


**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ДОПУСТИТЬ К ЗАЩИТЕ
Профессор с возложенными
обязанностями заведующего
Кафедрой информационных
систем в искусстве и
гуманитарных науках**

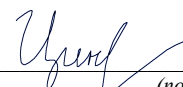
 (Борисов Н.В.)
" 21 " 05 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА


**Направление 09.03.03 «Прикладная информатика»
Уровень Бакалавриат
Основная образовательная программа
«Прикладная информатика в области искусств и гуманитарных наук»**

«Создание мобильного приложения для интерактивного изучения Санкт-Петербурга»

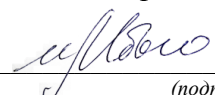
Студента Циневича Романа Константиновича


(подпись студента)

Руководитель: канд. физ.-мат. наук, доцент Захаркина Валентина Валентиновна


(подпись руководителя)

Консультант: ст. преп. Мбого Ирина Анатольевна


(подпись консультанта)

**Санкт-Петербург
2021**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ИСКУССТВ**

УТВЕРЖДАЮ
Заведующий Кафедрой



ЗАДАНИЕ

По подготовке выпускной квалификационной работы студента Циневича Романа Константиновича

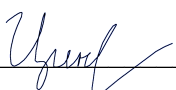
1. Тема работы: Создание мобильного приложения для интерактивного изучения Санкт-Петербурга
2. Срок сдачи студентом законченной работы: июнь 2021
3. Срок сдачи текста выпускной квалификационной работы для выкладывания в Blackboard: 25 мая 2021
4. Исходные данные к работе: -----
5. План-график выполнения дипломной работы

Номера и содержание этапов работы	Плановая дата сдачи
Разработка архитектуры мобильного приложения	Декабрь 2020
Разработка прототипов интерфейсов приложения. Создание системы прохождения квестов.	Декабрь 2020
Разработка и введение в эксплуатацию мобильного приложения	Апрель 2021
Написание теоретической части диплома	Май 2021
Защита теоретической и практической частей диплома	Июнь 2021

Руководители от Кафедры
Захаркина Валентина Валентиновна, к.ф.-м.н., доцент

Консультанты по работе
Мбого Ирина Анатольевна, старший преподаватель

Задание принял к исполнению Циневич Роман Константинович

Подпись студента 

Содержание

Содержание	3
Аннотация	5
Глоссарий	6
Введение	8
1. Обзор существующих приложений для интерактивного изучения городов	10
1.1. Приложение Questo	10
1.2. Приложение Fog of World	12
1.3. Вывод	13
2. Разработка концепта приложения	14
2.1. Система квестов	14
2.2. Посещение достопримечательностей	15
2.3. «Туман войны»	16
2.4. Система уровней	16
3. Проектирование структуры приложения и его функционала	17
3.1. Структура мобильного приложения	17
3.2. Функциональные требования мобильного приложения	18
4. Разработка дизайна	21
4.1. Разработка названия и логотипа	22
4.2. Цветовое решение	23
4.3. Исследование	23
4.4. Пользовательские сценарии	24
4.5. Прототипирование интерфейса	24
5. Разработка приложения	25
5.1. Выбор платформы для разработки	25
5.2. Выбор языка программирования	26
5.3. Маршрутизация	27
5.4. Геоинформационная система	29
5.4.1. Сравнение систем	29
5.4.2. Ограничения React Native и WebView	31

5.4.3. Вывод	32
5.5. Система геолокации	32
5.6. Разработка Backend	33
5.6.1. Системы регистрации и аутентификации	34
5.6.2. База данных	34
6. Наполнение контентом	37
Заключение	39
Список источников	40
Приложение А. Use case диаграмма мобильного приложения	42
Приложение Б. User Flow диаграммы	43
Приложение В. Пользовательские интерфейсы	46

Аннотация

Автор: Циневич Роман Константинович


Выпускная квалификационная работа на тему: «Создание мобильного приложения для интерактивного изучения Санкт-Петербурга» содержит 48 страниц, 8 рисунков, 2 схемы, 12 источников, 3 приложения.

Работа состоит из введения, шести разделов, заключения, списка использованных источников и приложений.

Во введении ставится цель и задачи данной работы, а также раскрывается ее актуальность.

В первых двух разделах проводится анализ существующих платформ для интерактивного изучения городов, сравниваются их достоинства и недостатки. Выбирается вектор, в направлении которого будет вестись проектирование и разработка.

Последующие разделы посвящены созданию мобильного приложения. Здесь поэтапно описан процесс проектирования структуры и базы данных, разработки дизайна, а также реализации клиентской и серверной частей. В заключении обобщается проделанная работа и подводятся итоги.

Автор работы _____  _____ Циневич Р.К.

Руководитель работы _____  _____ Захаркина В.В.

Глоссарий

1. Аватар – это картинка, которая используется в качестве изображения пользователя в социальных сетях, онлайн играх, форумах и др. Это может быть как фотография человека, так и просто любое графическое изображение.
2. Геоинформационная система – система сбора, хранения, анализа и графической визуализации пространственных (географических) данных и связанной с ними информации о необходимых объектах.
3. Диалоговое окно – специальный элемент интерфейса, окно, предназначенное для вывода информации и (или) получения ответа от пользователя.
4. Игровая механика – набор правил и способов, реализующий определённым образом некоторую часть интерактивного взаимодействия игрока и игры.
5. Нативный код – система команд (язык) конкретной вычислительной машины, который интерпретируется непосредственно микропроцессором или микропрограммами данной вычислительной машины.
6. Нативное приложение – это прикладные программы, которые были разработаны для использования на определённой платформе или на определённом устройстве. Нативные приложения оптимизированы под конкретные операционные системы, поэтому они могут работать корректно и быстро. Также они имеют доступ к аппаратной части устройств, то есть могут использовать в своём функционале камеру смартфона, микрофон, акселерометр, геолокацию, адресную книгу, плеер и т.д.
7. Туман войны – классическая игровая механика, при которой действие происходит на большой карте, и при этом окружающая территория на

начало партии является скрытой, и о них игрок узнает только после исследования соответствующей части карты. Стоит отметить, что в контексте мобильного приложения «Fog of World» термин «туман войны» не является ошибкой перевода, он лишь обозначает используемую механику.

8. Фреймворк – это программный продукт, который упрощает создание и поддержку технически сложных или нагруженных проектов.
9. API (англ. application programming interface) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

Введение

Санкт-Петербург, пожалуй, один из немногих городов России, которому удалось сохранить свою богатую историю и архитектурный облик. Город постоянно хочется исследовать, однако не всегда очевидно, как к этому подойти: все общеизвестные маршруты могут быть давно пройдены, туристические агрегаторы зачастую рассказывают об одних и тех же достопримечательностях, а чтобы найти что-то уникальное, приходится потратить изрядное количество времени.

Так появилась идея создать мобильное приложение, которое бы в игровом формате позволяло отслеживать места и маршруты, в которых пользователь уже побывал, постоянно предлагать новые точки интереса для изучения, а также проходить различные интерактивные квесты, своего рода экскурсии, которые бы открывали известные места с новых сторон.

На сегодняшний день туризм – один из самых популярных вариантов отдыха. Туристический рынок Санкт-Петербурга в особенности развит: город привлекает огромное количество людей не только со всей России, но и со всего мира. Город предлагает бесчисленное количество экскурсий, различных музеев, достопримечательностей и т.д. Также в интернете распространены ресурсы, помогающие организовать путешествие: эти сайты содержат информацию о различных популярных местах.

Однако при такой популярности туризма рынок мобильных приложений в этой сфере развит достаточно скромно и однобоко. В особенности в этом сегменте сильно не хватает приложений с интерактивной составляющей для индивидуального изучения городов. Большинство приложений являются лишь аналогами интернет-ресурсов и предлагают огромные массивы информации о местах, а пользователь уже самостоятельно должен отбирать те точки, которые ему интересны, что может занимать достаточно большое количество времени.

Данное мобильное приложение призвано упростить этот процесс, а также разнообразить его различными игровыми механиками, которые бы мотивировали пользователя исследовать город всё больше и больше.

Также такое мобильное приложение может быть интересно не только в рамках туризма, но в повседневной жизни. Люди склонны выделять время для изучения собственных городов на выходных. Данное мобильное приложение поможет разнообразить даже такие уже привычные действия, как путешествие от дома до работы и обратно.

Таким образом, цель данного мобильного приложения – разнообразить процесс исследования городов с помощью современных мобильных технологий. Выпускная квалификационная работа сфокусирована на исследовании и анализе приложений и интернет-ресурсов по интерактивному изучению городов и разработке самого мобильного приложения.

В работе решаются следующие задачи:

- разработка концепта мобильного приложения;
- проектирование структуры приложения;
- проектирование дизайна мобильного приложения;
- разработка самого приложения;
- наполнение приложения контентом.

1. Обзор существующих приложений для интерактивного изучения городов

В ходе исследования выяснилось, что на сегодняшний день ресурсов и приложений, направленных на интерактивное изучение городов, существует не так уж и много. Основная масса сервисов, связанных с достопримечательностями, скорее являются туристическими агрегаторами, цель которых предоставить актуальную информацию об известных местах и текущих мероприятиях, поэтому их можно рассматривать только как источник информации о популярных местах, и они не являются аналогами разрабатываемого приложения. Но всё же удалось найти несколько уникальных в своём роде приложений.

Для анализа были использованы два мобильных приложения: Questo и Fog of World. Несмотря на схожую тематику приложения используют два совершенно разных подхода к изучению городов.

1.1. Приложение Questo

Questo предлагает исследовать города при помощи системы квестов. Каждый квест – это некоторая вымышленная история, в которой пользователь является главным героем. В ходе прохождения квеста пользователю предлагается решить несколько загадок, ответы на которые находятся в окружающей городской среде.

Приложение доступно во многих городах по всему миру, однако до России сервис ещё не добрался. Все квесты платные, цена разнится от 5 до 20 евро. Пользователь может начинать и приостанавливать квесты в любом момент. По длительности квесты занимают в среднем 60 минут.

Как работают квесты?

1. Пользователь приходит на стартовую точку и запускает квест.

2. Пользователю рассказывается некоторая история, для продвижения по которой нужно решать головоломки. Как правило, ответ представляет из себя слово, число или дату, которые в той или иной форме можно найти в городской среде.

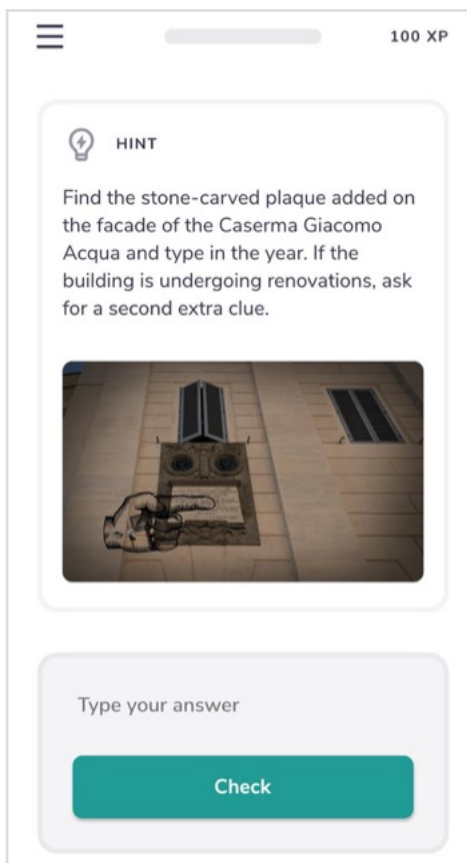


Рис. 1. Пример загадки. Источник: <https://questoapp.com>

3. После решения всех загадок квест заканчивается.

Таким образом, сами квесты рассказывают истории о местах в городе, а загадки в этих квестах побуждают пользователей исследовать городскую среду.

1.2. Приложение Fog of World

Fog of World предлагает совершенно другой способ исследования городов. Приложение использует популярную в компьютерных играх механику туман войны. Туман войны представляет собой игровую механику, когда действие происходит на большой карте, при этом территория на начало игры является скрытой, и она открывается игроку только после исследования соответствующей части карты. В этом приложении исследование осуществляется посредством геолокации: изначально вся карта покрыта полупрозрачным туманом, при передвижении пользователя по городу приложение получает координаты о его местоположении и рассеивает туман в этой точке на карте.

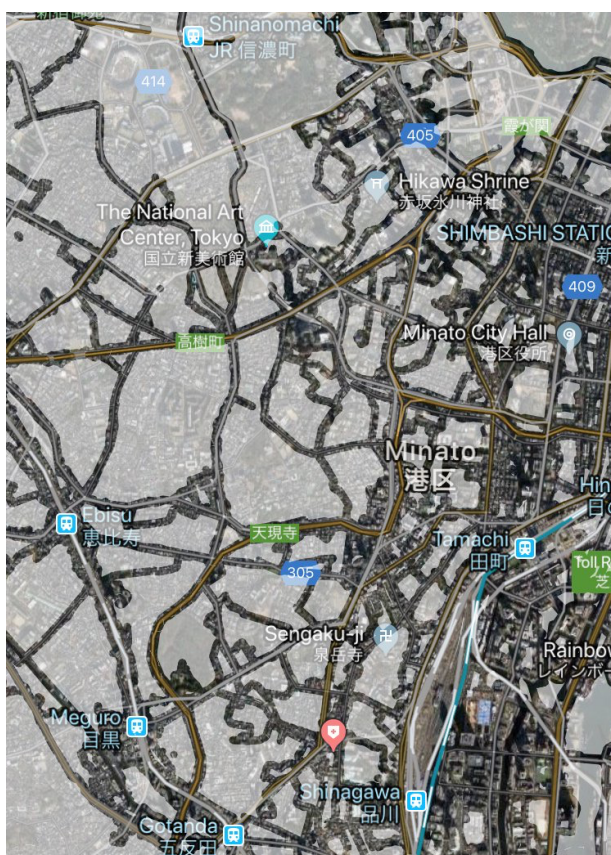


Рис. 2. Fog of World. Источник:
<https://fogofworld.com>

Таким образом, пользователь получает точную картину того, какие места в городе он уже посетил, а какие нет.

1.3. Вывод

Изучив примеры приложений по интерактивному изучению городов, было решено внедрить в приложение схожие с рассмотренными игровые концепции.

2. Разработка концепта приложения

При разработке данного приложения во главе угла стоит интерактивная составляющая. Задача приложения – не просто рассказать некоторую историю города и достопримечательности, а мотивировать пользователя самостоятельно изучать город. Перед нами стоял важный вопрос: как сделать самостоятельное изучение города интересным? На основе вышеперечисленных аналогов и собственной фантазии были придуманы три основные механики игры. Каждая из этих механик подразумевает непосредственное перемещение человека по городу.

2.1. Система квестов

Первой такой механикой стала система квестов. Квест – это своего рода интерактивные экскурсии. В ходе такого квеста рассказывается некоторая история, настоящая или вымышленная, о реальном месте, а для прогрессии пользователь должен решать предоставленные этим квестом задачи.

Каждая такая задача требует непосредственного нахождения пользователя на месте действия. Ответы на поставленные задачи находятся в окружающей среде, таким образом пользователь не сможет завершить квест удалённо.

Сценарий прохождения квеста следующий:

1. Пользователь приходит на стартовую точку квеста, указанную в приложении, и нажимает на кнопку «Старт»;
2. Далее открывается интерфейс задачи, в которой описывается некоторое повествование и в конце задается задача с подсказками с местоположением ответа на неё;
3. Пользователь изучает местность в поисках ответа на вопрос;

4. Найдя ответ вписывает его в специальное поле, после чего происходит проверка на правильность. При правильном ответе предоставляется следующая задача. В случае неправильного ответа появляется дополнительная подсказка, которая более конкретно указывает на правильный ответ.
5. По истечению задач квест считается пройденным. Пользователю открывается экран с результатами.

2.2. Посещение достопримечательностей

Другим важным аспектом приложения является посещение достопримечательностей, по своей сути это работает как коллекционирование. Люди любят коллекционирование, эта механика используется во многих настольных и компьютерных играх.

Все достопримечательности сгруппированы по типу: мосты, соборы, площади и т.д. У каждой такой группы есть счётчик, отображающий количество достопримечательностей из данной категории, которые пользователь посетил.

Отметка достопримечательностей работает с использованием геолокации. Происходит сравнение текущей позиции человека и метки геолокации достопримечательности, вычисляется расстояние между ними, если оно не превышает 50 м, то достопримечательность может быть помечена пользователем, как посещенная.

Также у каждой достопримечательности есть небольшая историческая справка.

2.3. «Туман войны»

Последней важной механикой является так называемый «туман войны». Туман войны представляет собой игровую механику, когда действие происходит на большой карте, при этом территория на начало игры является скрытой, и она открывается игроку только после исследования соответствующей части карты.

В приложении исследование осуществляется посредством геолокации: изначально вся карта покрыта полупрозрачным туманом, при передвижении пользователя по городу приложение получает координаты о его местоположении и рассеивает туман в этой точке на карте.

Благодаря этой механике пользователь сможет отслеживать места и улицы, по которым он уже ходил. Это позволит пользователю находить новые маршруты в городе, на которых он ещё не бывал.

2.4. Система уровней

В приложении используется система уровней. Уровни отображают общий прогресс пользователя в исследовании города. Каждая из вышеперечисленных активностей приносит очки опыта, которые увеличивают уровень пользователя. Это позволит внести соревновательный элемент: пользователи смогут соревноваться друг с другом по уровню.

3. Проектирование структуры приложения и его функционала

3.1. Структура мобильного приложения

Структура мобильного приложения – это схема всех разделов и экранов приложения, а также навигационная связь между этими структурными элементами. Разработка структуры приложения является важным этапом в проектировании системы.

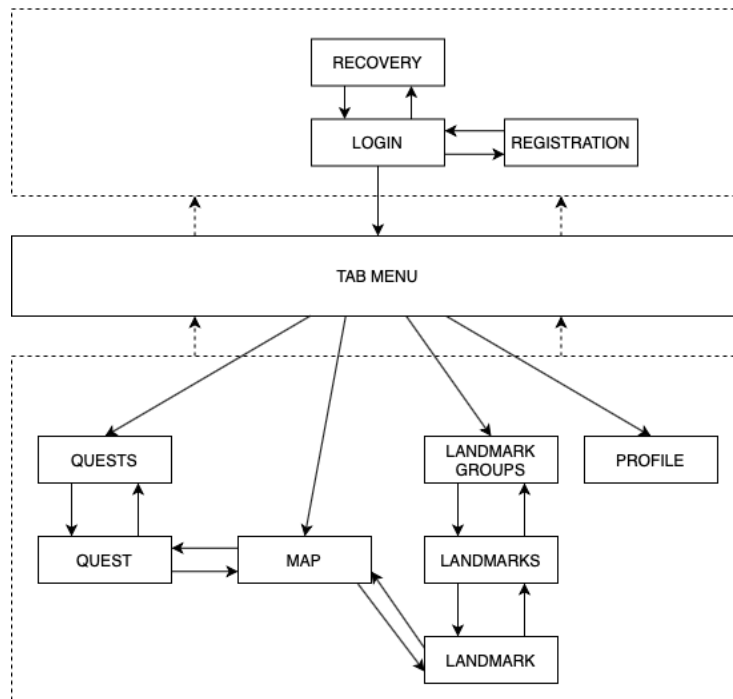


Схема 1. Структура мобильного приложения

Для доступа к основному функционалу приложения пользователь должен авторизоваться, после чего он попадает в само приложение.

Навигационное меню, находящееся внизу мобильного приложения, находится практически на каждом экране, за исключением экрана прохождения квеста, и содержит ссылки на важные разделы приложения: карта, квесты, достопримечательности и профиль пользователя. Таким образом, пользователь сможет свободно перемещаться между основными экранами приложения.

3.2. Функциональные требования мобильного приложения

1. Действия с аккаунтом:

- *Регистрация нового пользователя*

Для регистрации пользователю необходимо ввести электронную почту, придумать пароль, а также ввести своё имя. Опционально можно добавить аватар.

- *Авторизация пользователя с использованием электронной почты и пароля*

- *Восстановление пароля с помощью электронной почты*

При восстановлении пароля пользователю на почту придёт письмо со специальной ссылкой, перейдя по которой пользователя попросят ввести новый пароль.

- *Изменение пароля*

Для намеренного изменения пароля пользователь должен быть авторизован и должен помнить текущий пароль.

2. Действия с навигацией:

- *Перемещение пользователя между экранами с помощью навигационного меню*

3. Действия с картой:

- *Отображение меток квестов и достопримечательностей на карте*

После нажатия на метку должно появиться диалоговое окно, в котором дана краткая информация о квесте или достопримечательности и ссылка, ведущая на экран с полной информацией.

- *Отображение игровой концепции «туман войны»*

- *Отслеживание текущей локации пользователя*

При первоначальном входе приложение должно запросить разрешение у пользователя на отслеживание его местоположения.

- *Отображение маршрута от пользователя до некоторой точки*

4. Действия с квестами:

- *Отображение информации о квесте*

В этот пункт входят название, описание, средняя длительность, обложка квеста.

- *Отображение начала квеста на карте*

Специальная кнопка отобразит местоположение начала квеста на карте и построит маршрут до него.

- *Прохождение квеста*

Процесс прохождения квесты был описан подробно в предыдущих главах.

5. Действия с достопримечательностями:

- *Отображение информации о достопримечательности*

В этот пункт входят название, описание, фото, достопримечательности.

- *Отображение достопримечательности на карте*

Специальная кнопка отобразит местоположение достопримечательности на карте и построит маршрут до него.

6. Действия с профилем:

- *Отображение информации о пользователе*

На странице профиля отображается имя, аватар, статистика, уровень пользователя.

- *Редактирование информации о пользователе*

Пользователь может редактировать свои электронную почту, имя и аватар при условии, что пользователь авторизован.

На основе функциональных требований была построена Use case диаграмма, её можно найти в приложении А.

4. Разработка дизайна

Для разработки дизайна и интерфейса был использован инструмент Figma – онлайн-сервис для разработки интерфейсов и прототипирования. Ключевыми особенностями Figma являются возможность совместной работы над одним документом и доступ к макету прямо из окна браузера.

Figma использует систему компонентов. Компонент – это элементы пользовательского интерфейса, которым можно задавать общие стили, а при необходимости их можно быстро менять во всём макете сразу.

4.1. Разработка названия и логотипа

Любое приложение нуждается в названии, при этом желательно, чтобы оно легко читалось, было достаточно коротким и выделяющимся. Вполне разумно было попробовать отталкиваться от основной идеи приложения. Основная идея приложения – это исследование Санкт-Петербурга. Было перепробовано множество вариантов, однако финальное название оказалось достаточно очевидным – Explorin'Бург. Название максимально точно отражает основную идею мобильного приложения.

Также был разработан логотип приложения в двух вариантах. Апостроф у слова Explore выполнен в виде подошвы, что также подчеркивает интерактивность приложения.



Рис. 3. Логотип приложения

4.2. Цветовое решение

В качестве основной цветовой палитры мобильного приложения были выбраны цвета, которые соответствуют окраске фасадов в историческом центре Санкт-Петербурга.



Рис. 4. Цветовая палитра

4.3. Исследование

На этапе исследования были разработаны основная концепция приложения, его функциональные требования, выделены основные сущности, а также составлена структура мобильного приложения. Также были обозначены основные платформы продукта и изучены особенности интерфейсов каждой отдельной платформы.

4.4. Пользовательские сценарии

На основе материала, полученного в ходе исследования, был составлен список задач, которые сможет выполнять пользователь. Каждая задача была разбита на шаги, которые пользователю необходимо выполнить для достижения желаемого результата. На основе этих задач были составлены диаграммы сценариев использования приложения. Подробнее с диаграммами можно ознакомиться в Приложении Б.

4.5. Прототипирование интерфейса

На этапе прототипирования создается набросок продукта, в котором планируются расположение элементов и контента, функциональность и первоначальный стиль. Прототипирование включает создание всех экранов приложения и связей между ними, т.е. демонстрирует к чему и куда приводит взаимодействие будущего пользователя с элементами интерфейса.

В приложении было выделено 5 основных разделов: карта, квесты, достопримечательности, профиль и авторизация.

Дизайн претерпел две итерации. На первой итерации были сделаны только наброски всех экранов без единого стиля. На второй итерации уже была разработана логика взаимодействия между экранами, также все элементы были приведены в единый стиль.

С прототипами интерфейсов мобильного приложения можно ознакомиться в Приложении В.

5. Разработка приложения

5.1. Выбор платформы для разработки

Основным инструментом для разработки мобильного приложения стал React Native – фреймворк от Facebook, основанный на их же библиотеке для разработки пользовательских интерфейсов веб-приложений ReactJS, который позволяет создавать кроссплатформенные приложения. React Native позволяет собирать сложные интерфейсы из маленьких кусочков, которые именуются «компонентами».

Главным преимуществом React Native является использование единой кодовой базы, написанной на языке программирования JavaScript, которая позволяет одновременно разрабатывать приложение под iOS и Android. Все стандартные компоненты React Native автоматически конвертируются под соответствующую платформу, не нужно писать код дважды под каждую из них.

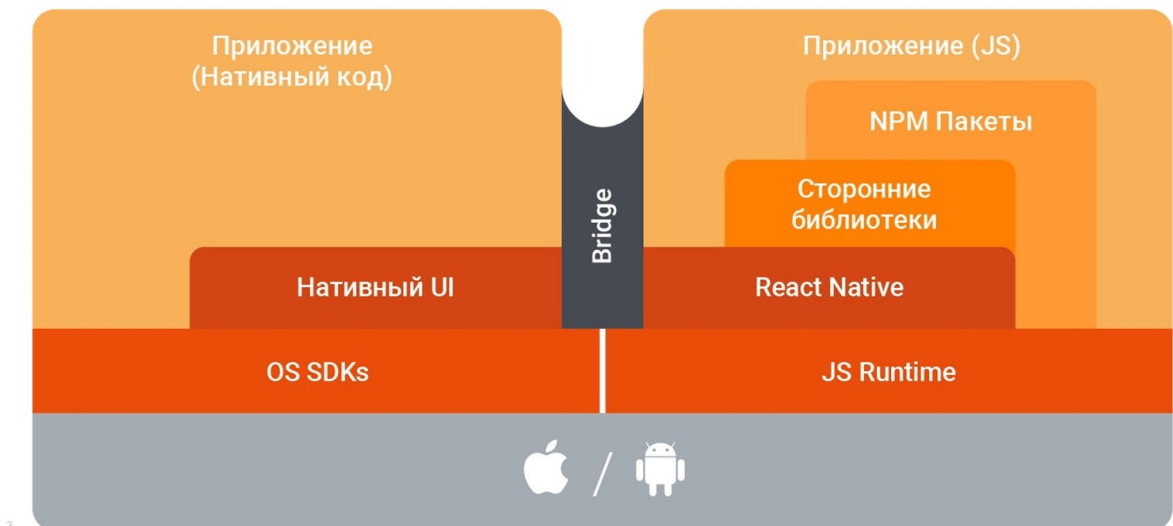


Рис. 5. Архитектура React Native. Источник: <https://habr.com/ru/company/oleg-bunin/blog/499636/>

Несмотря на то, что фреймворк использует JavaScript, приложение, которое мы запускаем на мобильном устройстве, использует нативный код: все

базовые элементы интерфейса, жесты и т.д. работают, как в нативном приложении используемой системы. Для этого каждое React Native приложение имеет два потока: основной и фоновый. Основной поток отвечает за рендер нативного пользовательского интерфейса платформы. Фоновый поток или JavaScript-поток отвечает за бизнес-логику приложения, т.е. в этом потоке выполняются JavaScript-код, вызовы API, обрабатываются события касания экрана и т.д. Общение между этими потоками происходит с помощью асинхронной системы обмена данными Bridge.

Другие преимущества React Native:

- Для отладки приложения можно использовать браузер
- Fast Reload – инструмент, который при любых изменениях в коде сразу же отображает их на экране мобильного устройства, для этого не нужно повторно собирать приложение.
- Большое сообщество
- Богатый выбор компонентов и готовых решений

5.2. Выбор языка программирования

Основным языком программирования стал TypeScript – это расширенная версия языка JavaScript от Microsoft, на сегодняшний день он является одним из самых популярных в веб-разработке.

TypeScript является компилируемым языком, т.е. код компилируется в понятный для браузеров JavaScript. Но благодаря статичной типизации, которая является главным преимуществом TypeScript'а, становится проще писать более понятный и надежный код, что помогает избежать многих проблем и ошибок, которые могли бы возникнуть при разработке на JavaScript, особенно при работе над большими проектами.

5.3. Маршрутизация

Так как мобильное приложение предполагает наличие нескольких экранов, то нужно реализовать маршрутизацию между этими экранами. Фреймворк React Native сам по себе не предоставляет такой возможности, для этого существуют множество отдельных библиотек. Для приложения была выбрана популярная библиотека React Navigation, которую рекомендует сам Facebook, разработчик фреймворка.

Библиотека предоставляет богатый выбор компонентов для самых разных видов навигаций. Каждый из этих компонентов имеет нативный UI под каждую платформу, т.е. каждый элемент выглядит и работает на Android и iOS так, как это предполагает операционная система. Но в тоже время библио-

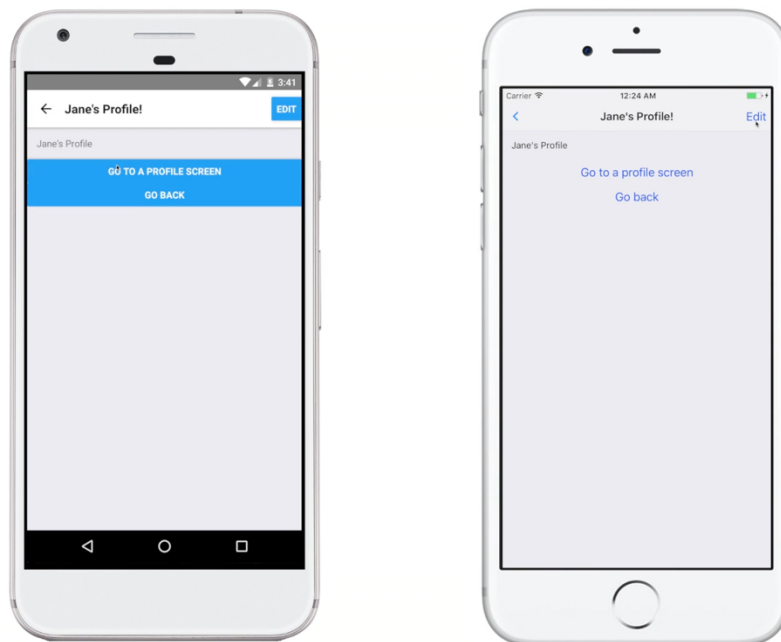


Рис. 6. Сравнение интерфейсов в React Navigation. Источник:

<https://m.oursky.com/tricky-way-to-update-the-title-in-navigation-bar-in-react-navigation-d3de5487bfeb>

тека предоставляет широкий инструментарий как для кастомизации существующих компонентов, так и для создания своих собственных.

Библиотека также предоставляет готовые решения различных навигационных меню. Например, нижнее и верхнее «табовое» меню, боковое меню. В приложении было использовано первое.

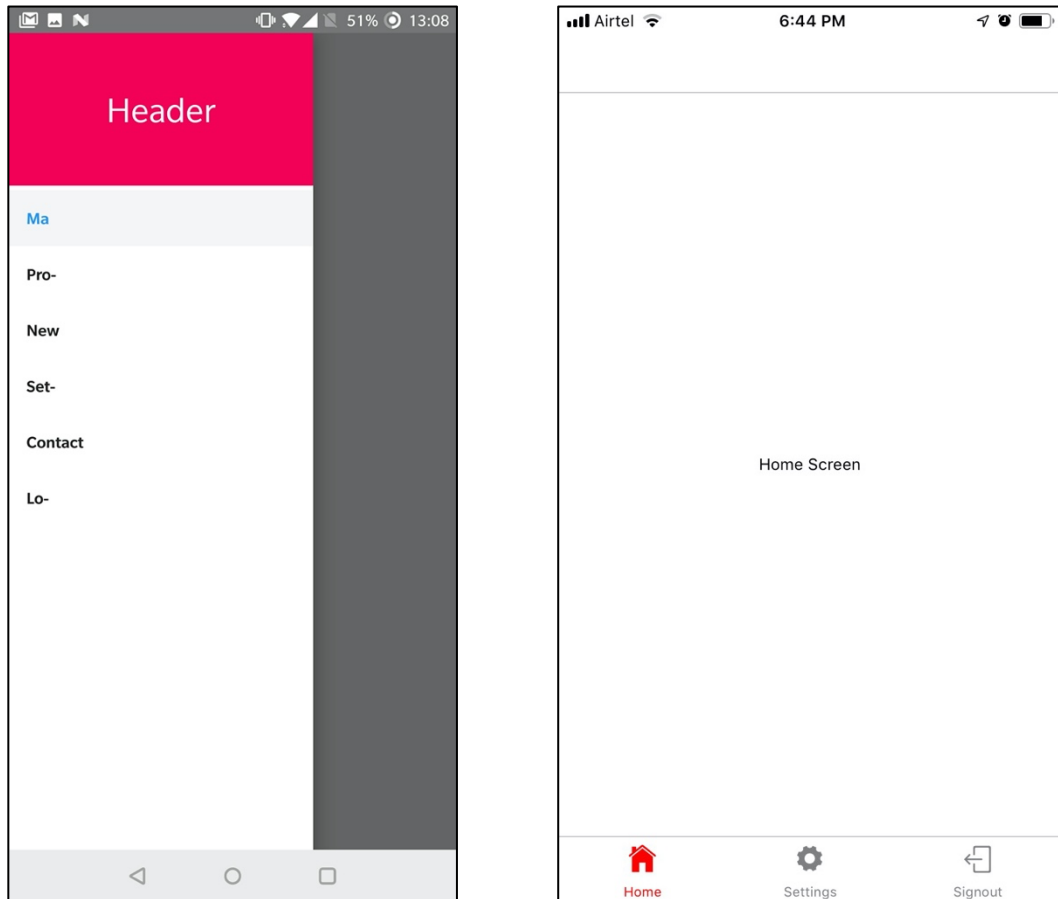


Рис. 7. Примеры навигационных меню. Источник: <https://github.com/react-navigation/react-navigation/issues/3389>

Библиотека оперирует понятием навигационный стек. Стек позволяет приложению переходить между экранами, а также позволяет управлять историей навигации. По умолчанию для каждого экрана стек предоставляет шапку, в которой указано название экрана, и, в случае перехода на новый экран, ссылку для возвращения на предыдущий.

Для навигации используется объект `navigation`. Для перемещения между экранами у этого объекта есть методы `navigate` и `push`, которые принимают

параметр `name` (название экрана). Также можно передавать различную информацию на следующий экран, это делается путём добавления параметра `params` после `name`.

5.4. Геоинформационная система

В разрабатываемом мобильном приложении важную роль играет геоинформационная система (далее – ГИС). Она содержит данные о квестах, достопримечательностях, а также отображает местоположение пользователя и игровую концепцию «туман войны».

Таким образом, ГИС должна иметь следующий базовый функционал:

- отображать различные метки;
- отображать местоположение пользователя;
- отображать маршруты от пользователя до некоторого места;
- работать с растровыми и векторными слоями.

В итоге были изучены следующие системы: Яндекс.Карты, 2ГИС, Google Maps, Apple Maps и Leaflet. Во всех этих системах присутствует базовый функционал, однако у всех есть свои ограничения.

5.4.1. Сравнение систем

Системы Яндекс.Карты и Google Maps не подошли сразу по причине ограниченного количества запросов к серверам соответствующих сервисов в бесплатных версиях.

Apple Maps доступна бесплатно только при нативном использовании на устройствах Apple, операционную систему Android ГИС не поддерживает.

Веб-версия системы имеет также ограничения в бесплатной версии, как и две предыдущие ГИС.

2ГИС полностью бесплатный для некоммерческого использования. Эта же система в свою очередь основана на открытой JavaScript-библиотеке Leaflet, которая также бесплатная, имеет огромное сообщество и большой выбор готовых решений.

Для Leaflet сообществом был разработан плагин Leaflet MaskCanvas, который идеально подходит для реализации исследовательской механики туман войны. При подключении плагина на всю карту наносится сплошной одноцветный слой, после чего системе передается массив географических координат, которые отображаются в виде отверстий в этом слое.

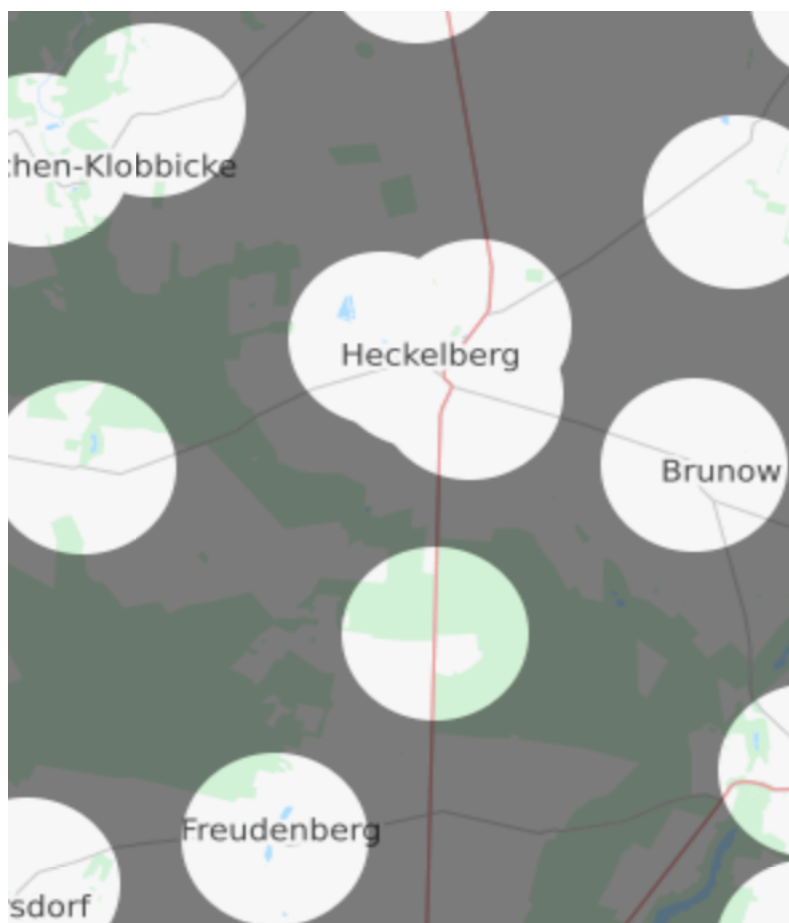


Рис. 8. Leaflet MaskCanvas. Источник:
<https://domoritz.github.io/vbb-coverage/>

Leaflet выглядит выигрышно на фоне остальных ГИС, однако у этой библиотеки есть один недостаток – у неё существует только веб-версия, т.е. её нельзя напрямую запускать на мобильных устройствах.

5.4.2. Ограничения React Native и WebView

В процессе выбора ГИС выявились ограничения у фреймворка React Native. Выяснилось, что фреймворк нативно поддерживает весьма узкий спектр ГИС, ограничиваясь лишь картами от Google и Apple.

Для использования других ГИС есть возможность использовать их веб-версию с помощью библиотеки WebView. Эта библиотека предоставляет компонент, который позволяет загружать и отображать веб-страниц. Есть два варианта использования этого компонента:

1. предоставить компоненту ссылку на веб-сайт и при наличии подключения к сети эта страница загрузится и отобразится на экране мобильного устройства.
2. предоставить заранее подготовленный HTML-код, который уже без подключения к сети сможет отобразиться на экране.

Второй вариант представляется лучшим, так как у него нет зависимости от сети, а значит гарантируется более стабильная работа.

Веб-страница не является нативным элементом, из-за чего сильно усложняется общение между приложением и самой страницей, т.к. между ними нет прямого канала. Чтобы решить эту проблему библиотека WebView предоставляет API для такой коммуникации. Общение между приложением и веб-страницей происходит посредством POST-запросов. Процесс выглядит примерно следующим образом:

1. Приложение исполняет функцию, которая должна как-то изменить состояние веб-страницы.

2. WebView отправляет POST-запрос, содержащий информацию об изменениях, которые должны произойти.
3. Веб-страница принимает этот запрос и уже на самой странице обрабатывает некоторый JavaScript-код, который изменяет её состояние.

Такой же алгоритм применим и для общения в обратную сторону.

5.4.3. Вывод

Проанализировав перечисленные ГИС, было решено использовать библиотеку Leaflet, т.к. она удовлетворяет всем базовым требованиям проекта, а также обладает большим количеством готовых решений, которые решают более сложные технические задачи проекта. Так как эта система существует только в вебе, то для её реализации на мобильных устройствах была использована библиотека WebView, позволяющая отображать на мобильных устройствах веб-страницы.

5.5. Система геолокации

Официальный модуль для геолокации React Native использует устаревший сервис для отслеживания местоположения на Android. Поэтому было решено использовать библиотеку react-native-geolocation-service. Эта библиотека расширяет возможности официального модуля, но для платформы Android использует рекомендуемый компанией Google сервис. На данный момент это, пожалуй, самый популярный модуль для отслеживания геолокации для React Native.

5.6. Разработка Backend

Backend – это программно-аппаратная часть приложения. Для ускорения разработки было принято решение использовать сервисы платформы Firebase. Платформа Firebase предоставляет свои сервисы по модели Backend-as-a-Service (с англ. «бэкенд как услуга»). По этой модели платформа предоставляет разработчикам веб-приложений и мобильных приложений инструментарий для реализации таких функций, как управление пользователями, отправление уведомлений, интеграция с социальными сетями, а также позволяет связать приложение с облачными функциями соответствующей платформы.

Firebase предоставляет свои услуги по нескольким тарифам, в том числе есть и бесплатный тариф, который подходит для разработки и тестирования приложения, а также для первоначального выпуска мобильного приложения в магазинах приложений.

Для использования своих сервисов Firebase предоставляет API для различных платформ: React Native, iOS, Android, Web и т.д. Благодаря умной архитектуре возможно использование этих сервисов не только при подключении к сети, но и при её отсутствии.

Для разработки мобильного приложения были использованы следующий сервисы Firebase:

1. Cloud Firestore – это облачная NoSQL база данных, предназначенное для хранения документов небольших размеров (до 6 Мб).
2. Cloud Storage – это облачное хранилище, предназначенное для крупных медиа-файлов.
3. Authentication – это сервис аутентификации.

5.6.1. Системы регистрации и аутентификации

Для получения доступа к функционалу мобильного приложения пользователь должен войти под своей учетной записью или зарегистрироваться в системе, если учетной записи нет.

Для реализации такой функциональности был использован сервис Authentication. Этот сервис предоставляет возможность регистрации новых пользователей по логину и паролю, по мобильному телефону, также у сервиса есть возможность отправлять различные уведомления на электронную почту и в виде SMS на мобильные устройства.

Регистрация и аутентификация происходят по электронной почте и паролю. При желании пользователь может изменить свои электронную почту и пароль. В случае утери пароля пользователь может запросить восстановление пароля, в таком случае ему на электронную почту придёт инструкция для установки нового пароля.

Также этот сервис обеспечивает высокую безопасность создаваемым и передаваемым данным.

5.6.2. База данных

Для приложения была разработана схема базы данных. В базе данных были выделены 3 ключевые сущности:

1. Пользователь
2. Квест
3. Достопримечательность

Сущность «Пользователь» необходима для авторизации, регистрации, смены пароля, отображения общей информации о пользователе и его прогрессе в изучении города. Имеет следующие поля: идентификатор, имя, электронная почта, пароль, уровень.

Сущность «Квест» содержит описание и его координаты. Имеет следующие поля: идентификатор, название, описание, координаты, очки опыта, длительность, дистанция. Также с этой сущностью напрямую связана другая – «Задача». Каждый квест состоит из нескольких задач. У каждой задачи есть описание, тип задачи, правильный ответ.

Сущность «Достопримечательность» содержит историческую справку и координаты. Имеет следующие поля: идентификатор, название, описание, координаты.

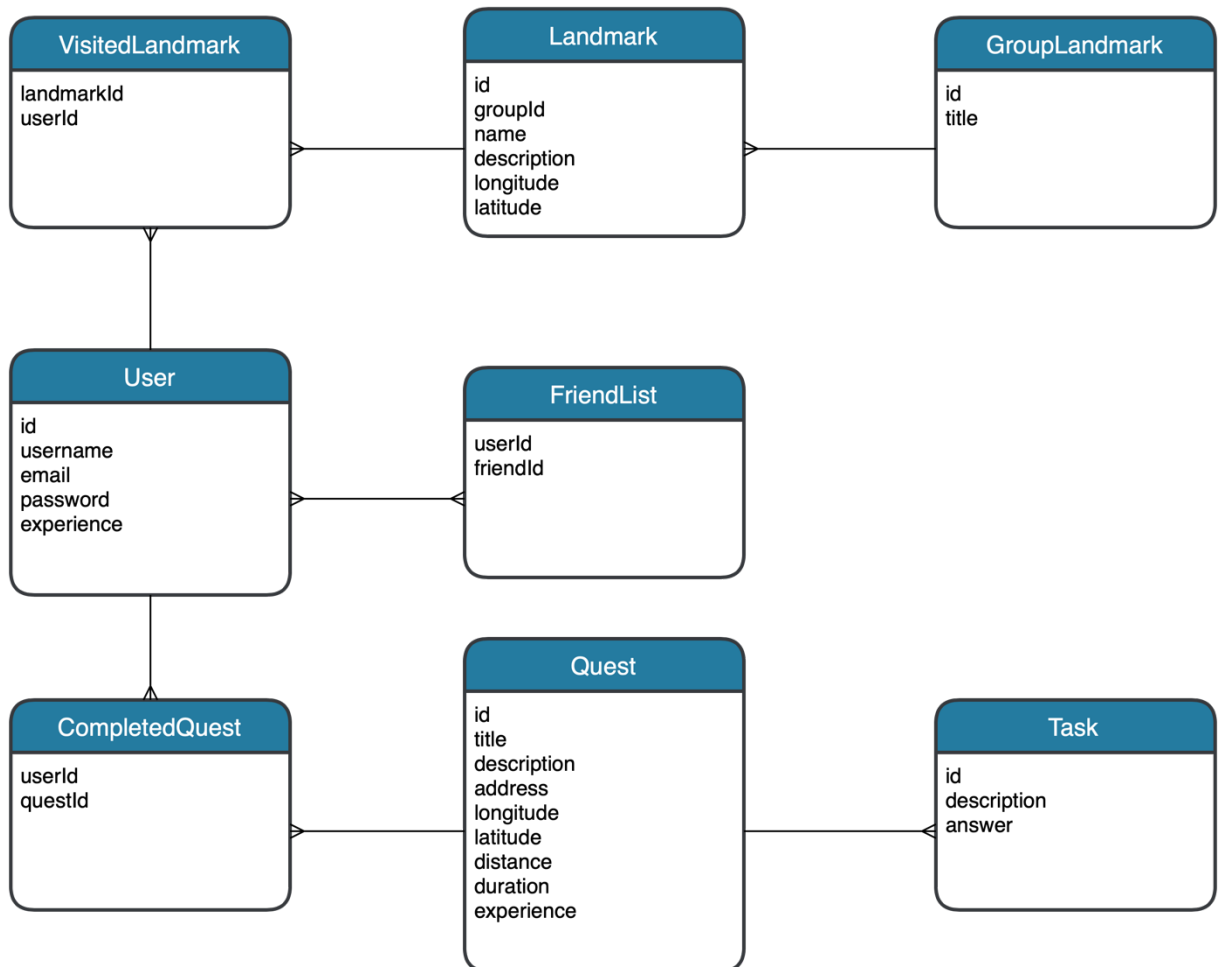


Схема 2. Структура базы данных

В приложении в качестве базы данных используется сервис Cloud Firestore. Это гибкая, масштабируемая, облачная NoSQL база данных, основанная на объектах, которые называются документами, данные хранятся в виде ключ-значение.

Для хранения картинок используется сервис Cloud Storage. Этот сервис использует эффективные алгоритмы, которые позволяют скачивать файлы независимо от качества сети.

Оба сервиса дают возможность настроить правила безопасности, что защищает данные от злоумышленников. Правила безопасности позволяют контролировать кто и как может получать доступ к данным, например, можно ограничить пользователей, которые имеют доступ к данным.

6. Наполнение контентом

Одной из игровых механик этого приложения является посещение достопримечательностей, а также возможность просматривать информацию об этих местах. Таким образом, приложение должно иметь некоторую базу с достопримечательностями.

С получением такой базы данных нам помог портал «Открытые данные Санкт-Петербурга», разработанный по инициативе Правительства Санкт-Петербурга. Портал предоставляет большое количество наборов данных, все эти данные можно использовать бесплатно для некоммерческих и коммерческих проектов, единственное условие использования – указать источник этих данных. Все наборы данных можно получить в следующих форматах: csv, xlsx, zip, xml.

Одним из наборов данных являются достопримечательности. В этом наборе представлено около 700 различных достопримечательностей с различной информацией о них. Для каждой достопримечательности из предоставленного набора данных есть следующие поля: тип объекта, название (на русском и английском языках), адрес, описание объекта и его историческая справка (на русском и английском языках), координаты и др.

Было решено использовать этот набор данных, т.к. он предоставляет все нужные данные для разработанной игровой механики.

Однако был использован не весь набор, а лишь небольшая часть с целью более удобного тестирования мобильного приложения. Из этого набора были взяты данные о следующих типах объектов: мосты, парки и сады. В итоге вышло чуть больше 200 достопримечательностей.

Далее последовала обработка этих данных: удаление лишних полей, очистка данных (в текстовых полях присутствовали артефакты в виде специальных символов). После обработки данные были конвертированы в формат JSON и загружены в базу данных.

Заключение

В результате данной выпускной квалификационной работы было разработано мобильное приложение для интерактивного изучения Санкт-Петербурга.

Для мобильного приложения была придумана концепция и разработаны игровые механики, которые помогают в изучении города.

В процессе разработки были придуманы и реализованы название, логотип и пользовательские интерфейсы мобильного приложения. Также были проанализированы различные технологии и были выбраны наиболее подходящие под разработку.

За время разработки обучающей платформы были пройдены полные циклы разработки дизайна, проектирования и реализации клиентской и серверной частей.

Список источников

1. Questo [Электронный ресурс] – Режим доступа: <https://questoapp.com>, свободный
2. Fog of World [Электронный ресурс] – Режим доступа: <https://fogofworld.com>, свободный
3. Документация Яндекс.Карты [Электронный ресурс] – Режим доступа: <https://yandex.ru/dev/maps/jsapi/doc/2.1/quick-start/index.html/>, свободный
4. Документация Google Maps [Электронный ресурс] – Режим доступа: <https://developers.google.com/maps/documentation/javascript/overview>, свободный
5. Документация Apple Maps [Электронный ресурс] – Режим доступа: <https://developer.apple.com/documentation/mapkit/>, свободный
6. Документация 2ГИС [Электронный ресурс] – Режим доступа: <https://api.2gis.ru/doc/maps/ru/quickstart/>, свободный
7. Документация Leaflet [Электронный ресурс] – Режим доступа: <https://leafletjs.com/reference-1.7.1.html>, свободный
8. Плагин Leaflet MaskCanvas [Электронный ресурс] – Режим доступа: <https://github.com/domoritz/leaflet-maskcanvas>, свободный
9. Документация React Native [Электронный ресурс] – Режим доступа: <https://reactnative.dev/docs/getting-started>, свободный
10. Документация TypeScript [Электронный ресурс] – Режим доступа: <https://www.typescriptlang.org/docs>, свободный

11. Документация React Navigation [Электронный ресурс] – Режим доступа: <https://reactnavigation.org>, свободный
12. Документация Firebase [Электронный ресурс] – Режим доступа: <https://firebase.google.com/docs>, свободный

Приложение А. Use case диаграмма мобильного приложения



Рис. 1. Use case диаграмма мобильного приложения

Приложение Б. User Flow диаграммы

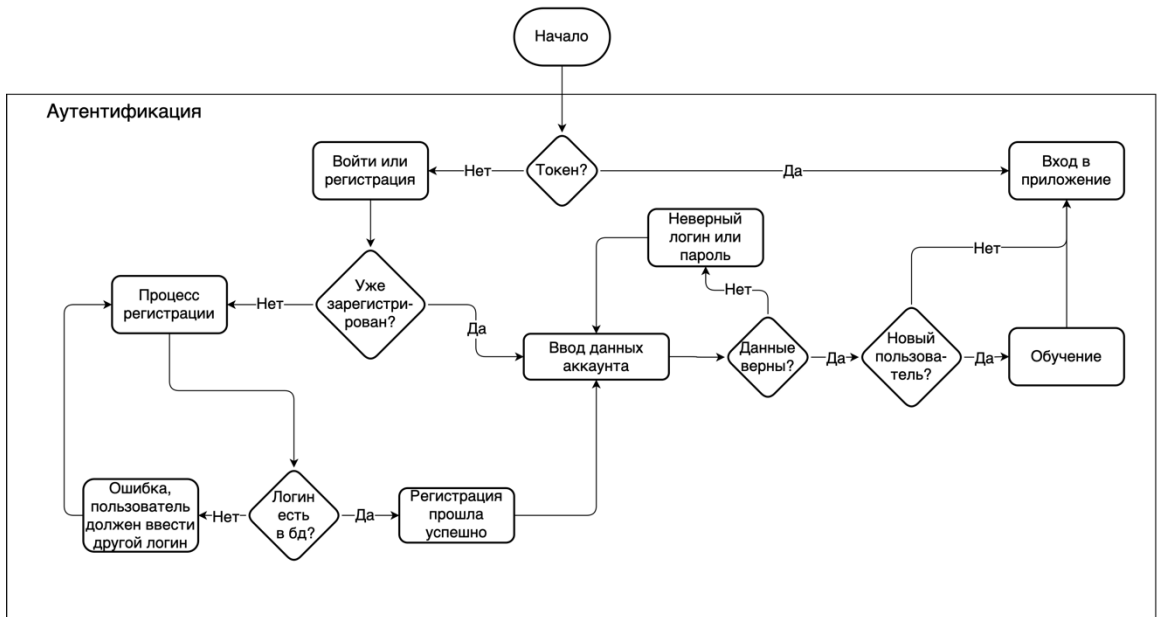


Рис. 1. User Flow диаграмма раздела «Аутентификация»

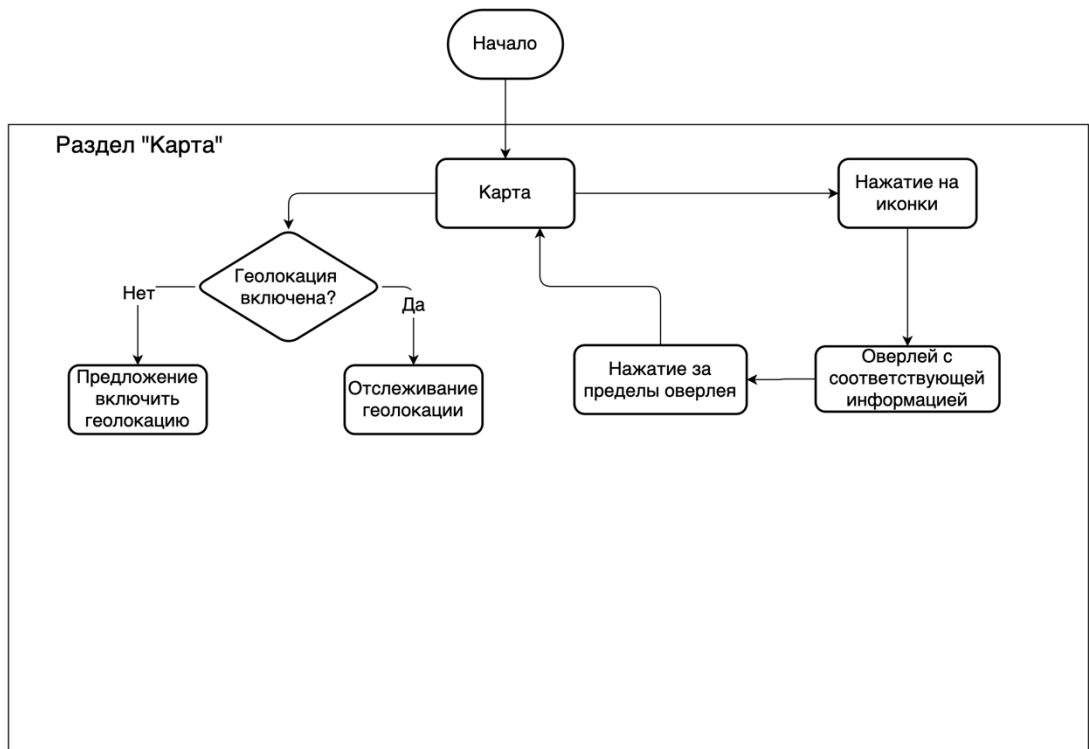


Рис. 2. User Flow диаграмма раздела «Карта»

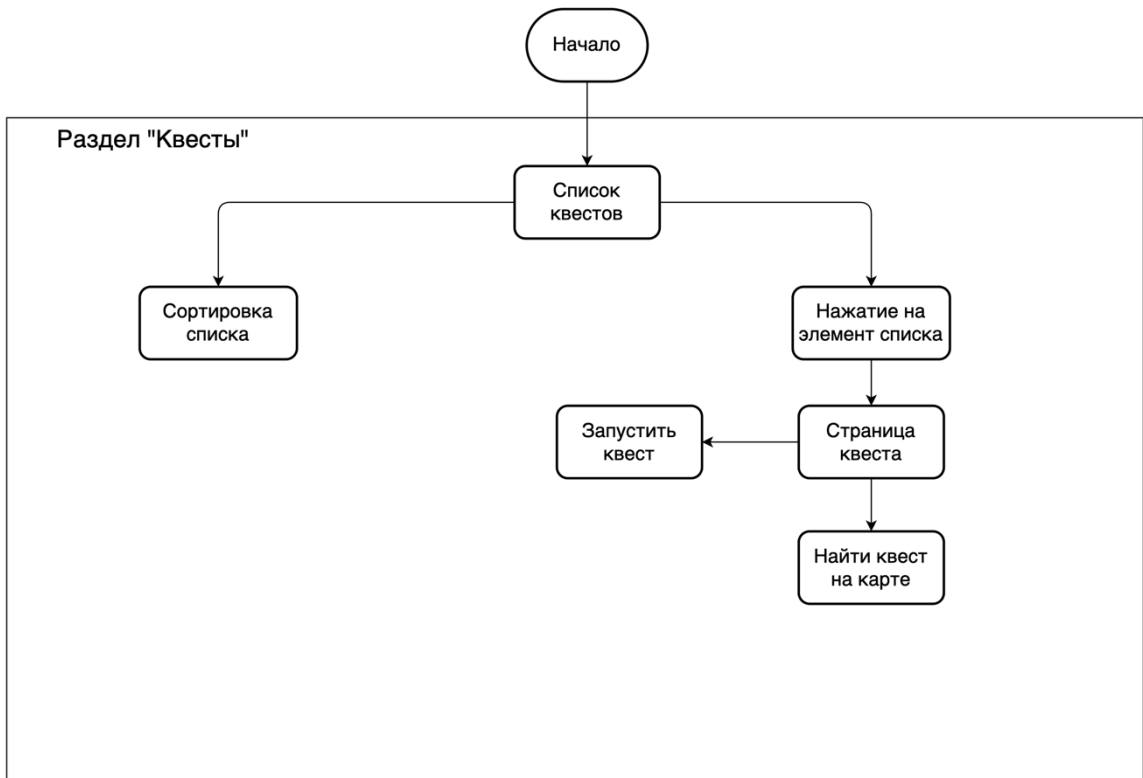


Рис. 3. User Flow диаграмма раздела «Квесты»

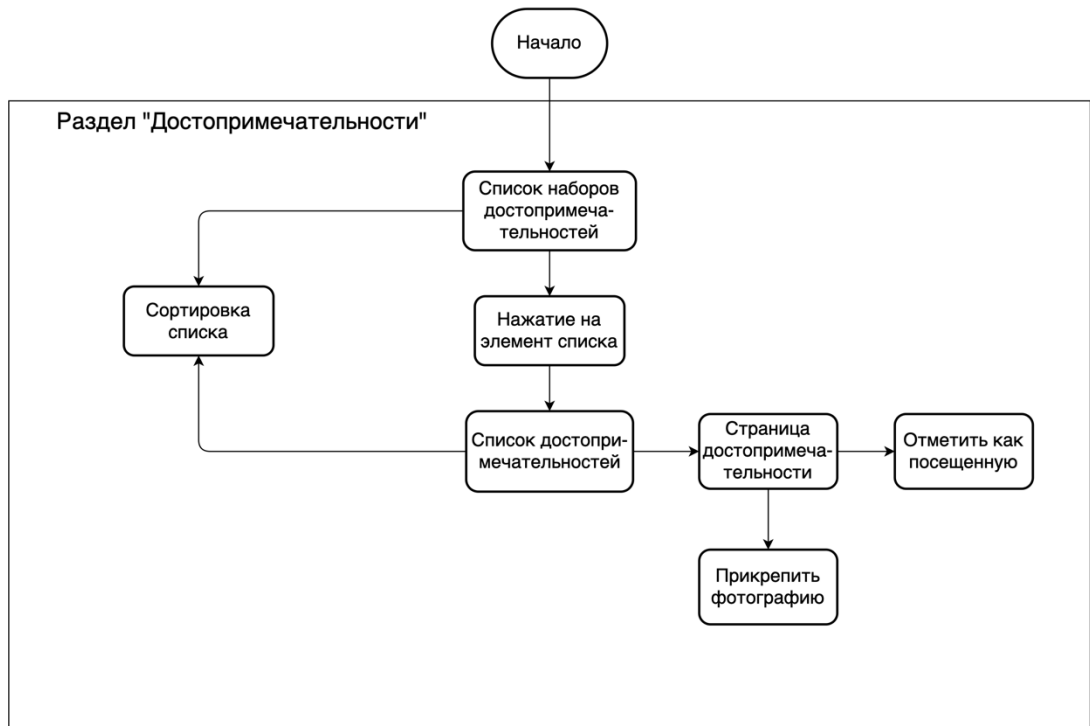


Рис. 4. User Flow диаграмма раздела «Достопримечательности»

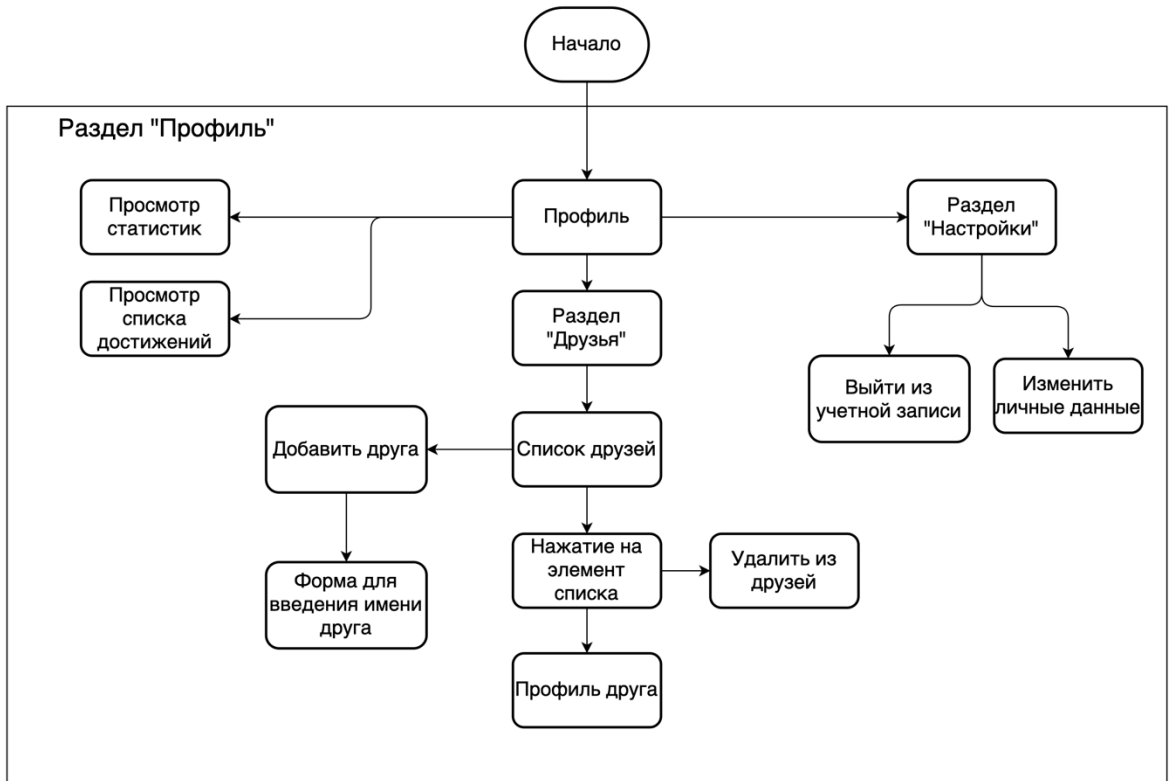


Рис. 5. User Flow диаграмма раздела «Профиль»

Приложение В. Пользовательские интерфейсы

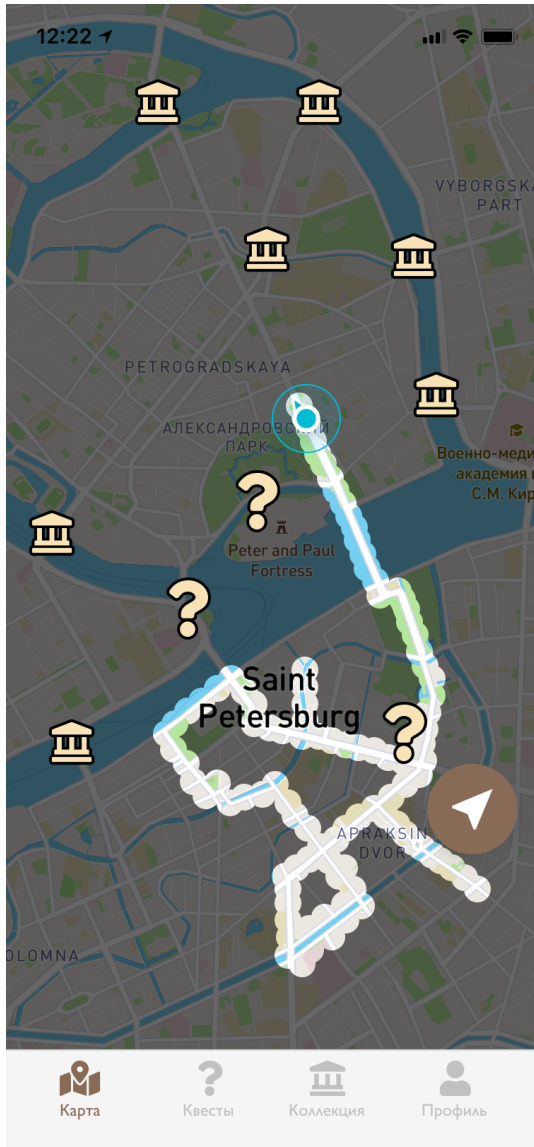


Рис. 1. Интерфейс раздела «Карта»

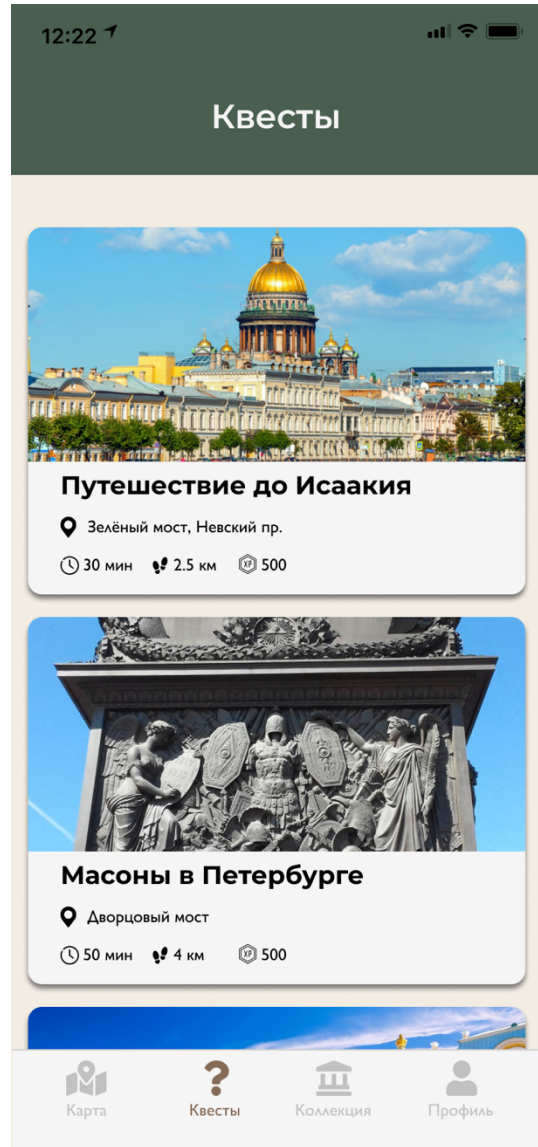


Рис. 2. Интерфейс раздела «Квесты»

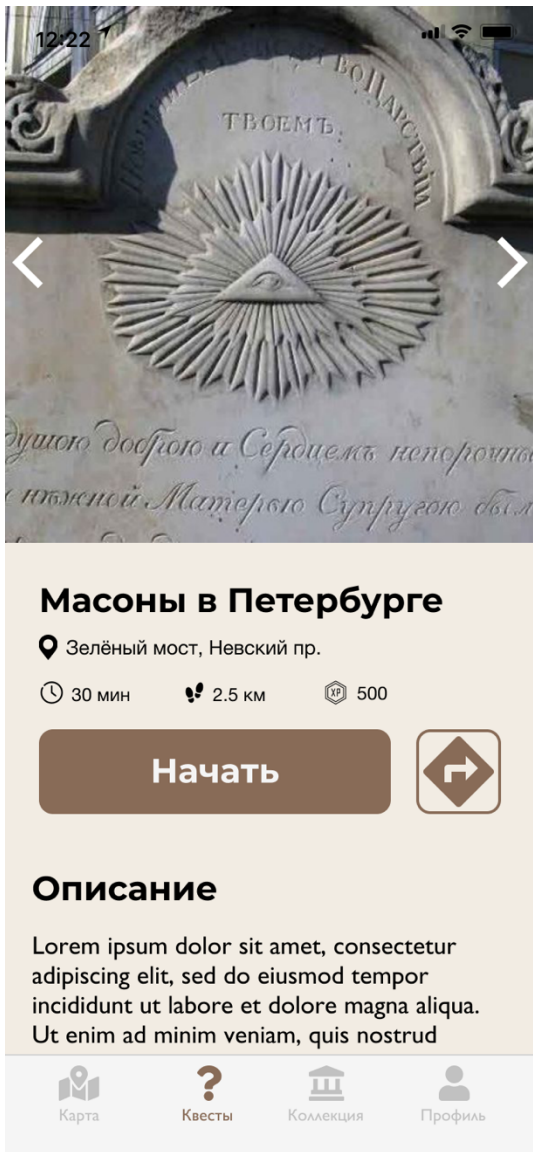


Рис. 3. Интерфейс страницы квеста

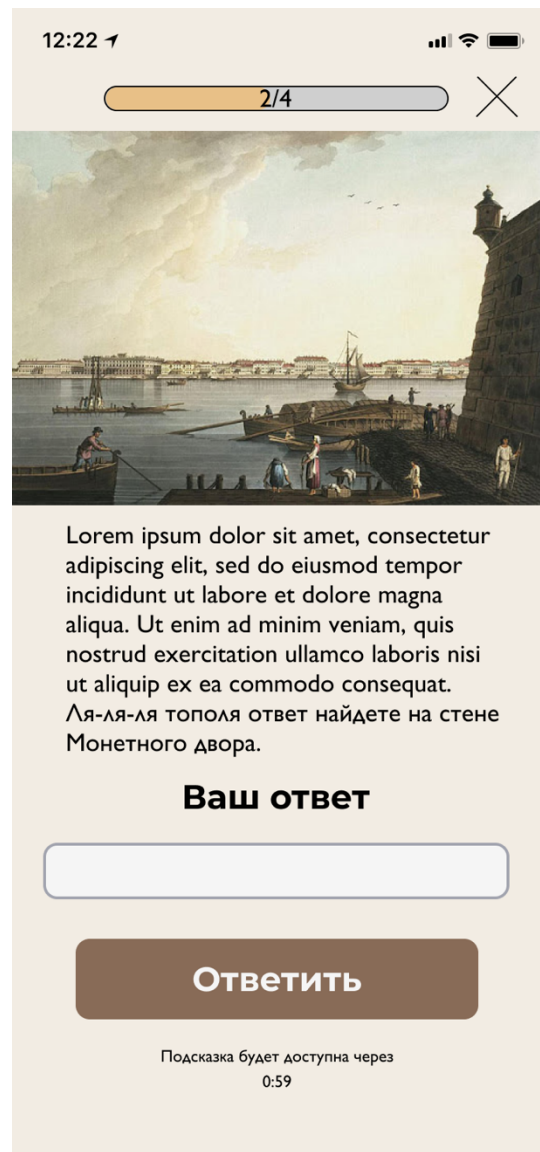


Рис. 4. Интерфейс прохождения квеста

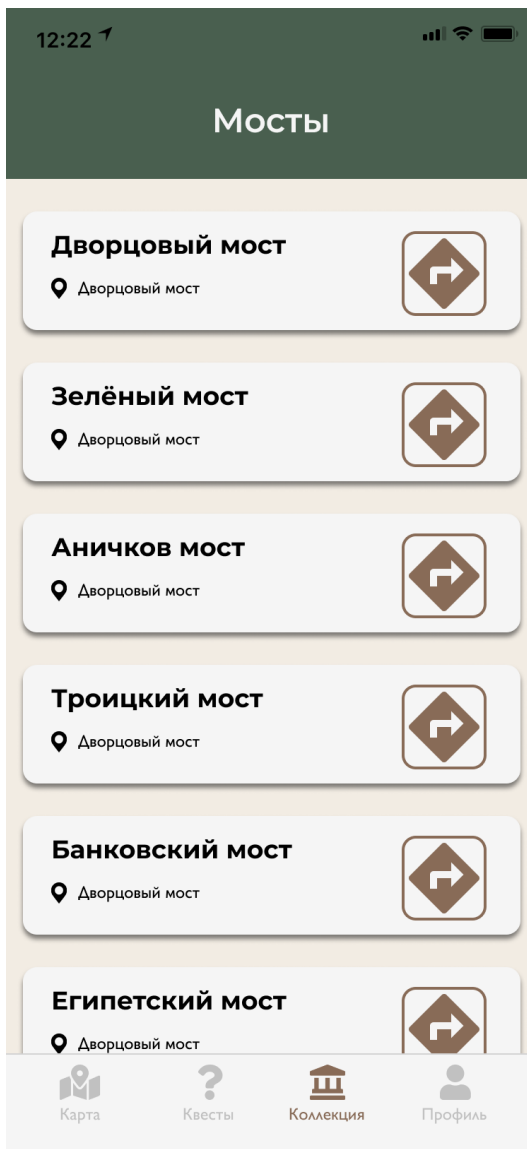


Рис. 5. Интерфейс набора достопримечательностей

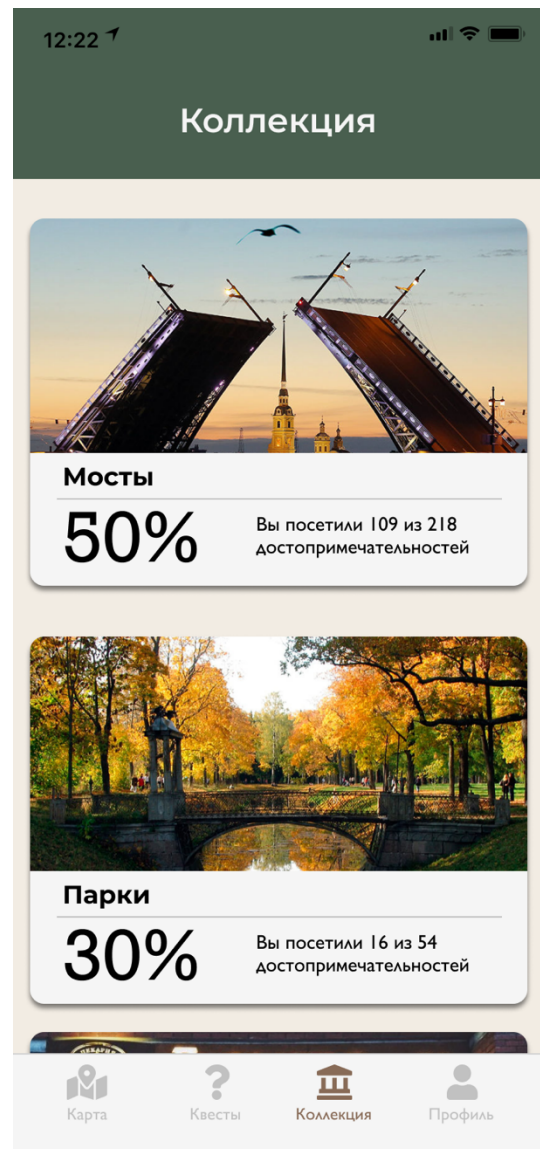


Рис. 6. Интерфейс раздела «Коллекция»



Рис. 7. Интерфейс страницы достопримечательности



Рис. 8. Интерфейс маршрута до метки

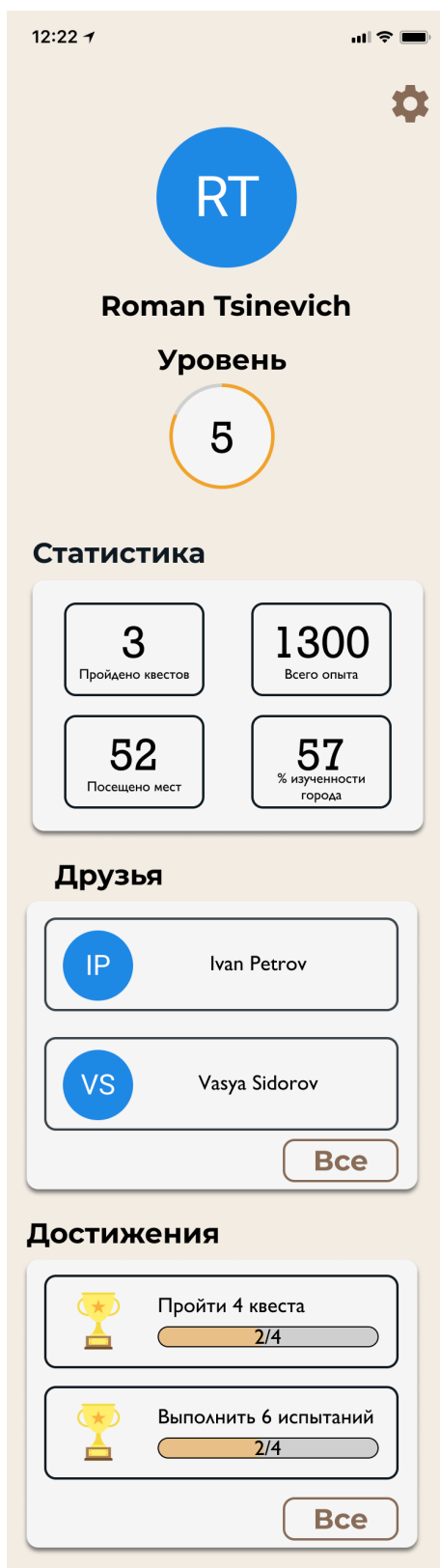


Рис. 9. Интерфейс раздела «Профиль»

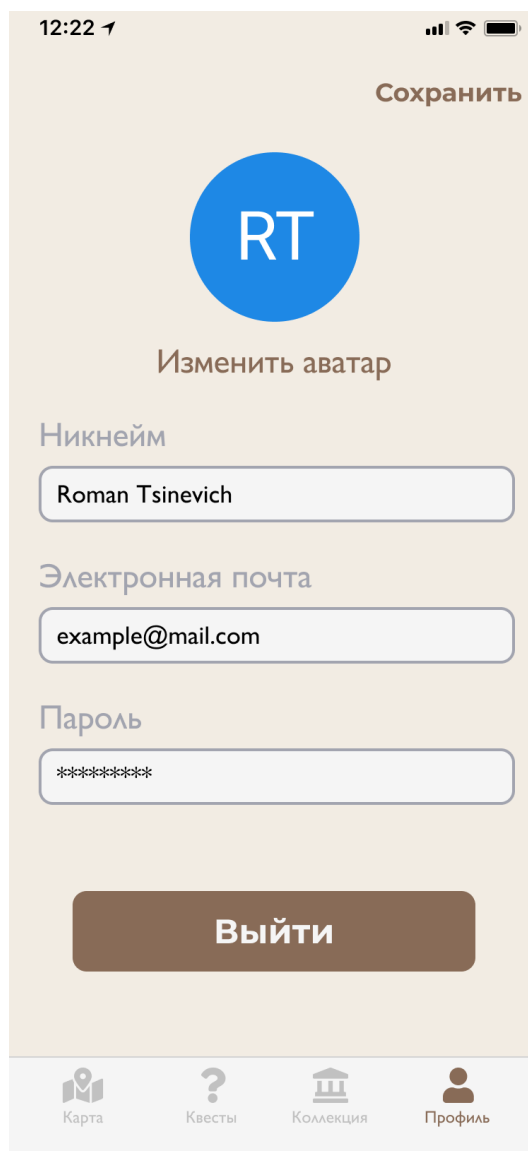


Рис. 10. Интерфейс настроек профиля