

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

**Факультет      инфокоммуникационных технологий**

**Образовательная программа «Мобильные и сетевые технологии»**

**Направление подготовки 09.03.03 Прикладная информатика**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Инфокоммуникационные системы и технологии»**

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

**Обучающийся: Артемов Илья, группа К3141**

**Санкт-Петербург  
2025**

## СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ В ПРОЕКТ .....	5
2 ПРОЦЕСС РАБОТЫ НАД ПРОЕКТОМ.....	6
2.1 Планирование .....	6
2.2 Разработка.....	7
2.3 Тестирование .....	7
2.4 Защита .....	8
3 СУТЬ ПОСТАВЛЕННЫХ ЗАДАЧ.....	9
4 РЕШЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ.....	11
4.1 Проектировка архитектуры приложения .....	11
4.2 Подключение локальной базы данных.....	13
4.3 Отбор API, которые будут покрывать логику приложения. Интеграция выбранных API в приложение .....	13
4.4 Реализация экрана списка активностей.....	15
5 АНАЛИЗ ПРОДЕЛАННОЙ МНОЮ РАБОТЫ .....	16
6 ВЗАИМОДЕЙСТВИЕ С КОМАНДОЙ И РУКОВОДИТЕЛЕМ ПРОЕКТА	17
7 ОЦЕНКА РАБОТЫ РУКОВОДИТЕЛЯ .....	18

## ВВЕДЕНИЕ

Современный мир развивается в условиях цифровизации, что открывает новые возможности для автоматизации повседневных процессов. Одной из таких областей является планирование путешествий, где мобильные приложения играют важную роль, предоставляя пользователям удобный инструмент для организации маршрутов, поиска достопримечательностей и планирования путешествий. Популярность таких решений обусловлена их способностью экономить время на планирование путешествий и подбор мест для посещения, повышать комфорт процесса планирования и упрощать доступ к информации о путешествии. Особенно актуальным является наличие информации о путешествии в быстром доступе, поскольку необходимость проверить какие-либо данные о путешествии может возникнуть в любой момент. Таким образом, разработка мобильного приложения с таким набором функций является актуальной задачей.

Существует множество мобильных приложений для путешественников, таких как Google Maps [1], TripAdvisor [2] и Sygic Travel [3]. Эти сервисы предлагают широкий функционал: от прокладывания маршрутов до поиска популярных достопримечательностей и ресторанов. Однако большинство аналогов сосредоточены на решении отдельных задач, таких как карты, организация поездок или сбор отзывов, и не предоставляют комплексного подхода к путешествиям в пределах конкретного города. Например, Google Maps и TripAdvisor обладают мощными инструментами поиска мест, но не позволяют полностью управлять процессом планирования путешествий и хранить всю информацию о нем в одном месте. Sygic Travel и Visit a City [4] предлагают готовые маршруты, однако требуют премиум-доступа для расширенных функций и часто ограничены стандартными туристическими направлениями.

Учитывая эти недостатки, актуальность разработки нового приложения заключается в создании единой платформы, которая сочетает в себе поиск

достопримечательностей, построение маршрута и деталей путешествия, а также локализацию данных о путешествии в одном месте. Уникальной особенностью приложения является оффлайн-доступ к деталям путешествия и локализация всей информации в одном приложении, что делает его особенно актуальным для путешественников.

Проектирование и разработка мобильного приложения для планирования путешествий – это сложный процесс, который включает этапы исследования, проектирования интерфейсов, реализации функционала и тестирования. В рамках данной работы рассматривается создание прототипа Android-приложения, которое объединяет возможности построения путешествий путем подбора актуальных авиабилетов и отелей для проживания в выбранные даты, а также поиска достопримечательностей.

Целью данной работы является проектирование прототипа Android-приложения для помощи путешественникам в планировании путешествий. На основании поставленной цели можно выделить следующие задачи, реализуемые в рамках данной работы:

- 1) описать требования к прототипу и составить техническое задание,
- 2) спроектировать макеты прототипа приложения,
- 3) спроектировать архитектуру приложения,
- 4) реализовать прототип согласно техническому заданию,
- 5) провести ручное и автоматизированное тестирование прототипа,
- 6) анализ полученных в ходе тестирования недочетов,
- 7) подготовить доклад о проделанной работе с презентацией.

## **1 ВВЕДЕНИЕ В ПРОЕКТ**

Основная задача, которая стояла перед нашей командой – разработка прототипа Android-приложения, предназначенного для помощи путешественникам в организации и планировании их поездок. Эта цель определялась необходимостью создания инструмента, который бы объединял в себе широкий спектр функций, направленных на упрощение и оптимизацию процесса подготовки к путешествию. Как уже отмечалось во введении, актуальность разработки данного прототипа обусловлена востребованностью единой универсальной платформы, которая могла бы не только предоставлять пользователям доступ к информации о достопримечательности, но и включала в себя возможности составления детализированного плана поездки, а также хранения всех данных о путешествии в одном месте. При этом уникальной и особо важной особенностью разрабатываемого прототипа является его способность работать оффлайн, что предоставляет пользователям доступ к информации даже при отсутствии подключения к Интернету. Локализация данных непосредственно в рамках одного приложения делает его особенно удобным и практичным для путешественников, которые часто сталкиваются с необходимостью быстро и эффективно пользоваться доступными данными в условиях ограниченных ресурсов.

## **2 ПРОЦЕСС РАБОТЫ НАД ПРОЕКТОМ**

Наша работа над проектом была разделена на четыре основных этапа:

- планирование
- разработка
- тестирование
- защита

### **2.1 Планирование**

Планирование стало важнейшим этапом, без которого дальнейшая работа над проектом была бы просто невозможна. Чтобы обеспечить чёткое понимание требований и ожидаемого результата, наша команда приняла решение о необходимости составить техническое задание (ТЗ). Согласно ГОСТ 15.016-2016 [5], техническое задание является основным документом, который задаёт направление всей работы, формулируя требования к создаваемому изделию, технической документации, объёмам, срокам проведения работ и форме представления их результатов.

Таким образом, для дальнейшего развития проекта было составлено техническое задание, содержащее следующую информацию:

- название проекта,
- цели проекта,
- сроки выполнения,
- руководитель проекта,
- используемые в проекте термины и сокращения,
- требования к системе (функциональные, нефункциональные, требования к системе в целом, требования к видам обеспечения) [6],
- содержание работы (этапы проекта, задачи для каждого этапа),
- результаты работы.

Данный документ стал основой для нашей дальнейшей работы поскольку позволил нам структурировать все процессы. Итоговое техническое задание отражено в Приложении.

## **2.2 Разработка**

Этап разработки стал наиболее объёмной и важной частью работы над проектом. Наша команда состояла из пяти человек, включая одного тестировщика, одного дизайнера и трёх разработчиков. Руководитель проекта разделил задачи таким образом, чтобы каждый участник мог эффективно внести свой вклад в общий результат. Каждая задача была чётко прописана в техническом задании, включая сроки её выполнения и имена ответственных. Это позволило нам работать более организованно и отслеживать прогресс. Подробный перечень задач и их распределение приведены в Приложении.

## **2.3 Тестирование**

После завершения основных этапов разработки начался процесс тестирования. Как указано в разделе 2.2, в нашей команде был выделен один тестировщик, который отвечал за написание тестов и проверку работы прототипа в условиях различных входных данных.

Тестирование включало:

- проверку корректности работы функциональных модулей
- выявление ошибок и их документирование
- передачу отчётов о недочётах разработчикам для последующего исправления

После каждого цикла тестирования наша команда исправляла выявленные проблемы, чтобы гарантировать высокое качество конечного продукта. Это был итеративный процесс, который значительно улучшил стабильность и функциональность прототипа.

## 2.4 Защита

Финальным этапом нашей работы стала защита проекта. Это событие потребовало от нас тщательной подготовки, чтобы максимально эффективно представить результаты. Мы разработали:

- речь для каждого члена команды
- презентацию, демонстрирующую ключевые аспекты проекта
- план выступления, чтобы защита прошла структурированно и организованно

Защита проводилась в дистанционном формате перед студентами и преподавателями. Мы заранее отрепетировали наше выступление, чтобы избежать возможных проблем и сделать наше выступление уверенным и убедительным.



### 3 СУТЬ ПОСТАВЛЕННЫХ ЗАДАЧ

В рамках проекта я выполнял роль разработчика. Основные задачи, стоявшие передо мной, включали в себя несколько ключевых направлений работы, каждое из которых было важной составляющей в создании качественного прототипа приложения.

Первой и одной из самых ответственных задач стала проектировка архитектуры приложения. Это включало в себя разработку необходимых программных классов и структуры модулей проекта. На этом этапе я тщательно должен был продумать взаимодействие между компонентами, чтобы обеспечить простоту поддержки и масштабирования проекта в будущем.

Следующей задачей стал отбор API, которые бы могли покрывать всю требуемую логику приложения. Для достижения этой цели я должен был провести анализ доступных решений, сравнив их функционал, удобство интеграции и надежность. После отбора подходящих API я бы реализовал сетевые запросы к ним, что позволило бы приложению работать с внешними данными, обеспечивая пользователям доступ к актуальной информации о достопримечательностях, мероприятиях и кафе.

Также я должен был интегрировать локальную базу данных. Это решение продиктовано необходимостью обеспечить пользователям возможность оффлайн-доступа к данным. Я должен был подключить базу данных, спроектировать её структуру, а также реализовать методы взаимодействия с ней, что позволило бы приложению сохранять данные локально и предоставлять их пользователю даже при отсутствии подключения к Интернету.

Под конец нашей работы над прототипом мне дали задачу реализовать экран списка активностей. Этот экран был разработан в соответствии с макетами, созданными в Figma, и предоставлял пользователям возможность выбирать интересующий их вид активности, будь то достопримечательности,

мероприятия или кафе. Реализация этого элемента интерфейса требовала учёта всех мелочей, чтобы сделать взаимодействие пользователя с приложением максимально удобным и интуитивно понятным.

## 4 РЕШЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ

### 4.1 Проектировка архитектуры приложения

Первая задача, которая стояла передо мной, заключалась в проектировке архитектуры приложения. Чтобы подойти к этой задаче осознанно и профессионально, я решил сначала изучить существующие архитектурные подходы, которые применяются для разработки Android-приложений. Среди них выделяются MVC, MVP и MVVM.

#### 1. MVC (Model-View-Controller)

Архитектура MVC является одной из самых ранних и классических моделей проектирования. Она разделяет приложение на три основные компонента:

- Model: отвечает за управление данными, их хранение и бизнес-логику. Например, это может быть база данных или API, которые предоставляют данные для отображения.
- View: отвечает за отображение данных пользователю и получение пользовательского ввода. В Android это может быть Activity, Fragment или View.
- Controller: связывает Model и View, обрабатывает пользовательский ввод и обновляет Model или View в зависимости от изменений.

Преимущество MVC заключается в простоте реализации и разделении ответственности. Однако в контексте Android-приложений этот подход может привести к перегрузке контроллеров (например, Activity), что делает их сложными для тестирования и сопровождения.

#### 2. MVP (Model-View-Presenter)

MVP можно рассматривать как эволюцию MVC, адаптированную для мобильных приложений. В этой архитектуре:

- Model: аналогично MVC, отвечает за данные и бизнес-логику.
- View: отображает данные и взаимодействует с пользователем
- Presenter: является посредником между View и Model. Он управляет логикой приложения, вызывая методы Model и обновляя View.

Основное отличие MVP от MVC заключается в том, что View не взаимодействует напрямую с Model. Вместо этого она делегирует все запросы и действия Presenter. Это упрощает тестирование, так как Presenter не зависит от Android-компонентов.

### 3. MVVM (Model-View-ViewModel)

MVVM – это современный подход, который стал популярным благодаря поддержке Android Jetpack и библиотек, таких как Data Binding и LiveData. В этой архитектуре:

- Model: отвечает за данные и бизнес-логику, аналогично предыдущим подходам.
- View: отвечает за отображение данных и взаимодействие с пользователем.
- ViewModel: связывает Model и View, предоставляет данные для UI и обрабатывает действия пользователя.

Особенностью MVVM является использование механизмов двустороннего связывания (two-way data binding), что позволяет автоматически обновлять UI при изменении данных и наоборот. Это делает код лаконичным и упрощает поддержку.

Так как MVC неудобен для разработки мобильных приложений, а MVVM довольно трудный для освоения новичками, наша команда решила выбрать архитектуру MVP.

Далее я спроектировал структуру модулей проекта. При проектировании я опирался на рекомендации Google [7]. По итогу я реализовал три модуля ui

(здесь реализация экранов приложения и логика их отображения), domain (здесь описание модели данных, которые используются в приложении, а также данный модуль содержит UseCase классы – описание бизнес-логики приложения) и data (здесь код, который выполняет операции с данными – например, загрузка/отправка данных на сервер, кэширование данных локально на устройстве и так далее).

## **4.2 Подключение локальной базы данных**

Локальная база данных была необходима для того, чтобы в ней хранилась информация о маршрутах пользователя, и чтобы у пользователя был доступ к этой информации даже при отсутствии подключения к Интернету. Для этого использовалась библиотека Room [8].

## **4.3 Отбор API, которые будут покрывать логику приложения. Интеграция выбранных API в приложение**

Необходимо было исследовать предложенные руководителем API и выбрать набор подходящих API и их запросов для работы приложения. Вот список предложенных руководителем API:

### **1. Поиск и бронирование билетов/отелей**

Skyscanner Flights API [9]

Позволяет искать и бронировать авиабилеты.

Данное API не подошло, так как являлось платным.

Booking.com API [10]

Предоставляет доступ к базе данных отелей и возможности их бронирования

Данное API не подошло, так как не покрывало логику приложения.

### Amadeus Travel APIs [11]

Позволяет искать билеты, отели и точки аренды автомобилей.

API от Amadeus использовались для поиска отелей и информации об отеле по коду аэропорта.

### AviationStack API [16]

Позволяет получать различную информацию (дата/время прибытия по UTC, аэропорт прибытия, часовой пояс)

С помощью этой API извлекалась информация о коде аэропорта прибытия для дальнейшей фильтрации всей остальной информации по нему.

## 2. Информация о достопримечательностях

### TripAdvisor API [12]

Позволяет получать данные о популярных местах, отзывах и рейтингах.

Возникла проблема с пониманием документации, поэтому я перешёл к поиску другой API.

### Foursquare Places API [13]

Предоставляет данные о достопримечательностях, ресторанах и других интересных местах.

Данная API использовалась для нахождения достопримечательностей и ресторанов, а также подробной информации о этих местах, включая фото.

## 3. События

### Eventbrite API [14]

Позволяет искать и фильтровать события по категориям, местоположению и времени.

Возникла проблема с пониманием документации, поэтому я перешёл к поиску другой API.

#### **Ticketmaster API [15]**

Подходит для поиска мероприятий (концертов, выставок, спортивных событий) и покупки билетов.

Данное API должно было использоваться для поиска мероприятий по коду аэропорта и информации о нём, однако на этапе реализации сетевых запросов выяснилось, что отфильтровать запрос по коду аэропорта нельзя, а только по названию самого города и только на английском, но проблема в том, что от API AviationStack мы не получали информацию о городе прибытия, а только о коде аэропорта.

#### **4.4 Реализация экрана списка активностей**

Для создания путешествия было необходимо реализовать экран списка активностей с возможностью выбора интересующего вида активности (достопримечательности, мероприятия, кафе). Данный экран я реализовал согласно макетам, спроектированным нашим дизайнером в Figma. В процессе реализации экрана списка активностей я впервые столкнулся с работой с Fragment [17] и отправке данных от Fragment к Activity [18].

## **5 АНАЛИЗ ПРОДЕЛАННОЙ МНОЮ РАБОТЫ**

По итоге проделанной мною работы удалось реализовать архитектуру приложения, экран списка активностей, локальную базу данных, а также отобрать подходящие API и частично реализовать сетевые запросы к ним.

Основной трудностью в процессе разработки стало то, что я впервые столкнулся с большинством библиотек и инструментов, которые использовал. Освоение новых технологий требовало значительных временных и умственных затрат, особенно на этапе их изучения и интеграции в проект.

Дополнительно на эффективность работы влияли и внешние факторы, такие как выполнение заданий по другим учебным дисциплинам, которые отнимали много времени. Порой, из-за накопившейся усталости и нехватки моральных сил, было сложно сосредоточиться на выполнении задач курсового проекта. Эти обстоятельства существенно затрудняли придерживаться установленного плана, что приводило к некоторым задержкам в работе.

Однако данный процесс разработки позволил мне узнать много нового. Я познакомился с различными библиотеками и инструментами, которыми пользуются разработчики в своей профессиональной деятельности. И очень хорошо, что я начал это изучать, будучи на 1-м курсе. Это позволило мне заложить хорошую базу знаний и навыков для дальнейшего профессионального развития.



## **6 ВЗАИМОДЕЙСТВИЕ С КОМАНДОЙ И РУКОВОДИТЕЛЕМ ПРОЕКТА**

Во время разработки приложения наша команда не только эффективно работала, но и сумела создать дружескую атмосферу, что стало важным фактором успешного выполнения проекта. Еженедельные встречи в Google Meet проходили в непринуждённой обстановке, где каждый участник мог высказаться, поделиться своими идеями, обсудить сложные моменты и получить поддержку от коллег. Это помогало чувствовать себя частью единого целого.

Для распределения задач и контроля за их выполнением мы использовали систему Odoo, в которой наш руководитель чётко устанавливал задачи и сроки их выполнения. Каждому участнику команды было ясно, за что он отвечает и в какие сроки необходимо выполнить поставленную задачу.

Для оперативного общения и обсуждения различных вопросов касавшихся разработки был создан общий чат в Telegram. Если руководителю проекта требовалось уточнить какой-либо момент или обсудить детали индивидуально, он всегда связывался с нами через личные сообщения.

## **7 ОЦЕНКА РАБОТЫ РУКОВОДИТЕЛЯ**

Во время работы над приложением в нашей команде царила дружественная атмосфера. Нельзя не сказать, что это большая заслуга нашего руководителя, Данила Игоревича. Его подход к взаимодействию с командой с самого начала создавал ощущение открытости и доверия.

В первую очередь, что сразу бросилось в глаза, - это его предложение обращаться к нему на “ты”, что сразу сняло барьеры и сделало общение более непринуждённым. Кроме того, в самом начале в чате Telegram он предложил каждому из нас рассказать немного о себе, что помогло нам не только узнать друг друга, но и почувствовать себя частью единой команды. Эти мелочи, казалось бы, не имеют отношения к технической стороне проекта, но именно они сыграли важную роль в создании комфортной рабочей атмосферы.

Также стоит отметить то, насколько Данил был вовлечён в проект. Его готовность помочь на любом этапе работы, будь то обсуждение дизайна или решение сложной технической задачи, впечатляла. Если у кого-то из нас возникали вопросы или трудности, он всегда был на связи, чтобы дать совет или предложить альтернативное решение. Особенно хотелось бы отметить структурированность и доступность объяснение поставленных задач Данилом.

## **ЗАКЛЮЧЕНИЕ**

Подводя итоги, хочется сказать, что цель, которая заключалась в том, чтобы разработать прототип Android-приложения для планирования путешествий, была достигнута. К сожалению, из-за проблем с доступностью документации и отсутствием опыта у меня и других участников команды, решить все поставленные задачи не удалось, однако мною и, я уверен, моими коллегами был получен огромный опыт в разработке мобильных приложений.

Я считаю, что внёс немаленький вклад в достижение нашей цели, так как я реализовал архитектуру приложения, подключил локальную базу данных, частично реализовал сетевые запросы к API и реализовал экран списка активностей. И считаю, что остальные члены команды также внесли огромный вклад в полученный продукт. Я благодарен им и своему руководителю за проведённый вместе месяц и рад, что я получил ценнейший опыт в разработке мобильных приложений, который мне поможет в моей дальнейшей профессиональной деятельности.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Google Map – URL: <https://www.google.com.mm/maps> (дата обращения: 03.01.2025).
2. TripAdvisor – URL: <https://www.tripadvisor.ru/> (дата обращения 03.01.2025).
3. Sygic Travel – URL: <https://www.sygic.com/travel> (дата обращения 03.01.2025).
4. Visit a City – URL: <https://www.visitacity.com/> (дата обращения 03.01.2025).
5. ГОСТ 15.016-2016 Система разработки и постановки продукции на производство. Техническое задание. Требования к содержанию и оформлению. – М.: Стандартинформ, 2016. – 2 с.
6. ГОСТ 15.016-2016 Система разработки и постановки продукции на производство. Техническое задание. Требования к содержанию и оформлению. – М.: Стандартинформ, 2016. – 4 с.
7. Android Developers [Электронный ресурс]. – 2024. – URL: <https://developer.android.com/topic/architecture> (дата обращения 05.01.2025).
8. Android Developers [Электронный ресурс]. – 2024. – URL: <https://developer.android.com/training/data-storage/room> (дата обращения 05.01.2025).
9. Skyscanner [Электронный ресурс]. – 2024. – URL: <https://developers.skyscanner.net> (дата обращения 05.01.2025).
10. Booking.com [Электронный ресурс]. – 2024. – URL: <https://developers.booking.com> (дата обращения 05.01.2025).
11. Amadeus [Электронный ресурс]. – 2024. – URL: <https://developers.amadeus.com> (дата обращения 05.01.2025).

12. TripAdvisor [Электронный ресурс]. – 2024. – URL: <https://tripadvisor.mediaram.com> (дата обращения 05.01.2025).
13. Foursquare [Электронный ресурс]. – 2024. – URL: <https://location.foursquare.com/developer/> (дата обращения 05.01.2025).
14. Eventbrite [Электронный ресурс]. – 2024. – URL: <http://eventbrite.com/developer> (дата обращения 05.01.2025).
15. Ticketmaster [Электронный ресурс]. – 2024. – URL: <https://developer.ticketmaster.com> (дата обращения 05.01.2025).
16. AviationStack [Электронный ресурс]. – 2024. – URL: <https://aviationstack.com/documentation> (дата обращения 05.01.2025).
17. Android Developers [Электронный ресурс]. – 2024. – URL: <https://developer.android.com/guide/fragments> (дата обращения 05.01.2025).
18. Android Developers [Электронный ресурс]. – 2024. – URL: <https://developer.android.com/guide/fragments/communicate> (дата обращения 05.01.2025).

## **ПРИЛОЖЕНИЕ**

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.



3	Реализация архитектуры прототипа приложения	1-14 ноября	Артемов И.В.	Реализованные модули системы и основные 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: <ul style="list-style-type: none"> <li>- название проекта,</li> <li>- цели проекта,</li> <li>- сроки выполнения,</li> <li>- руководителя проекта,</li> <li>- используемые в проекте термины и сокращения,</li> <li>- требования к системе (функциональные, нефункциональные, требования к системе в целом, требования к видам обеспечения),</li> <li>- содержание работы (этапы проекта, задачи для каждого этапа),</li> <li>- результаты работы.</li> </ul> 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. 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.	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. 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.	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. 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.	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. 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 – 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 прделанной 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 прделанной 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 прделанной 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 прделанной 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 прделанную 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 прделанную работу и подготовленную презентацию с демонстрацией работы прототипа приложения на итоговой защите.	Мошкина Г.А.	18.11.2024 – 20.11.2024

