотивировать команду и создавать позитивную атмосферу, что способствовало повышению продуктивности и сплоченности коллектива.

Особо хочется отметить выдержку руководителя, благодаря которой он мог несколько часов давать уточняющие комментарии даже по небольшой задаче, помогать с решением любых проблем.

Таким образом, благодаря усилиям и руководству Даниила Игоревича наша команда работала крайне эффективно и выполнила большой объем работы с учетом сроков проекта.

ЗАКЛЮЧЕНИЕ

Таким образом, основная цель команды – разработать прототип Android-приложения для планирования путешествий была достигнута. Несмотря на то, что не все поставленные задачи были выполнены, все члены команды получили опыт в самых разных областях в сфере разработки мобильных приложений. Результаты проделанной работы подтверждают, что наше приложение способствует упрощению планирования маршрутов путешествий. В перспективе возможно усовершенствование данного прототипа до полнофункционального продукта, включая оптимизацию производительности и расширение функционала. Проект продемонстрировал важность и актуальность разработки мобильных приложений для полноценного планирования путешествий, а также подтвердил возможность успешной реализации проектов такого типа.

Для меня, как и для большинства членов команды, это был первый опыт командой разработки, однако под чутким руководством Бердникова Данила, мы работали не каждый по себе, а как единое целое. Это стало основополагающим фактором для последующей эффективной работы и получения практически значимого опыта разработки мобильных приложений, так как при работе в команде даже знакомые ранее технические решения реализуются иначе. Особенно выделяется опыт работы с системой контроля версий, когда работу ведут сразу несколько разработчиков.

Выражаю благодарность своему руководителю и всем членам нашей команды за получение опыта командной разработки и приобретение практически значимых навыков в программировании, которые несомненно пригодятся в дальнейшей профессиональной деятельности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Google Map – URL: <https://www.google.com.mm/maps>
2. TripAdvisor – URL: <https://www.tripadvisor.ru/>
3. Android Studio – URL: <https://developer.android.com/>
4. Clean Architecture – URL: https://reconshell.com/wp-content/uploads/2022/06/Clean_Android_Architecture.pdf
5. Figma – URL: <https://www.figma.com/>
6. GitHub – URL: <https://github.com/>
7. Git – URL: <https://git-scm.com/>
8. Telegram – URL: <https://telegram.org/>
9. Google Meet – URL: <https://meet.google.com/>
10. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

ПРИЛОЖЕНИЕ А

Техническое задание к проекту

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Название проекта:

Разработка прототипа мобильного Android-приложения для планирования путешествий.

2. Цель (назначение)

Спроектировать, реализовать и провести ручное тестирование прототипа Android-приложения для помощи путешественникам в планировании путешествий.

3. Сроки выполнения

Начало: 01 ноября 2024 г.

Окончание: 20 декабря 2024 г.

4. Руководитель проекта

Бердников Данил Игоревич

5. Термины и сокращения

Активность – Организованное событие, которое может включать концерты, выставки, фестивали, лекции и другие виды мероприятий, а также посещение достопримечательностей и кафе, ресторанов.

Путешествие – Запланированный список активностей в рамках поездки, а также информация об авиабилетах и отелях.

Рекомендации – Предложенные мероприятия или места, подобранные на основе дат путешествия пользователя.

Оффлайн-режим – Режим работы приложения без подключения к интернету, когда доступна только сохраненная информация.

Календарь событий – Функционал для отображения запланированных и рекомендованных мероприятий по датам.

API – Application Programming Interface, интерфейс для интеграции с внешними сервисами.

CRUD – Create, Read, Update, Delete, базовые операции для работы с данными пользователя и событий.

6. Технические требования

Нефункциональные требования:

- использование Figma для проектирования макетов приложения,
- поддерживаемая ОС Android версии 8.0 и выше,
- использование IDE Android Studio версии "Koala" и выше,
- фиксирование процесса разработки при помощи GitHub,
- должно производиться кэширование информации о путешествии на устройстве пользователя,
- использование открытых API для получения информации об авиабилетах, отелях и активностях.
- данные о путешествии пользователя должны храниться на облачном хранилище.

Функциональные требования:

- у пользователей должна быть возможность авторизации и регистрации в системе,
- приложение должно производить подбор авиабилетов по заданной дате и аэропортам,
- пользователь должен иметь возможность сортировать список подобранных авиабилетов по цене и времени в пути,
- при первом запуске приложения должны отображаться приветственные экраны, коротко описывающие приложение,
- приложение должно осуществлять CRUD-операции с путешествиями,
- приложение должно предлагать активности на период путешествия,
- у пользователя должна быть возможность сортировки списка активностей по дате и цене,
- приложение должно предлагать выбор категории активности: мероприятие, еда, достопримечательности,

- приложение должно осуществлять CRUD-операции с информацией о пользователе,
- приложение должно осуществлять подбор доступных свободных номеров в отелях на период путешествия,
- пользователь должен иметь возможность сортировать список подобранных номеров в отелях по цене,
- пользователям с путешествием должны рекомендоваться другие активности, совпадающие с датой путешествия, с возможностью их добавления в путешествие,
- планирование путешествия в приложении должно производиться поэтапно: сначала подбор авиабилетов, затем выбор отеля и номера (либо отказ от бронирования жилья), последним - выбор активностей,
- у пользователя должен отображаться календарь событий, описывающий его активности на выбранный период времени,
- у пользователей должна быть возможность включать или выключать напоминания о путешествии и активностях.

7. Содержание работы

Таблица А.1 – Этапы проекта

№	Этапы проекта	Сроки выполнения этапов	Ответственный за этап	Вид представления результатов этапа
1	Разработка технического задания (ТЗ)	1-10 ноября	Бердников Д.И.	Текстовый документ с описанием необходимых пунктов ТЗ: название проекта, сроки выполнения, руководитель, термины, требования, содержание работ и результат.
2	Проектирование макетов прототипа приложения	1-26 ноября	Мошкина Г.А.	Готовые макеты приложения в сервисе Figma.

Продолжение таблицы А.1

3	Реализация архитектуры	1-14 ноября	Артемов И.В.	Реализованные модули системы и основные
---	------------------------	-------------	--------------	---

	прототипа приложения			компоненты архитектуры проекта, а также диаграмма модулей системы и диаграмма классов, отражающая структуру компонентов архитектурного подхода, диаграммы должны быть выполнены при помощи нотации UML.
4	Реализация функциональности прототипа приложения	14 ноября – 6 декабря	Говоров П.И.	Реализованный прототип приложения, соответствующий требованиям, выбранной архитектуре и спроектированным макетам.
5	Тестирование прототипа приложения	21 ноября – 16 декабря	Левчук С.А.	Описанные тест-кейсы и код автоматизированного тестирования.
6	Исправление недочетов прототипа приложения, выявленных после тестирования	30 ноября – 18 декабря	Артемов И.В.	Исправленный согласно описанным тест-кейсам прототип приложения.
7	Защита проекта (сдача отчета и представление доклада с презентацией)	18-20 декабря	Мошкина Г.А.	Презентация, описывающая продукт и проделанную работу, а также электронный письменный отчет.

8. Основные результаты работы и формы их представления

Результатом работы является прототип Android-приложения для планирования путешествий с соответствующим принципам построения Android интерфейсом. Весь набор функциональности приложения должен исправно работать. Спроектированный интерфейс приложения должен удовлетворять стандартам Material Design. Приложение должно позволять пользователям регистрироваться и авторизоваться в системе; осуществлять предложение авиабилетов, отелей и активностей, помогать составлять и изменять путешествие; предлагать рекомендации по путешествию, а также отображать детали путешествия, в том числе в оффлайн-режиме.

9. Задачи

Таблица А.2 – Задачи проекта

№ этапа	Название задачи	Описание	Ответственный	Сроки выполнения
1	Подготовить техническое задание для проекта	Необходимо подготовить техническое задание (далее - ТЗ) для текущего проекта. В ТЗ необходимо отразить: - название проекта, - цели проекта, - сроки выполнения, - руководителя проекта, - используемые в проекте термины и сокращения, - требования к системе (функциональные, нефункциональные, требования к системе в целом, требования к видам обеспечения), - содержание работы (этапы проекта, задачи для каждого этапа), - результаты работы. Данное ТЗ необходимо внести в систему Odoo в раздел ТЗ текущего проекта.	Бердников Д.И.	01.11.2024 – 04.11.2024
2	Спроектировать макеты приветственных экранов	При первом запуске приложения необходимо показать приветственные (on boarding) экраны с кратким описанием, для чего нужно приложение. Необходимо спроектировать макеты данных экранов. На данных экранах кратко и поэтапно описать особенности работы приложения, какие этапы составления маршрута существуют. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	06.11.2024 – 09.11.2024
2	Изучить требования к UI Android-приложений	Для проектирования макетов приложения необходимо изучить особенности элементов пользовательского интерфейса Android-приложений, в частности, стандарты Material Design. Необходимо обратиться к официальному сайту и прочитать информацию о принципах, которые используются в Material Design, а также попробовать использовать предлагаемые на официальном сайте готовые элементы UI в Figma. О результатах изучения отчитаться на онлайн-созвоне.	Мошкина Г.А.	01.11.2024 – 05.11.2024

Продолжение таблицы А.2

2	Спроектировать макеты экранов со списками	Необходимо спроектировать макеты экранов приложения со списками данных для настройки маршрута - список авиабилетов, отелей, активностей (достопримечательностей, мероприятий, кафе). Экраны могут выглядеть схоже, однако сам внешний вид элемента может отличаться в зависимости от сущности. Также необходимо использовать шаблонные элементы UI в Figma, предоставляемые разработчиками принципов Material Design. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	09.11.2024 – 13.11.2024
2	Спроектировать макеты экранов информации об элементе списка	Необходимо спроектировать макеты экранов приложения информацией об элементе списка - отели, активностей (достопримечательностей, мероприятий, кафе). Данные экраны будут отображаться после того, как пользователь выбрал элемент из списка. Информация может отображаться как в виде выпадающего списка, так и переходом на новый экран. Также необходимо использовать шаблонные элементы UI в Figma, предоставляемые разработчиками принципов Material Design. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	13.11.2024 – 15.11.2024
2	Спроектировать макет экрана настройки путешествия	Необходимо спроектировать экраны, отвечающие за бизнес-логику изменения деталей путешествия - настройку авиабилетов, отеля или активностей. Необходимо продумать логику отображения и последовательность процесса, можно использовать реализованные ранее макеты списков элементов. Также необходимо использовать шаблонные элементы UI в Figma, предоставляемые разработчиками принципов Material Design. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	15.11.2024 – 19.11.2024
2	Спроектировать макет экрана рекомендаций путешествия	Необходимо спроектировать экран рекомендаций активностей. Данный экран может быть главной страницей приложения, где находятся рекомендуемые к добавлению активности, либо отображаться на экране конкретного маршрута. Также необходимо использовать шаблонные элементы UI в Figma, предоставляемые разработчиками принципов Material Design. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	21.11.2024 – 24.11.2024

Продолжение таблицы А.2

2	Спроектировать макет экрана календаря событий	Пользователю должна быть доступна информация о предстоящих мероприятиях (активностях). Для этого необходимо спроектировать экран календаря событий. Данный экран может быть отображен как после изучения деталей конкретного маршрута, так и быть отдельным экраном приложения, где содержатся все активности всех маршрутов - на усмотрение исполнителя. Также необходимо использовать шаблонные элементы UI в Figma, предоставляемые разработчиками принципов Material Design. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	19.11.2024 – 21.11.2024
2	Спроектировать макет push-уведомления	Необходимо спроектировать макеты push-уведомлений о путешествии - напоминания о путешествии. Также необходимо использовать шаблонные элементы UI в Figma, предоставляемые разработчиками принципов Material Design. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	24.11.2024 – 26.11.2024
2	Спроектировать экраны авторизации и регистрации в приложение	Для работы в приложении и сохранении маршрутов необходимо осуществить регистрацию или авторизацию пользователя в системе. Необходимо спроектировать макеты экранов регистрации и авторизации. Данные процессы осуществляются при помощи почты пользователя и пароля. При регистрации необходимо продублировать ввод пароля. После проектирования будет проведено ревью и даны комментарии в системе Figma.	Мошкина Г.А.	12.11.2024 – 14.11.2024
3	Изучить архитектурные подходы	Для реализации приложения необходимо выбрать архитектурный подход. Рекомендуемые подходы для изучения - MVC, MVP, MVVM. Рекомендуемый к использованию - MVP. Необходимо отчитаться об изученных подходах и выбранной архитектуре на последующем онлайн-созвоне.	Артемов И.В.	01.11.2024 – 06.11.2024
3	Спроектировать структуру модулей приложения	Необходимо спроектировать структуру модулей проекта. При проектировании стоит опираться на рекомендации Google. Согласно им, необходимо реализовать три модуля - app, domain и data. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Артемов И.В.	07.11.2024 – 11.11.2024

Продолжение таблицы А.2

3	Спроектировать архитектуру приложения	Необходимо спроектировать архитектуру приложения на основе выбранного исполнителем подхода. Для этого, необходимо реализовать необходимые программные классы и реализовать необходимую структуру проекта, соответствующую выбранному архитектурному подходу. Прodelанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прodelанной работы в основную ветку репозитория.	Артемов И.В.	11.11.2024 – 14.11.2024
4	Реализовать экран профиля пользователя	Согласно спроектированным макетам необходимо реализовать экран профиля пользователя, а также возможность редактирования данных - логина и пароля. Прodelанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прodelанной работы в основную ветку репозитория.	Говоров П.И.	14.11.2024 – 17.11.2024
4	Реализовать отображение приветственных экранов	Согласно спроектированным макетам необходимо реализовать приветственные экраны приложения, которые должны отображаться при первом запуске приложения, то есть, система должна запоминать, что первый запуск приложения уже был произведен. Для сохранения данных о запуске приложения следует использовать Shared Preferences. Прodelанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прodelанной работы в основную ветку репозитория.	Тихонов Р.А.	13.11.2024 – 17.11.2024
4	Реализовать экран списка отелей	Для создания поездки необходимо реализовать список отелей с краткой информацией согласно спроектированным макетам. Данные необходимо получать от серверного API. При подборе отелей необходимо учитывать выбранные пользователем на предыдущем этапе даты поездки и отображать только актуальные отели со свободными номерами. Прodelанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прodelанной работы в основную ветку репозитория.	Говоров П.И.	16.11.2024 – 19.11.2024

Продолжение таблицы А.2

4	Реализовать экран списка авиабилетов	Для создания маршрута путешествия необходимо реализовать экран выбора авиабилетов согласно спроектированным макетам. Данные необходимо получать от серверного API, отображать данные с учетом введенных пользователем данных о датах и пунктах назначения. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Говоров П.И.	15.11.2024 – 18.11.2024
4	Реализовать экран списка активностей	Необходимо реализовать экран списка активностей с возможностью выбора интересующего вида активности согласно спроектированным макетам. Данные следует получать от серверного API. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Говоров П.И.	18.11.2024 – 21.11.2024
4	Реализовать экраны деталей активностей	Необходимо реализовать отображение информации о выбранной путем нажатия из списка активности согласно спроектированным макет. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Говоров П.И.	19.11.2024 – 22.11.2024
4	Реализовать экран информации об отеле	Необходимо реализовать отображение информации об отеле по нажатию на интересующий отель из списка согласно спроектированным макетам. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Говоров П.И.	16.11.2024 – 20.11.2024
4	Реализовать экран информации о путешествии	Необходимо реализовать экран отображения информации о путешествии по нажатию на него в соответствии со спроектированными макетами. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Артемов И.В.	19.11.2024 – 22.11.2024

Продолжение таблицы А.2

4	Реализовать бизнес-логику создания путешествия	Необходимо настроить логику отображения экранов настройки маршрута и реализовать процесс создания маршрута. Для этого необходимо использовать реализованные ранее экраны выбора авиабилетов, отеля и активностей. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Артемов И.В.	22.11.2024 – 26.11.2024
4	Реализовать настройку путешествия	Необходимо реализовать возможность редактирования путешествия. Необходимая логика отображения находится в макетах приложения. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Артемов И.В.	29.11.2024 – 02.12.2024
4	Реализовать отображение уведомлений-напоминаний о путешествии	Необходимо реализовать отображение push-уведомлений для напоминания о путешествии на предстоящие даты. Вид уведомления должен соответствовать спроектированным макетам приложения. Для уведомлений необходимо использовать класс Service. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Тихонов Р.А.	27.11.2024 – 30.11.2024
4	Реализовать кэширование деталей путешествия	Приложение должно содержать оффлайн-режим, то есть, отображать детали путешествия без доступа к сети Интернет. В оффлайн-режиме доступно только отображение деталей ранее запланированных путешествий, на других экранах должны отображаться заглушки об отсутствии подключения согласно спроектированным макетам. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Артемов И.В.	26.11.2024 – 29.11.2024

Продолжение таблицы А.2

4	Реализовать экран с отображением календаря событий	Необходимо реализовать отображение календаря событий, который показывает мероприятия предстоящих путешествий, согласно спроектированным макетам. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление проделанной работы в основную ветку репозитория.	Тихонов Р.А.	22.11.2024 – 26.11.2024
4	Реализовать экран подбора рекомендаций к путешествию	Для возможности пользователем корректировать и узнавать о новых путешествиях в даты поездки необходимо реализовать экран рекомендаций активностей к путешествию согласно спроектированным макетам. Прделанную работу необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление проделанной работы в основную ветку репозитория.	Тихонов Р.А.	02.12.2024 – 06.12.2024
5	Провести тестирование отображения приветственных экранов	Необходимо убедиться, что ранее реализованные приветственные экраны отображаются лишь при первом запуске приложения. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	19.11.2024 – 21.11.2024
5	Провести тестирование редактирования профиля пользователя	Необходимо провести UI-тестирования реализованных ранее экранов профиля пользователя, а также провести ручное тестирование для проверки возможности редактировать информацию о пользователе. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа. Прделанную работу по написанию автоматизированного теста необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление проделанной работы в основную ветку репозитория.	Левчук С.А.	21.11.2024 – 23.11.2024

Продолжение таблицы А.2

5	Провести Unit-тестирование отображения списков отелей, авиабилетов и активностей	Необходимо описать Unit-тесты для проверки отображения списков авиабилетов, отелей и активностей на реализованных ранее экранах. Необходимо убедиться, что отображаемые данные соответствуют полученным приложением. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа. Прделанную работу по автоматизированному тестированию необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление прделанной работы в основную ветку репозитория.	Левчук С.А.	22.11.2024 – 25.11.2024
5	Провести тестирование создания путешествия	Необходимо проверить бизнес-процесс создания маршрута. Для этого, следует провести ручное тестирование и создать маршрут, а также проверить, что маршрут сохранился в системе. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	02.12.2024 – 05.12.2024
5	Провести тестирование изменения путешествия	Для корректной настройки маршрута необходимо провести тестирования изменения путешествия. Для этого необходимо создать путешествие, используя соответствующие ранее реализованные экраны приложения, а после изменить его детали и проверить, что система сохраняет полученные изменения. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	06.12.2024 – 10.12.2024
5	Провести UI-тестирование календаря событий	Необходимо проверить корректность отображения календаря событий для заранее заданных маршрутов. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	26.11.2024 – 28.11.2024

Продолжение таблицы А.2

5	Провести тестирование уведомлений	Необходимо провести тестирование уведомлений приложения. Для этого, следует спланировать маршрут на ближайшее время, используя соответствующие ранее реализованные экраны приложения, и отметить наличие или отсутствие уведомления (за сутки). В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	01.12.2024 – 04.12.2024
5	Провести тестирование рекомендаций путешествия	Для данного тестирования необходимо заранее составить путешествие, используя соответствующие ранее реализованные экраны приложения, после чего проверить, что приложение предлагает активности, которые будут проходить в даты данного путешествия. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	06.12.2024 – 09.12.2024
5	Провести тестирование оффлайн-режима приложения	Необходимо провести тестирование оффлайн-режима приложения. Для этого, необходимо создать маршрут, используя соответствующие ранее реализованные экраны приложения, и перезапустить приложение без доступа к сети Интернет, и убедиться в возможности просмотра деталей сохраненных ранее маршрутов. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	09.12.2024 – 11.12.2024
5	Провести UI-тестирование приложения на устройствах с разной диагональю	Необходимо провести тестирование реализованного пользовательского интерфейса на корректность отображения на устройствах с маленькой и большой диагональю. Для маленького устройства - диагональ размером 5-6, для большого - диагонали планшетов. Интерфейс приложения должен соответствовать спроектированным экранам, а также должен быть доступ ко всем функциям приложения. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	14.12.2024 – 16.12.2024

Продолжение таблицы А.2

5	Провести тестирование приложения на разных версиях ОС	Необходимо провести ручное тестирование приложения на старых и новых версиях ОС - на 8.0 и 14 для проверки исправности работы всех функций приложения, а также соответствия спроектированным макетам отображаемого интерфейса. Для этого можно использовать эмуляторы или физические устройства. В случае обнаружения недочета, необходимо описать test case по пунктам "предусловия", "шаги выполнения", "ожидаемый результат", "полученный результат" и представить данные недочеты в виде текстового документа.	Левчук С.А.	13.12.2024 – 16.12.2024
6	Исправить полученные в ходе тестирования недочеты	После проведения тестирования необходимо устранить выявленные недочеты приложения. Прделанную работу по каждому недочету необходимо оформить в виде отдельной ветки в Git-репозитории проекта и создать pull request, после чего будет проведено код-ревью и добавление проделанной работы в основную ветку репозитория.	Артемов И.В.	30.11.2024 – 17.12.2024
7	Подготовить презентацию для демонстрации проекта	Для защиты проекта необходимо подготовить презентацию, которая демонстрирует работу разрабатываемого прототипа приложения. В презентации необходимо отразить название проекта, состав команды, цели и задачи проекта, а также описать проделанную работу с демонстрацией экранов прототипа приложения (скриншоты, записи экрана и тд), а также полученные в ходе выполнения проекта результаты. Выполненную презентацию необходимо представить руководителю проекта на онлайн-созвоне, а также провести репетицию выступления для защиты проекта.	Мошкина Г.А.	14.18.2024 – 17.12.2024
7	Подготовить .apk файл прототипа приложения	Для демонстрации работы разработанного прототипа приложения необходимо подготовить установочный .apk-файл. Выполняется данная задача при помощи встроенных инструментов Android Studio. Установочный файл необходимо подготовить к онлайн-созвону и продемонстрировать его установку и работу приложения вместе с репетицией защиты проекта.	Артемов И.В.	16.11.2024 – 18.11.2024
7	Выступить на защите проекта	Необходимо представить проделанную работу и подготовленную презентацию с демонстрацией работы прототипа приложения на итоговой защите.	Мошкина Г.А.	18.11.2024 – 20.11.2024