



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент программной инженерии и искусственного интеллекта

Бондаренко Денис Сергеевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Бакалаврская работа

ГИД ПО ДОСТОПРЕМЕЧАТЕЛЬНОСТЯМ ВЛАДИВОСТОКА.

по направлению подготовки бакалавров 02.03.03 Математическое обеспечение и
администрирование информационных систем

Владивосток

2023

В материалах данной выпускной квалификационной работы не содержатся сведения, составляющие государственную тайну, и сведения, подлежащие экспортному контролю

Уполномоченный по экспортному контролю

подпись И.О. Фамилия

«_____» _____ 20__ г.

Защищена в ГЭК с оценкой

Секретарь ГЭК

подпись И.О. Фамилия

«_____» _____ 20__ г.

Автор работы _____,
п
одпись

группа

Б9119-02.03.03техпро

«_____» 2023 г.

Руководитель ВКР
профессор департамента ПИиИИ,

д.т.н., профессор
(должность, ученое звание)

Артемяева И. Л.
подпись И.О. Фамилия

«_» _____ 2023 г.

«Допустить к защите»

И.о. директора Департамента

К.Т.Н,

О.А Крестникова
подпись И.О. Фамилия

«_____»
_____ 2023 г.

Оглавление

Введение	5
Глава 1 Обзор состояния области.....	7
1. Анализ предметной области	7
2. Подходы к созданию систем для хранения и предоставления информации.....	10
3. Обзор существующих систем	10
Выводы обзора	13
Глава 2 Анализ предметной области и построение модели	14
2.1 Термины предметной области:	14
2.2 Анализ предметной области	15
2.2.1 Объекты города	16
2.2.2 Пользователи системы	17
2.2.3 Карта	18
2.2.4 Маршрут	18
2.3 Анализ задач	19

2.4	Анализ	ситуаций	предметной	области	20
2.5	Модель		предметной	области	21
2.5.1	Определение		вспомогательных	терминов	21
2.5.2	Онтология			знаний	21
2.5.3	Ограничения		целостности	знаний	23
2.5.4	Онтология			действительности	23
2.5.5	Ограничения		целостности	действительности	25
2.5.6	Связи	онтологии	действительности	и онтологии	знаний
					25
2.6	Фрагмент		модели	знаний	26
Глава 3 Проект программной системы «Гид по Владивостоку»					27
3.2	Архитектурно-контекстная			диаграмма	29
3.3	Варианты		использования	системы	33
Рисунок 6 - диаграмма вариантов использования туристом					34
3.4	Описание			функции	34

3.4.1	Описание	функций	серверной	части	34
3.4.2	Спецификация	функций	для	Серверной	части
					35
3.4.3	Описание	функции	клиентской	части	36
3.4.4	Спецификация	функции	для	клиентской	части
					37
3.5	Описания	содержания		информации	38
3.6	Сценарий	диалога	программной	системы	с администратором
					40
3.7	Сценарий	диалога	программной	системы	с туристом
					41
	«Запрос на обновление данных к серверу»				41
	«Построение маршрута»				42
3.8.1.1	Проект	интерфейса		клиента	43
Глава 4 Реализация и тестирования.....					54
4.2	Реализация	клиентской		части	54
4.3	Характеристика			прототипа	54
4.4	Тестирование		и	эксперименты	54
4.4.2	Цель	экспериментальных		исследований	54

4.4.3Описание среды выполнения тестовых сценариев и проведение экспериментов	54
4.4.4Описание тестовых ситуаций и результаты тестирования	54
Заключение	55
Список литературы.....	56

Введение

Город Владивосток богат своей историей, практически каждый город имеет свою уникальную и интересную историю [1][2], свои достопримечательности, свои памятники, музеи, театры, кафе и так далее. Разнообразие типов и количества таких мест требуют хранения этих данных в информационных источниках, чтобы люди могли пользоваться этими знаниями. Конечно, существуют сайты, книги, приложения, которые содержат информацию о городе, его достопримечательностях и т. д. Но проблема заключается в том, что не обо всех городах эта информация находится в доступных источниках, часто она представлена в разных источниках, что затрудняет человеку поиск нужной информации. Если эта информация представлена в книгах, то она может очень быстро стать не актуальной. В действительности может случиться что угодно, и все может меняться так быстро, что новые издания книг просто не будут предоставлять достоверную информацию. Конечно, для нахождения актуальной информации существуют интернет-сайты, на которых информация может обновляться по мере ее изменения или добавления новой. Но не всегда и не везде есть доступ в интернет, чтобы просматривать эти страницы. А сохранение сайта в кэш не гарантирует, что функционал сайта не будет утерян, следовательно, часть информации может стать не доступной. Также проводятся различные экскурсии по достопримечательностям, однако существуют такие туристы, которые любят путешествовать самостоятельно, не прибегая к услугам экскурсоводов. Однако при самостоятельном путешествии часто возникает вопрос: как добраться до того или иного места? Для этого уже существует множество сервисов, которые помогают проложить маршрут из точки А в точку Б, однако нет сервисов, которые бы могли построить маршрут из нескольких таких точек. Например, турист захотел сначала прогуляться до парка, потом завернуть в кафе перекусить, а затем сходить в музей. При этом ему бы сообщалось, сколько времени примерно уйдет на этот маршрут, чтобы он мог разумнее распоряжаться своим временем. Учитывая указанные проблемы,

необходимо систематизировать все данные по Владивостоку и создать персональный гид для туристов – приложение. Оно бы помогало легко получить информацию с использованием смартфонов. Приложение позволяет сохранять всю информацию о Владивостоке на устройстве при наличии доступа к сети интернет. В дальнейшем просмотр информации происходит в любое время и в любом месте без доступа к сети интернет. При наличии доступа к сети интернет, на основе определенных критериев, указанных туристом, система должна прокладывать маршрут и отображать его на карте мира, сообщая затрачиваемое время на данный маршрут.

У систем гида можно выявить следующие общие проблемы, которые должны быть решены при разработке программного средства для гида по достопримечательностям:

- 1) область приложения системы;
- 2) время работы систем чрезмерно велико;
- 3) несовместимость систем гидов между собой, вследствие того, что системы с единым функциональным назначением имеют различные модели, интерфейсы, программные и аппаратные средства реализации;
- 4) система построена на основе интернет-технологий, т.е. является облачным сервисом;
- 5) отсутствие объяснения процесса исправления дефекта (неисправности, аномалии и т.д.);
- 6) отсутствие у эксперта возможности модифицировать систему достопримечательностей.

Цель выпускной квалификационной работы состоит в разработке прототипа гида по достопримечательностям Владивостока, позволяющего туристам, используя мобильное приложение, просматривать информацию о городе и на ее основе строить маршрут.

Для достижения поставленной цели необходимо решить следующие

задачи:

1. проанализировать существующие программные системы для туристов;
2. выполнить анализ предметной области «Гид по достопримечательностям Владивостока»,
построить ее модель, определить задачи, которые должно решать приложение, дать их содержательную и формальную постановку, а также разработать методы решения задач;
3. разработать проект системы «Гид по достопримечательностям Владивостока»;
4. реализовать прототип гида по достопримечательностям Владивостока и провести его тестирование и испытания.

Глава 1 Обзор состояния области

1. Анализ предметной области

В наше время в Приморском крае все активнее развивается туризм, создаются специальные условия для этого, например, такие как создание смотровых зоны, от правительства выдвигаются программы для развития туризма в Приморском крае [4]. Это привлекает в наш край туристов из разных городов России и других стран, которые, возможно, захотят познакомиться с культурой Приморья более глубоко.

Поэтому возникает задача по предоставлению сервисов для туристов, позволяющих им наиболее эффективно искать места для посещения, получать информацию о пути к различным объектам.

Существует несколько видов сервисов, которые могут предоставлять данную информацию: книжные издания, интернет сайты, приложения.

Книжные издания в наше время как источник информации для туриста безнадежно устарел, потому что книги занимают место в багаже, имеют ощутимый вес и имеют ряд недостатков по сравнению с другими способами получения информации, как например, устаревание информации.

Интернет -сайты более верное решение для туриста, т.к. позволяют просматривать информацию в актуальном виде, отслеживать какие -либо изменения в реальном времени, например, режим работы магазина, и использовать дополнительные функции для взаимодействия с информацией, например функцию прокладки маршрута. Недостатком является лишь необходимость быть подключенным к сети интернет. А если вы турист и приехали в другую страну, то доступ в интернет чаще всего вы можете иметь только в публичных местах, таких как аэропорт, отель, кафе и т.д.

Приложения имеют те же преимущества что и сайты, но, так же, как и сайты в основном зависят от сети интернет. Сейчас все чаще стали появляться приложения, которым нужна связь с сетью лишь для первоначального запуска, в дальнейшем они могут работать практически автономно, используя сеть только для дополнительных функций.

Выделим основные критерии для оценивания сервисов для туриста по предоставлению и взаимодействию с информацией о городах:

1. Легкость – имеет ли средство получения информации физический вес
2. Независимость от доступа к сети интернет – может ли сервис функционировать без подключения к сети интернет, либо предоставлять большую часть функциональности
3. Возможность взаимодействовать с информацией – может ли сервис предоставлять дополнительные функции для работы с информацией.

Результаты сравнения представлены в
таблице 1. Таблица 1 - Сравнение сервисов

Критерии оценки	Книжные издания	Интернет сайты	Приложения
Легкость	-	+	+
Независимос ть от доступа к сети интернет	+	-	+
Возможность взаимодействовать с информацией	-	+	+

Как мы видим из таблицы, приложение - наиболее универсальное средство для предоставления информации о городах и поддержки взаимодействия с этой информацией. Поэтому в современном мире это решение для туристов является наиболее подходящим и удобным.

В развитых странах практически у каждого человека есть собственное мобильное устройство. Доля смартфонов на мировом рынке 82%, остальные 18% принадлежат обычным мобильным телефонам [3]. Как можно заметить, каждый второй пользователь мобильных устройств имеет смартфон, а значит, он имеет возможность пользоваться некоторыми возможностями, которые предоставляет им такое устройство, а именно доступ в интернет и возможность устанавливать сторонние приложения.

Поэтому можно сделать вывод, что приложение должно быть для персональных мобильных устройств – смартфонов.

Система гида по достопримечательностям Владивостока должна уметь решать следующие задачи:

1. Хранение данных о городе
2. Передача хранимых данных в приложение на смартфоне
3. Прокладка маршрутов

2. Подходы к созданию систем для хранения и предоставления информации

Клиент. Данный подход подразумевает наличие приложения и связанной с ним базы данных. Достоинство данного подхода в простоте реализации. Недостатки – для того, чтобы произвести какие-то изменения в данных глобально (на нескольких устройствах где установлено приложение) необходимо создавать новую версию приложения с внесёнными изменениями и переустанавливать приложения на устройстве.

Клиент – сервер. Данный подход подразумевает наличие двух подсистем клиента и сервера. Сервер – подсистема, где хранится основная для системы информация, имеется собственный интерфейс для взаимодействия с ней. Клиент – подсистема, которая связана с сервером и может получать данные, хранящиеся на сервере. Достоинства – возможность изменять данные на сервере и получать их на клиенте. Недостатки – сложность разработки.

Так как в разрабатываемой системе будет храниться множество информации о городе, то должна быть возможность добавлять новые, то целесообразно применить архитектуру приложения «клиент-сервер»

3. Обзор существующих систем

Оценивание существующих систем для туристов будет производиться по следующим критериям:

1. Возможность просматривать объекты на карте мира с помощью приложения без использования постоянного доступа к сети интернет;
2. С помощью приложения строить маршруты из нескольких заданных объектов;
3. Ориентированность на туристов – большее количество информации, которая может быть интересна туристу;
4. Ориентированность на туризм во Владивостоке – более подробная информация для туризма

Для обзора существующих разработок в области приложений для туристов во Владивостоке выбраны следующие системы

5. Карты Google – приложение, построенное на основе бесплатного картографического сервиса и технологии, предоставляемых компанией Google. Сервис представляет собой карту и спутниковые снимки планеты Земля. Для многих регионов доступны высоко детализированные аэрофотоснимки. С сервисом интегрирован бизнес- справочник и карта автомобильных дорог с поиском маршрутов, охватывающая США, Канаду, Японию, Россию, Гонконг, Китай, Великобританию, Ирландию и некоторые районы Европы [6].
6. ForeverMap 2 – Мобильное приложение под систему IOS. Подробная карта мира на основе данных OSM. Имеется информативный путеводитель [7].

7. AroundMe - Мобильное приложение под систему IOS. Позволяет быстро узнать информацию об окрестностях вокруг. [8]

8. TripAdvisor – Мобильное приложение для систем IOS и Android. Позволяет просматривать различные данные об объектах города, читать отзывы других пользователей. [9]

Возможность просматривать объекты на карте мира с помощью приложения без использования постоянного доступа к сети интернет, либо просмотр большей части информации

1. Карты Google – позволяет загружать участки местности на внутреннее хранилище устройства
2. ForeverMap 2 – позволяет пользоваться всеми функциями как онлайн так и офлайн
3. AroundMe - нет возможности просмотра объектов на карте без доступа к сети интернет
4. TripAdvisor - нет возможности просмотра объектов на карте без доступа к сети интернет

С помощью приложения строить маршруты из нескольких заданных объектов

1. Карты Google – приложение строит маршрут только между двумя точками
2. ForeverMap 2 - приложение строит маршрут только между двумя точками
3. AroundMe – приложение использует сторонние приложения для построения маршрута
4. TripAdvisor – приложение использует сторонние приложения для построения маршрута

Ориентированность на туристов – большее количество информации, которая может быть интересна туристу

1. Карты Google – приложение является по большей части справочником, и не акцентирует внимание на туристах
2. ForeverMap 2 – приложение создано специально для туризма, и имеет множество функции которые будут полезны в путешествии
3. AroundMe – приложение является по больше части справочником, и не акцентирует внимание на туристах
4. TripAdvisor – приложение создано специально для туризма, и имеет множество функции которые будут полезны в путешествии

Ориентированность на туризм во Владивостоке – более подробная информация для туризма

1. Карты Google – приложение рассчитано на большой охват и нет ориентированности на определенной местности
2. ForeverMap 2 – приложение рассчитано на большой охват и нет ориентированности на определенной местности
3. AroundMe – приложение рассчитано на большой охват и нет ориентированности на определенной местности
4. TripAdvisor – приложение рассчитано на большой охват и нет ориентированности на определенной местности

Выводы обзора

Результаты обзора существующих систем для туристов представлены втаблице 2

Таблица 2 - Сравнение систем для туризма

Критерии оценки	Карты Google	Map 2 Forever	AroundMe	Trip Advisor
Возможность просматривать объекты на карте мира с помощью приложения без использования постоянного доступа к сети интернет, либо просмотр большей части информации	+	+	-	-
С помощью приложения строить маршруты из нескольких заданных объектов	-	-	-	-
Ориентированность на туристов – большее количество информации, которая может быть интересна туристу	-	+	-	+
Критерии оценки	Карты Google	Map 2 Forever	AroundMe	Trip Advisor
Ориентированность на туризм во Владивостоке – более	-	-	-	-

подробная информация для туризма				
----------------------------------	--	--	--	--

Как видно из приведённой выше таблицы, ни одна система не удовлетворяет абсолютно всем критериям, поэтому актуальная разработка системы для туристов - гида по Владивостоку, включающего все пункты.

Глава 2 Анализ предметной области и построение модели

Данная глава содержит результаты анализа предметной области, решаемых задач и методов их решения. На основе анализа производится построение математической модели предметной области и методов решения задач. Математическая модель представлена в виде модели онтологии.

Анализ предметной области выполнялся в соответствии с методологией, описанной в работе [5].

2.1 Термины предметной области:

1. Памятники города – название памятника
2. Достопримечательности города – название достопримечательности.
3. Местоположение – координата текущего местоположения пользователя на карте мира.
4. Точка назначения – координата объекта, которого необходимо достичь
5. Координата – пара чисел $[X, Y]$ $X, Y \in Q$, где X - широта, Y - долгота.

2.2 Анализ предметной области

Предметная область: гид по городам.

Для удобства разделим предметную область на 4 части и разберем каждую по отдельности.

2.2.1 Объекты города

В данной части предметной области решается задача определения объекта города.

Объектом задачи является объект города. Объект города обладает следующими характеристиками:

1. Город – значение Владивосток
2. Название – строка, обозначает имя (название) объекта
3. Текст – строка, обозначает текстовое описание объекта
4. Координата – пара чисел $[X,Y]$ $X,Y \in Q$, где X - широта , Y - долгота.
5. Идентификационный номер – число, которое однозначно идентифицирует объект города.

Пример: необходимо сохранить следующую информацию об объекте города: В городе Владивостоке по адресу ул. Светланская 123, находится замечательный сквер, который называется «Жариковский сквер ». В центре сквера установлен памятник, этот памятник был установлен в память жертвам американских воздушных пиратов - пассажиру и экипажам самолета ИЛ-12, сбитого над территорией КНР, выполнявшего рейс по международному коридору во Владивосток

1. Город: Владивосток
2. Название: Памятник в память жертвам американских воздушных пиратов
3. Текст: Этот памятник установлен в центре «Жариковского сквера ». Памятник установлен в память жертвам американских воздушных

пиратов - пассажиру и экипажам самолета ИЛ-12, сбитого над территорией КНР, выполнявшего рейс по международному коридору во Владивосток

5. Координата: [43.11405391, 131.91164351]

6. Дата добавления: 2016-04-11,15:34

2.2.2 Пользователи системы

Выделяется два типа пользователя: **Турист** и **Администратор**:

Турист: Турист должен иметь возможность сохранять информацию из системы на персональное мобильное устройство (Смартфон, Планшет) и просматривать его с помощью приложения. Турист может просматривать информацию на карте мира. Турист может посылать запрос системе для обновления информации о городах. Турист может определять свой тип туриста и на его основе система предлагает различные маршруты, учитывая информацию о наличии и длительности свободного времени.

Администратор: он имеет доступ ко всем данным, может просматривать их, изменять, удалять. Администратор может добавлять информацию о городе. Администратор осуществляет взаимодействие с системой через программу на компьютере.

Так как у администратора и у туриста разные способы взаимодействия с системой (у туриста мобильное устройство, у администратора компьютерная программа), то для них информация об объектах города может быть различной в одно и то же время. Связано с тем, что для администратора данные в системе всегда актуальные, а у туриста данные могут устаревать со временем.

2.2.3 Карта

Карта представляет собой матрицу, где каждый элемент – координата, где $X \in [-85.0 .. 85.0]$ широта, $Y \in [-180.0 .. 180.0]$ - долгота.

Матрица строится по следующим правилам:

Строки принимают значения: $[85.0, -180.0]$, $[85.0, -179.99999999]..$

$[85.0, 179.99999999], [85.0, 180.0]$. Значения могут быть с точностью до 10-ти миллионной.

Столбцы принимают значения: $[85.0, -180.0], [85.0, -180.0] ... 85.0, -180.0, 85.0, -180.0$

Представление карты:

Таблица 3 - Представление карты мира

85.00000000, - 180.00000000	85. 00000000,180. 00000000
...
-85.00000000, - 180.00000000	-85. 00000000,- 180. 00000000

2.2.4 Маршрут

Маршрут – это множество объектов города, соединённых ломаными линиями на карте. Маршрут строится на основе указанных предпочитаемых мест для конкретного пользователя.

Среднее пребывание на различных объектах (Часы: Минута):

1. Театр – 2:30

2. Музей – 1:00
3. Памятник - 0:15
4. Видовая площадка - 0:25.
5. Парк - 0:30
6. Клуб - 1:00
7. Исторические места - 0:30

Среднее пребывание на различных типах объекта необходимо для того, чтобы рассчитать, сколько времени займет тот или иной маршрут, и сообщить пользователю эту информацию, чтобы он мог планировать свои действия и свое время заранее.

2.3 Анализ задач

Задачи хранения информации.

1. Хранение объектов города

Задача состоит в хранении данных на двух подсистемах: клиенте и сервере.

Описание хранимой информации – множество объектов города

Задачи передачи информации.

1. Добавление информации об объекте города

Задача состоит в добавлении новой информации о городе на сервер. Отправитель — администратор,

получатель — хранилище данных
сервера.

Описание передаваемой информации – объект города

Пример: администратор добавляет новую информацию о городе на сервер.

2. Передача информации о городах пользователю.

Задача состоит в передачи информации, хранимой на сервере для клиента (туриста).

Отправитель – сервер, получатель – клиент.

Описание передаваемой информации - множество объектов города. Пример: сервер передает данные о городах туристу, который запросил обновление данных.

Задачи обработки информации.

1. Прокладка маршрута

- a. Известная информация: Множество объектов города и свободное время на основе введенных данных от пользователя.
- b. Неизвестная информация: Маршрут на карте или сообщение
- c. Содержательная постановка задачи: Пользователь указывает, какие места он хотел бы посетить. Система пытается строить маршрут относительно введенных объектов. Если маршрут успешно построен, то складывается время, которое необходимо затратить на весь маршрут (пешим шагом) + среднее время на пребывании на всех объектах, указанным пользователем. Полученное время отображается пользователю. Если маршрут не удалось построить, то выводится предупреждающее сообщение о неудаче.

2.4 Анализ ситуаций предметной области

1. Добавить информацию о новом объекте города

Необходимо предоставить возможность ввести данные и сохранить новую информацию в системе.

2. Пользователь турист, впервые запустил приложение.

Необходимо предоставить всю доступную информацию, хранящуюся в системе на текущий момент времени.

3. В предметной области ввели новую характеристику (Церкви)

Необходимо добавить данный тип к общему доступному множеству типов.

4. Турист указал данные для получения маршрута

Необходимо либо отобразить маршрут на карте, либо сообщить, что маршрут по указанным параметрам не может быть построен, либо ему не хватит времени на маршрут.

5. Турист решил обновить данные на клиенте

Необходимо предоставить возможность послать запрос на сервер для обновления данных. При отсутствии подключения к сети интернет сообщить, что отсутствует интернет-подключение.

2.5 Модель предметной области

2.5.1 Определение вспомогательных терминов

1. даты == <число, месяц, год>

Термин «даты» обозначает кортеж представления даты

2. число == $I[1,31]$

Термин «число » - целое число от 1 до 31

3. месяц == $I[1,12]$

Термин «месяц » - целое число от 1 до 12

4. год== $I[2014,2114]$

Термин «год» - целое число от 2021 до 2114

2.5.2 Онтология знаний

1. Сорт города: $\{ \} N \setminus \emptyset$

Термин «города » обозначает непустое конечное множество названий городов

2. Сорт объекты города: города -> $\{ \} N \setminus \emptyset$

Термин «объекты города» обозначает функцию, сопоставляющую каждому городу непустое конечное множество объектов города

3. объекты $\equiv (U(\text{г: города}) \text{ объекты города (г)})$ (вспомогательный термин)

Термин «объекты» - вспомогательный термин, обозначающий множество всех объектов города

4. Сорт типы объектов: $\{ \} N \setminus \emptyset$

Термин «типы объектов» обозначает непустое конечное множество названий типов объектов

5. Сорт тип объекта: объекты \rightarrow типы объектов

Термин «объекты» - функция, сопоставляющая множеству объектов множество типов объектов

6. Сорт среднее время пребывания на типе объекта: тип объекта $\rightarrow (I[1,24] \times I[1,59])$

Термин «среднее время пребывания на типе объекта» - функция, сопоставляющая множеству типов объектов кортеж из двух натуральных чисел, означающий время в формате часы: минуты

7. Сорт дата внесения информации для города: $(\text{г} \rightarrow \text{города}, \text{о-} > \text{объекты города (г)}) \rightarrow \{ \} \text{даты}$

Термин «дата внесения информации для города» - функция, сопоставляющая дату для объекта города

8. Сорт координаты объекта: объекты $\rightarrow (I[0, \infty] \times I[0, \infty])$

Термин «координаты объекта» - функция, сопоставляющая множеству объектов кортеж из двух натуральных чисел

9. Сорт расстояние между объектами: (объекты X объекты $\rightarrow \{ \} R[0, \infty] \setminus \emptyset$)
Термин «расстояние между объектами» - функция, сопоставляющая двум объектам не пустое множество целых чисел

10. Сорт пользователи: $\{ \} N$

Термин «пользователи» обозначает конечное множество имен пользователей

11. Сорт города пользователей: пользователи $\rightarrow \{ \}$ город

Термин «город пользователей» - функция, сопоставляющая множеству пользователей название города

12. Сорт даты внесения информации пользователей: (п \rightarrow пользователи, город пользователей (п)) \rightarrow даты

Термин «даты внесения информации пользователей» - функция, сопоставляющая множеству городов пользователя дату внесения.

13. Сорт множество объектов пользователя: (п \rightarrow пользователи, г \rightarrow город пользователей (п), д \rightarrow даты внесения информации пользователей (г)) \rightarrow

$\{ \}$ объекты

Термин «множество объектов пользователя» - функция, сопоставляющая городу пользователя множество объектов.

14. предположительные места == типы объектов

15. Сорт предпочтительные места для туриста: пользователи \rightarrow $\{ \}$ предположительные места

Термин «предпочтительные места для туриста» - функция, сопоставляющая туристу множество предпочтительных мест

16. Сорт карта == ((V:объекты) ($<$ координаты объекта (V), высота над уровнем моря (V) $>$))

2.5.3 Ограничения целостности знаний

1. $(\forall v_1 \in \text{пользователи}) (\forall v_2 \in \text{город пользователей}(v_1)) (\forall v_3 \in \text{даты внесения информации пользователей}(v_1, v_2))$
множество объектов пользователя $(v_1, v_2, v_3) \in \text{объекты города}(v_2)$

Для города пользователя, для любой даты добавления объекты в городе пользователя принадлежат к объектам этого города

2.5.4 Онтология действительности

1. Сорт туристы: $\{ \} \text{пользователи} \setminus \emptyset$

Термин «туристы» - обозначает конечное непустое множество туристов (подмножество множества пользователи).

2. Сорт город туриста: $\{ \} \text{города} \setminus \emptyset$

Термин «город туриста» - обозначает конечное непустое множество названий городов туриста

3. Сорт тип туриста: туристы $\rightarrow \{ \} \text{типы туристов} \setminus \emptyset$

Термин «тип туриста» - функция, сопоставляющая туристам конечно непустое множество типов туристов

4. Сорт что предпочитает турист: туристы $\rightarrow \{ \} \text{предпочтительные места}$
Термин «что предпочитает турист» - функция, сопоставляющая туристу конечное множество предпочтительных мест

5. Сорт доступное время : туристы $\rightarrow (I [0,24] \times I [0,59])$

Термин «доступное время» - функция, сопоставляющая туристам кортеж из двух натуральных чисел означающий доступное время в формате часы: минуты

6. Сорт начальная точка: туристы $\rightarrow (I [0, \infty] \times I [0, \infty])$

Термин «начальная точка » - функция, сопоставляющая туристам кортеж их двух чисел, состоящий из географических координат в формате широта: долгота

7. Сорт посещённые места: турист $\rightarrow \{ \}$ карта

Термин «посещённые места » - обозначает функцию, сопоставляющую туристу конечное непустое множество названий посещённых мест

8. момент времени $== (I [1, 24] \times I [1, 60] \times I [1, 60])$

Множество кортежей, содержащих моменты времени в формате часы:минуты: секунды

9. Сорт координата местоположения: $(v1 \rightarrow \text{турист}, v \rightarrow \text{момент времени}(v1)) \rightarrow (I [0, \infty] \times I [0, \infty])$

Термин «координата местоположения» - обозначает функцию отображающую местоположение туриста в определённый момент времени

10. Сорт карта туриста: туристы $\rightarrow \{ \}$ карта

Термин «карта туриста » - обозначает функцию, которая сопоставляет туристу множество карт

11. Сорт маршрут туриста : seq карта туриста

Термин «маршрут туриста » - обозначает последовательность из некоторых географических точек.

2.5.5 Ограничения целостности действительности

1. ($\forall v1 \in \text{туристы}$)

($\forall v2 \in \text{город туриста}(v1)$)

Посещённые места ($v1$) \in объекты города ($v2$)

Посещенные места города туриста принадлежат к городу туриста

2.5.6 Связи онтологии действительности и онтологии знаний

2. ($\forall v1 \in \text{туристы}$)

что предпочитает турист $\sqsubset(v1)$ предпочтительные места для типатуриста (тип туриста ($v1$))

Для любого туриста, то что предпочитает этот турист является подмножеством предпочтительных мест для этого типа туриста

3. ($v1 \in \text{туристы}$)

($v3 \in \text{тип туриста}(v1)$)

($v4 \in \text{доступное время}(v1)$) ($v5 \in \text{карта туриста}(v1)$)

(\cup расстояние между объектами ($v5(\text{координаты объекта}(\text{объекты}))) \times v5(\text{координаты объекта}(\text{объекты}))) \leq$

средняя скорость типа туриста ($v3$) * доступное время ($v1$)

Для любого туриста любого типа, для любого доступного времени у данного туриста и на любой карте данного туриста общая дистанция маршрута туриста должна быть меньше или равна его

доступному времени, умноженному на среднюю скорость этого типатуриста

4. ($v_1 \in \text{туристы}$)

($v_5 \in \text{карта туриста}(v_1)$)

Начальная точка (v_1) == min seq (расстояние между объектами ($\cup v_5(\text{координаты объекта (объекты)})$) X (координата местоположения (момент времени (v_1)))

Маршрут выбирается так, что бы как можно ближе к туристу была начата прокладка маршрута

5. ($v1 \in$ туристы)

($v5 \in$ карта туриста ($v1$))

Если координата местоположения ($v1$) == $v5$ (координата объекта(объекты))

То, посещённые места = посещённые места + $v5$ (координата объекта(объекты))

Маршрут туриста ($v5$) = Маршрут туриста ($v5$) \ посещённые места
Если текущее местоположение туриста = положению объекта на карте туриста, то добавляем этот объект к множеству посещенных мест.

2.6 Фрагмент модели знаний

1. Город \equiv Владивосток
2. Объекты \equiv {Театр, Музей, Памятник, Видовая площадка, Парк, Клуб, Исторические места}
3. Среднее время пребывания на объекте (Театр) \equiv 2:30
4. Среднее время пребывания на объекте (Музей) \equiv 1:0
5. Среднее время пребывания на объекте (Памятник) \equiv 0:15
6. Среднее время пребывания на объекте (Клуб) 1:00
7. Среднее время пребывания на объекте (Исторические места) 0:30
8. Среднее время пребывания на объекте (Интересное место) 0:30
9. Среднее время пребывания на объекта (Парк) \equiv 0:30

Глава 3 Проект программной системы «Гид по Владивостоку»

3.1 Спецификация требований к системе

Функциональные требования:

Для туриста:

1. Система должна предоставлять возможность посылать запрос на обновление данных
2. Система должна уметь принимать данные об объектах города с сервера
3. Система должна уметь хранить данные об объектах города
4. Система должна предоставлять возможность просматривать данные
5. Система должна на основе введённых туристом данных уметь строить маршрут на карте мира

Для администратора:

1. Система должна уметь хранить данные о городе
2. Система должна предоставлять возможность добавлять данные

Нефункциональные требования:

Требования к производительности:

1. Добавление информации о городе, при наличии всех входных данных, должно занимать не более 30-ти секунд
2. Посылка запроса в систему на обновление данных о городе должна происходить не более 60-ти секунд.

Надёжность:

1. Все данные о городе дублируются в системе

Требования к времени хранения:

1. Клиентская часть должна хранить данные о городе до тех пор, пока турист не удалит приложение с телефона.

Ограничения и пределы свободы разработчика

1. Система должна быть создана с применением технологии «клиент-сервер »
2. Для реализации клиентской части был выбран язык программирования Kotlin. Kotlin — статически типизированный, объектно -ориентированный язык программирования, работающий поверх Java Virtual Machine.
3. Для реализации сервера была выбрана программная платформа Firebase
4. Для реализации карт мира были выбраны карты от Яндекс. Они соответствуют всем требованиям, указанным на этапе анализа предметной области.
5. Для реализации маршрутов был выбран ЯндексMapKit. Это позволяет прокладывать маршруты на карте мира по дорогам города.

Требования к переносимости:

1. Серверная часть должна работать на платформах семейства MacOS/Windows7 и выше.
2. Клиентская часть должна работать на устройствах семейства Android 8 и выше

3.2 Архитектурно-контекстная диаграмма

Ниже на рисунке 1 представлен нулевой уровень архитектурно-контекстной диаграммы гида по городам. При анализе предметной области были выделены 2 вида пользователей: администратор и турист, которые взаимодействуют с системой. Их функциональные возможности по отношению к системе представлены на рисунке.



Рисунок 1 - АКД 0-го уровня.

На рисунке 2 представлена АКД 1-го уровня. Она позволяет углубиться в устройство системы. Как мы видим, система состоит из двух подсистем

«Серверная часть » и «Клиентская часть», связанных между собой. Подсистема «Клиентская часть» имеет внешнюю связь с сервисом, который поддерживает карты города в актуальном виде:

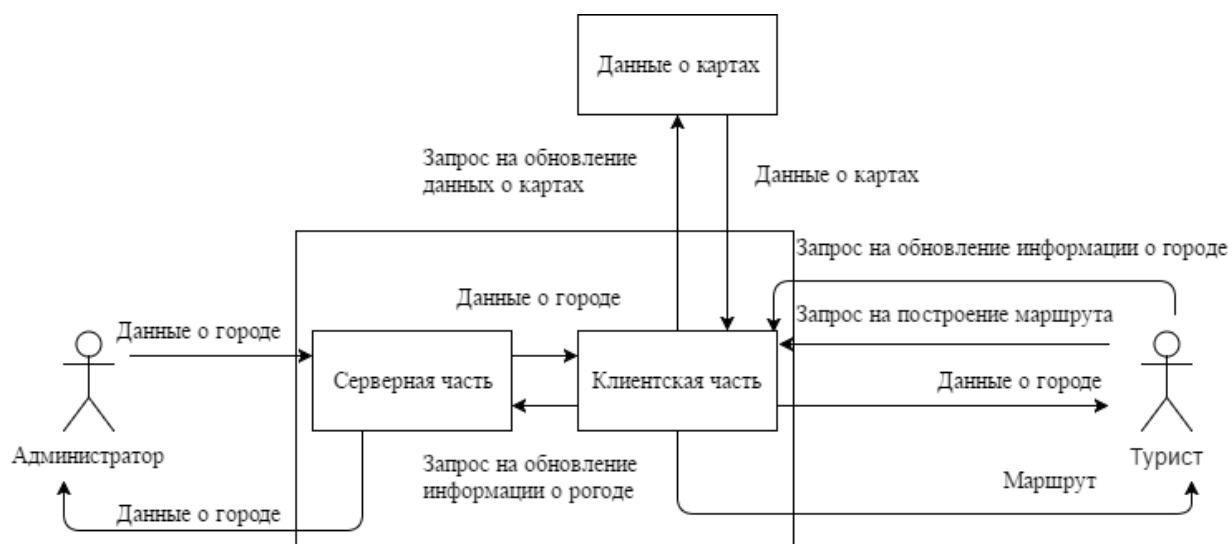


Рисунок 2 - АКД 1-го уровня

Разберем каждую подсистему по отдельности:

Описание серверной части:



Рисунок 3 - АКД серверной части

Серверная часть состоит из интерфейса, базы данных и обработчика данных

Интерфейс отображает все доступные функции для администратора, а именно: добавление данных о городе

Требования к серверной части:

1. Хранение данных о городе
2. Возможность вносить данные о городе
3. Возможность просматривать данные о городе

Описание Клиентской части:

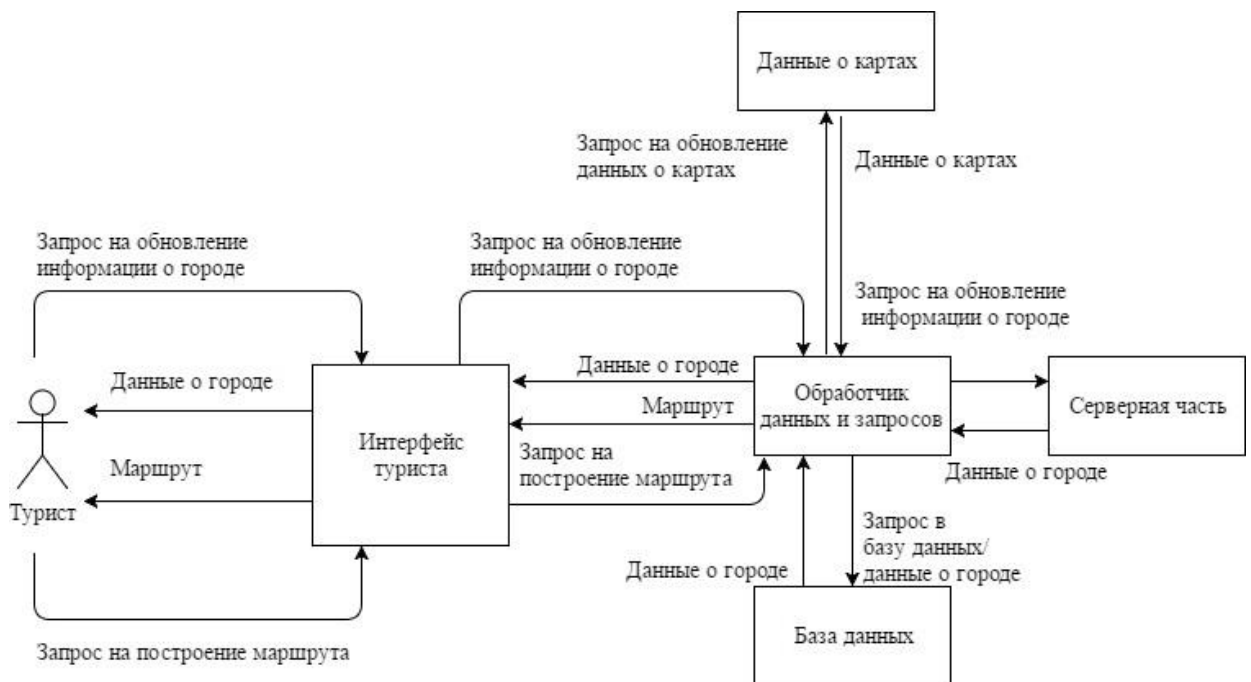


Рисунок 4 - АКД клиентской части

Клиентская часть также состоит из базы данных, интерфейса и обработчика данных и запросов. Так же она имеет связь с сервером карт, для возможности просматривать объекты города на актуальной карте местности

Требования к Клиентской части

1. Возможность посылать запрос к серверной части на обновление данных о городе.
2. Возможность строить маршрут
3. Хранение данных о городе
4. Возможность просматривать данные о городе на карте
5. Возможность иметь связь с сервером карт, для поддержания карт клиентской части в актуальном состоянии

Описание архитектурно-контекстной диаграммы:

1. Администратор – пользователь системы
2. Турист – пользователь системы
3. Данные о городе – информация о памятниках города, исторических места города, достопримечательностях города, интересных и необычных мест города и т.д. со своими атрибутами.
4. Запрос на обновление информации о городе – передача данных о городе.
5. База данных – структура, хранящая множество информации о городах.
6. Данные сайта – данные о городе, хранящиеся на внешнем ресурсе

7. Данные о картах – данные о соответствии отображаемых карт клиентской части действительности
8. Интерфейс администратора – программное средство, позволяющее с помощью визуального отображение на экран вносить, изменять и просматривать информацию о городе, позволяет подавать запросы.
9. Интерфейс туриста - программное средство, позволяющее с помощью визуального отображение на экран просматривать информацию о городе, позволяет подавать запросы.

3.3 Варианты использования системы

В данной системе можно выделить следующие группы пользователей:

1. Администратор – человек, который пополняет, изменяет и удаляет данные из базы данных, хранящей информацию огородах. Имеет возможность просматривать данные о городе.
2. Турист – человек, который может посылать запрос к серверу для обновления данных о городе. Может просматривать хранящуюся на персональном устройстве информацию о городах.

На рисунке 5 представлена диаграмма вариантов использования системы администратором:

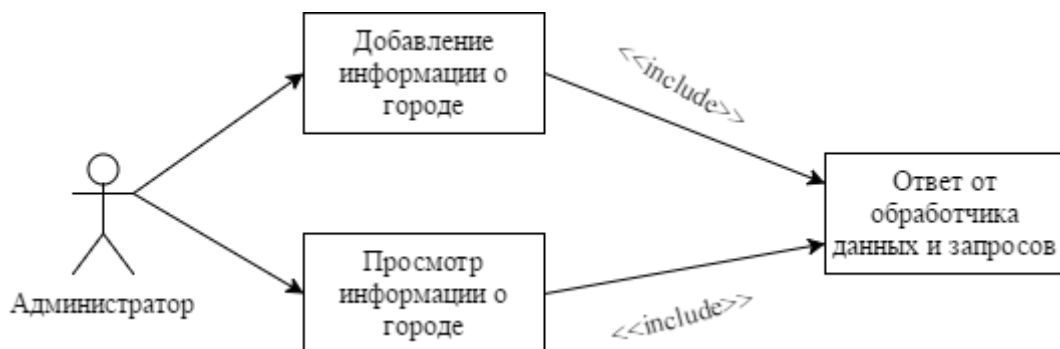


Рисунок 5 - диаграмма вариантов использования администратором

На рисунке 6 представлена диаграмма вариантов использования системы туристом

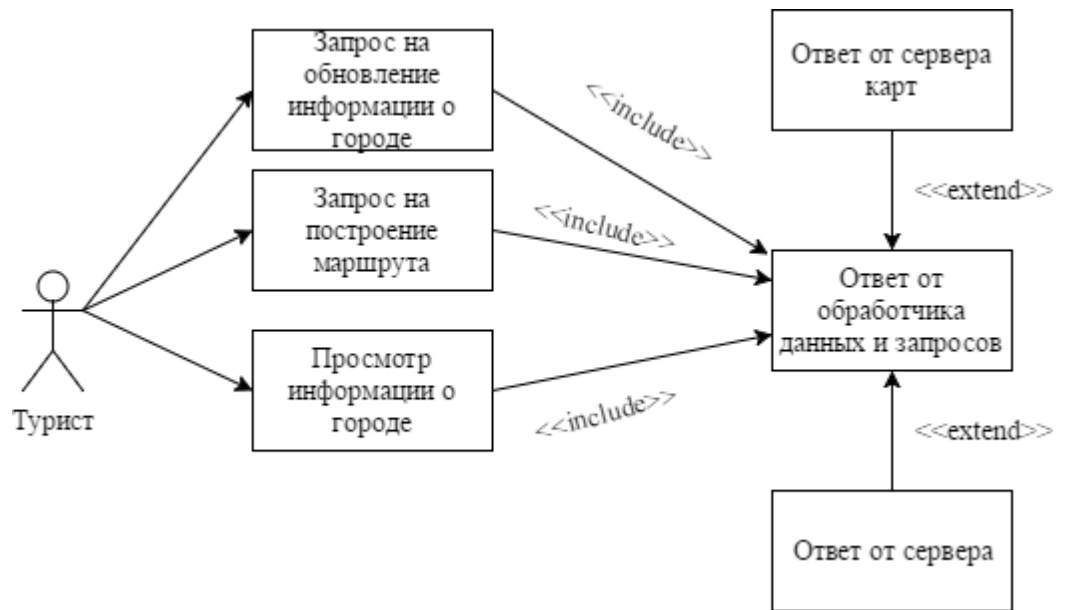


Рисунок 6 - диаграмма вариантов использования туристом

Описание диаграмм использование системы:

1. Администратор, турист – пользователи системы
2. Добавление информации о городе – администратор может добавлять информацию о городе
3. Просмотр информации о городе – администратор может просматривать информацию о городе.
4. Просмотр информации о городе - турист может просматривать информацию о городе
5. Обновление данных на устройстве - турист может посылать запрос на обновление данных о городах на устройстве.
6. Запрос на построение маршрута – турист может посылать запрос на построение маршрута

3.4 Описание функции

3.4.1 Описание функций серверной части

На рисунке 7 представлена горизонтальная декомпозиция функций серверной части программы

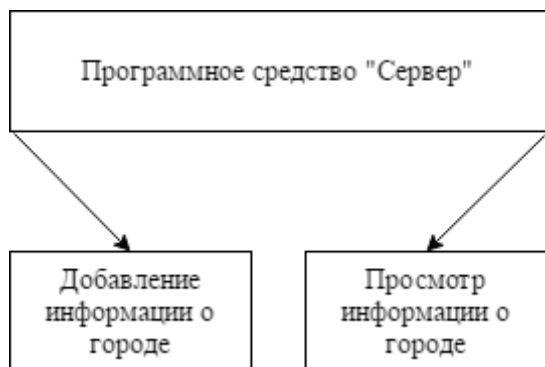


Рисунок 7 - Горизонтальная декомпозиция функций. Сервер

Описание:

1. Добавление информации о городе – на вход подается город, в который хотим добавить информацию, название объекта, описание этого объекта, его координата (месторасположение) и фотография.
2. Просмотр информации о городе – просмотр всего списка объектов города, хранящийся в системе.

3.4.2 Спецификация функций для Серверной части

1. Добавление информации о городе. Эта функция должна уметь выполнять действия:
 - 1.1. Сохранять данные в базу данных
 - 1.2. Входные данные: Новые данные о городе, база данных
 - 1.3. Выходные данные: Обновленная база данных сервера
2. Просмотр информации о городе. Эта функция должна уметь выполнять действия:

2.1. Отображать весь список объектов города

2.2. Входные данные:

2.2. Выходные данные: Данные о городе

3.4.3 Описание функции клиентской части

На рисунке 8 представлена горизонтальная декомпозиция функций клиентской части программы



Рисунок 8 - Горизонтальная декомпозиция функции. Клиент

Описание:

1. Запрос на обновление карт – на вход подается текущая версия карт у клиента, сервер сравнивает актуальную версию с заданной и, если есть более новая версия, то обновляет карты. Выходом функции является новая версия карт
2. Просмотр информации о городе. Турист – на вход подается город, в котором хотим просмотреть информацию.
3. Запрос на обновление данных к серверу – на вход подается дата последнего обращения на сервер. Функция сравнивает, какие данные на сервере старше данной даты. Выходным значением функции является множество данных старше входной даты.

4. Построение маршрута – на вход турист указывает, какие объекты в городе он хотел бы посетить. Турист выбирает сам объект. Система пытается построить маршрут относительно введенных объектов. Если маршрут успешно построен, то складывается время, которое необходимо затратить на весь маршрут (пешим шагом) + среднее время на пребывании на всех объектах указанным пользователем. Полученное время отображается пользователю. Если маршрут не удалось построить, то выводится сообщение, что маршрут не удалось построить. Выходными значениями является маршрут на карте мира, либо сообщение о невозможности построить маршрут

3.4.4 Спецификация функции для клиентской части

Запрос на обновление карт. Эта функция должна иметь следующий функционал:

1.1. Посылается запрос на заранее известный сервер

1.2.Сравнивать версию текущих карт и карт сервера

1.3.В случае если дата текущая меньше даты сервера то загрузить новые карты, иначе ничего не делать

1.4. Обработать карты для отображения на устройстве

1.5.Входные данные: Данные о городе

1.6.Выходные данные: Данные о городе или сообщение

2. Просмотр информации о городе. Эта функция должна иметь следующий функционал:

2.1.Выполнять поиск заданного объекта города в базе данных

2.2.Если найдено, то показывать информацию по найденному объекту, если не найдено, то сообщать, что такой объект не обнаружен.

2.3. Входные данные: Данные о городе

2.4. Выходные данные: Данные о городе или сообщение

3. Обновление данных на устройстве. Эта функция должна иметь следующий функционал:

3.1. Уметь посылать запрос по заранее заданному

URL 3.2. Принимать данные формата JSON

3.3. Обрабатывать эти данные

3.4. Удалять текущие хранимые данные на устройстве

3.5. Сохранять полученные данные в базу данных клиента

3.6. Входные данные: сервера

3.7. Выходные данные: Обновленная база данных

4. Построение маршрута. Эта функция должна иметь следующий функционал:

4.1. Строить маршрут на основе введенных объектов и доступного времени

4.2. Выдавать сообщение если нельзя построить маршрут на основе введенных данных.

4.3. Выдавать сообщение что не достаточно времени, если время на маршрут больше чем указанное время

4.4. Отображать маршрут на карте мира

4.5. Входные данные: доступное время туриста, объекты

4.6. Выходные данные: Маршрут на карте мира или сообщение

3.5 Описания содержания информации

В предметной области присутствуют следующие информационные объекты:

1. Объект города.

Атрибуты:

Город

- i. Название города, в котором находится объект
- ii. Поступает на вход от администратора при добавлении нового объекта города в базу данных.
- iii. Значение из множества {Владивосток}.
- iv. Используется для определения города, в котором находится объект.

b. Название

- i. Название объекта города.
- ii. Поступает на вход от администратора при добавлении нового объекта города в базу данных.
- iii. Последовательность символов длиной до 255.
- iv. Используется для определения названия объекта города.

c. Текст

- i. Текстовое описание объекта города.
- ii. Поступает на вход от администратора при добавлении нового объекта города в базу данных.
- iii. Последовательность символов до 500.
- iv. Используется для описания объекта города.

d. Координата -X

- i. Число, обозначающее широту.
 - ii. Поступает на вход от администратора при добавлении нового объекта города в базу данных.
 - iii. $X \in [-85.0..85.0]$.
 - iv. Используется для отображения объекта города на карте мира.
- e. Координата -Y
 - i. Число, обозначающее долготу.
 - ii. Поступает на вход от администратора при добавлении нового объекта города в базу данных.
 - iii. $Y \in [-85,0..85.0]$.
 - iv. Используется для отображения объекта города на карте мира.
- f. Идентификационный номер
 - i. Идентификационный номер объекта города в системе.
 - ii. Автоматически присваивается при занесении теории в базу данных.
 - iii. Натуральное число до 100000000.
 - iv. Используется для однозначной идентификации объекта города в системе.

2. Маршрут.

Атрибуты:

- a. Объекты города
 - i. Множество объектов города, которое хочет посетить турист
 - ii. Добавляется туристом, при выборе мест для маршрута.
 - iii. Значение элемента множества - объект города

- iv. Используется для построения маршрута по указанным объектам города

3.6 Сценарий диалога программной системы с администратором

Сценарий для задачи: «Добавление информации о городе »

1. Система отображает форму добавления данных
2. Администратор выбирает город
3. Администратор выбирает добавляемый объекта
4. Администратор заполняет все необходимые поля формы
5. Администратор нажимает сохранить
6. Система отображает добавленную информацию в виде JSON объекта
7. Если администратор хочет закончить добавление данных, то он закрывает окно.
8. Если администратор хочет добавить еще информацию, то он нажимает кнопку назад, и переходим к шагу 1.

Сценарий для задачи: «Просмотр информации о городе»

1. Администратор должен открыть папку Markers, в которой в каждом файле представлена вся информация об каждом объекте города хранящихся на сервере

3.7 Сценарий диалога программной системы с туристом

«Просмотр информации о городе»

1. Система отображает главное меню

2. Если на клиенте есть данные с сервера, то система отображает все данные о городе на карте в виде точек.
3. Турист может нажать на кнопку карта, и все объекты на карте будут скрыты, при повторном нажатии возвращает все объекты на карту.
4. Если на клиенте нет данных с сервера, то объекты не будут отображаться на карте.
5. Турист может нажать на интересующий его объект
6. Система отображает форму с информацией об этом объекте

«Запрос на обновление данных к серверу»

1. Система отображает главное меню
2. Турист выбирает меню настроек
3. Система отображает меню настроек
4. Система посылает запрос на сервер
5. При наличии соединения с интернетом сервер возвращает данные о городах, хранящиеся в системе. Иначе сообщает об отсутствии интернет -подключения.
6. Если сервер данных не доступен, то данные не будут обновлены на устройстве.

«Построение маршрута»

1. Система отображает главное меню.
2. Турист нажимает кнопку «Построить маршрут».
3. Система отображает окно со списком объектов для маршрутов, первоначально он пуст.
4. Турист выбирает добавить новый объект.

5. Система отображает окно со списком всех доступных типов объектов.
6. Турист выбирает интересующие его типы объектов.
7. Система отображает список со всем объектами выбранного типа.
8. Турист выбирает объект города и нажимает «Добавить в маршрут».
9. Турист может остаться на этом окне, либо вернуться на выбор типов объектов, либо завершить добавление.
10. Турист выбирает «Завершить добавление».
11. Система отображает список из объектов, добавленных ранее.
12. Турист нажимает построить маршрут.
13. При наличии соединения с интернетом система пытается строить маршрут относительно указанных объектов, иначе система сообщает об отсутствии подключения к сети интернет
14. Если маршрут удалось построить, то система отображает главное меню с картой, маршрутом на карте и информацией о необходимом времени на данных маршрут.
15. Если маршрут не удалось построить, то выводится сообщение об ошибке при построении маршрута

3.8.1.1 Проект интерфейса клиента

На рисунке 19 представлено меню смартфона vivo y31, работающего под управлением операционной системы Android 13. Что бы запустить приложение нужно нажать на иконку с приложением под названием GuideToVladivostok.

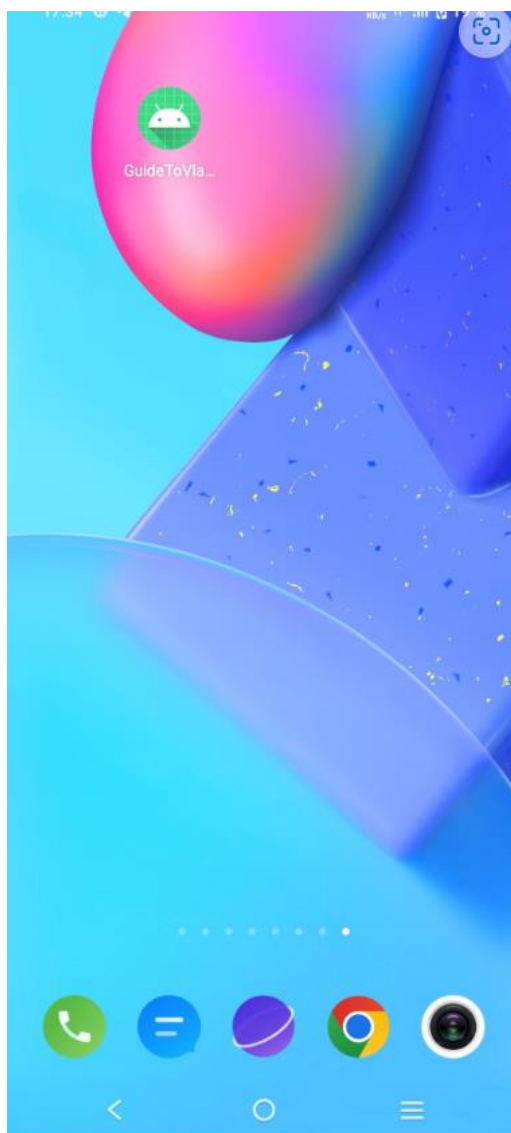


Рисунок 19 - Запуск приложения со смартфона

При первом запуске, рисунок 20, программа попросит дать доступ к геопозиции, это нужно для того, чтобы определить местоположение пользователя на карте. Рекомендуется разрешить доступ, для использования дополнительного функционала приложения, а именно центрирование карты на пользователе.

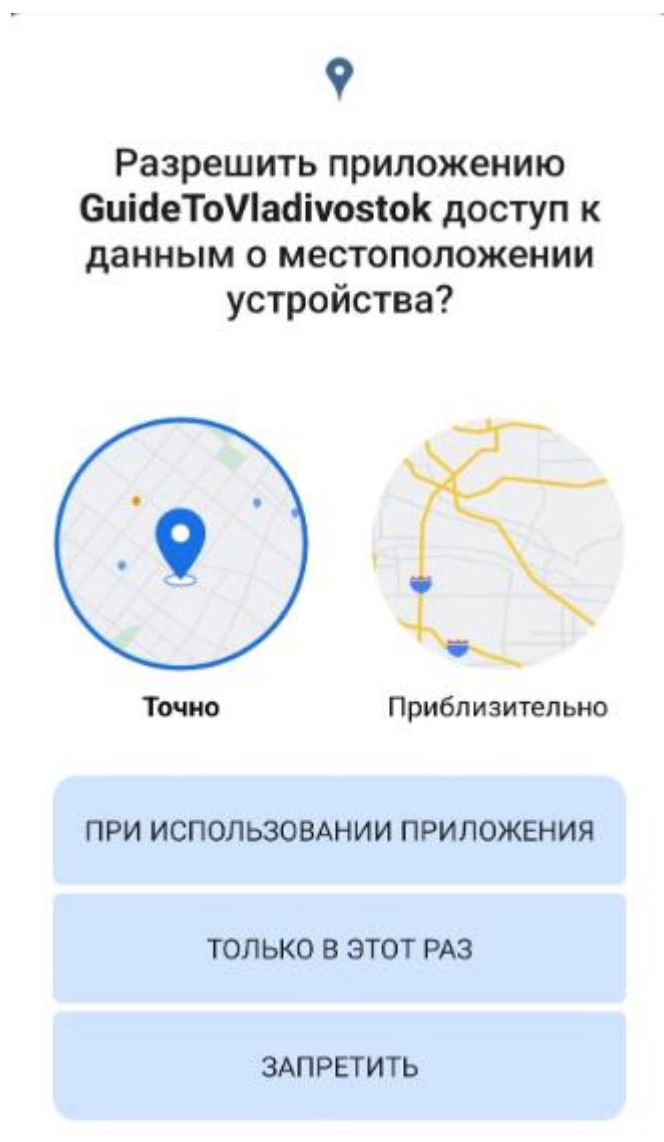


Рисунок 20 - Первый запуск приложения

В случае если вы запретили доступ, то его можно дать через настройки самого смартфона. Перейдите в: Настройки -> Приложения -> GuideToVladivostok -> Разрешения и выберите Разрешить доступ к местоположению -> Разрешить только во время использования приложения

После того как вы разрешите доступ приложению к местоположению, рисунок 21, то отобразится карта мира, центрированная на пользователе.

1. Карта мира. На ней отображается ваше текущее местоположение, будут отображаться объекты города и маршруты
2. Список локаций.
3. Показ пробок в городе
4. Отобразить отели на карте. При нажатии, на карте будут отображаться отели города, находящиеся рядом. При повторном нажатии они будут скрываться.
5. О программе. Открывает окно о программе и указано, что

данная программа используется в учебных целях маршрута

На рисунке 22 изображен список локаций.

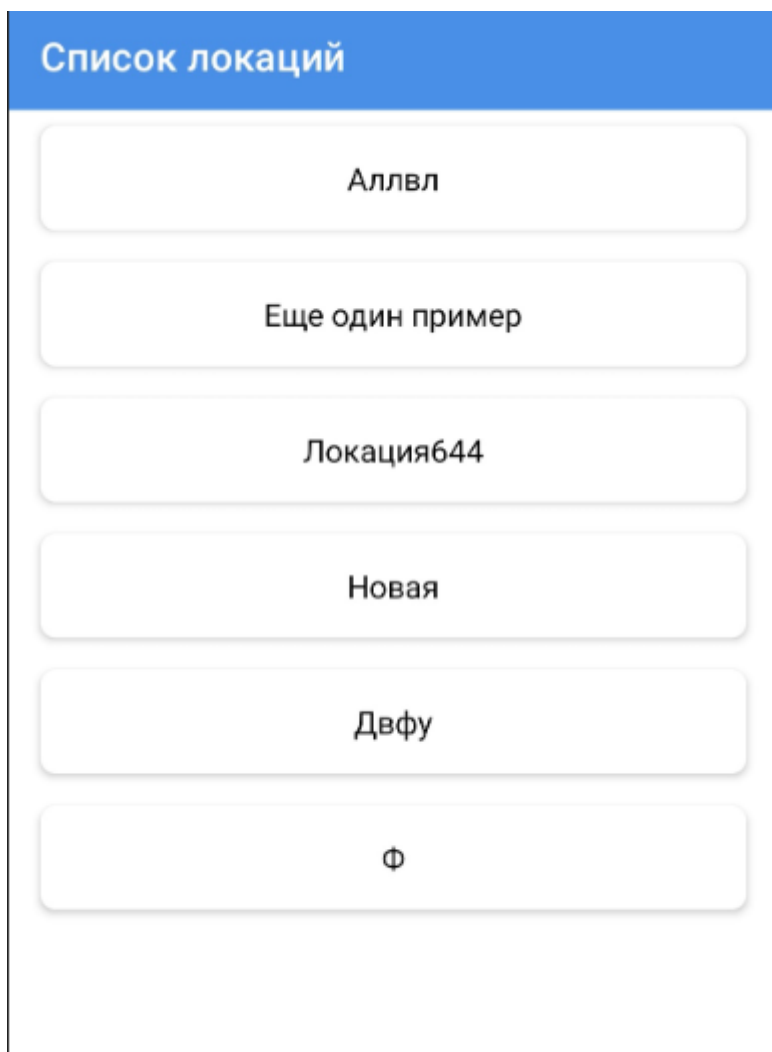


Рисунок 22 - Меню списка локаций

При нажатии кнопки назад, отобразится главное меню.

При нажатии кнопки «Пробки» откроется меню обновления на карте отобразятся пробки в городе, рисунок 23

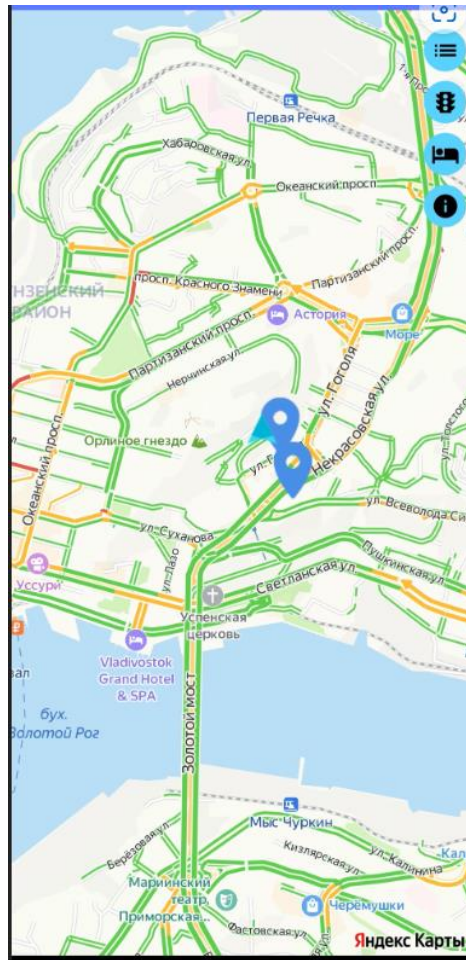


Рисунок 23 – Пробки в городе

При нажатии кнопки назад, отобразится меню настроек приложения

Если устройство не будет подключено к сети интернет, то будет выведено сообщение об ошибке, рисунок 24.

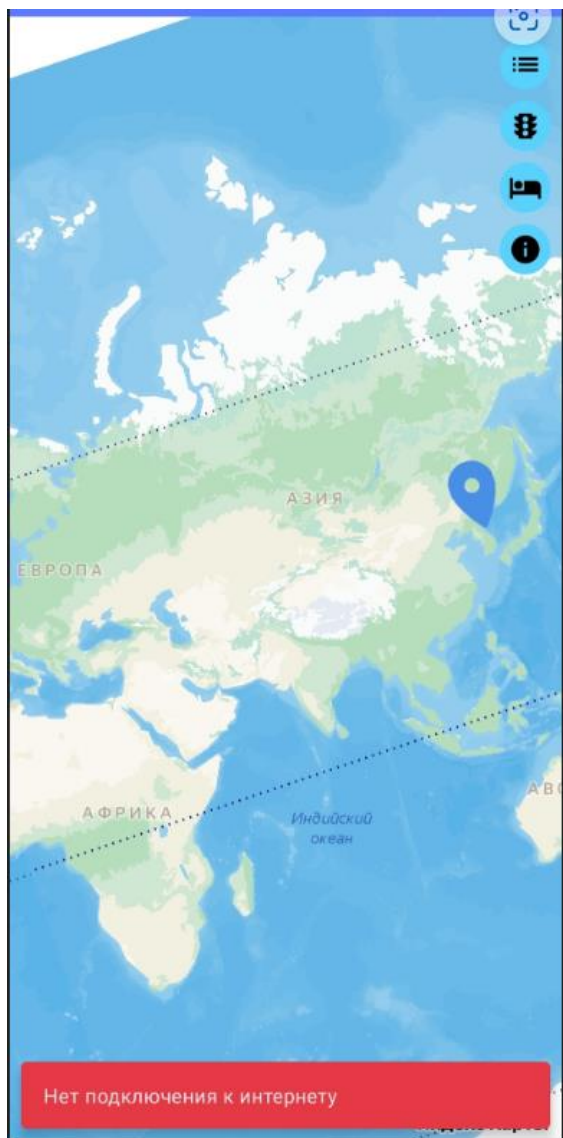


Рисунок 24 – Нет подключения к интернету

После успешного получения данных с сервера, вернувшись в главное меню на карте отобразятся все объекты, полученные с сервера, рисунок 25.

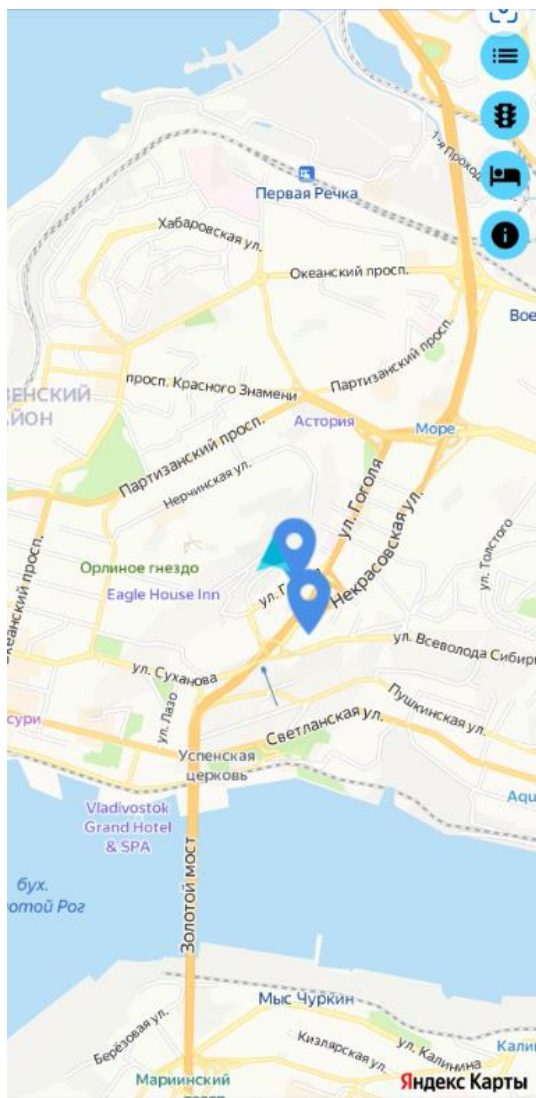


Рисунок 25 - Объекты города на карте

При нажатии на метку открывается окно с информацией о метке и прокладке маршрута, рисунок 26.

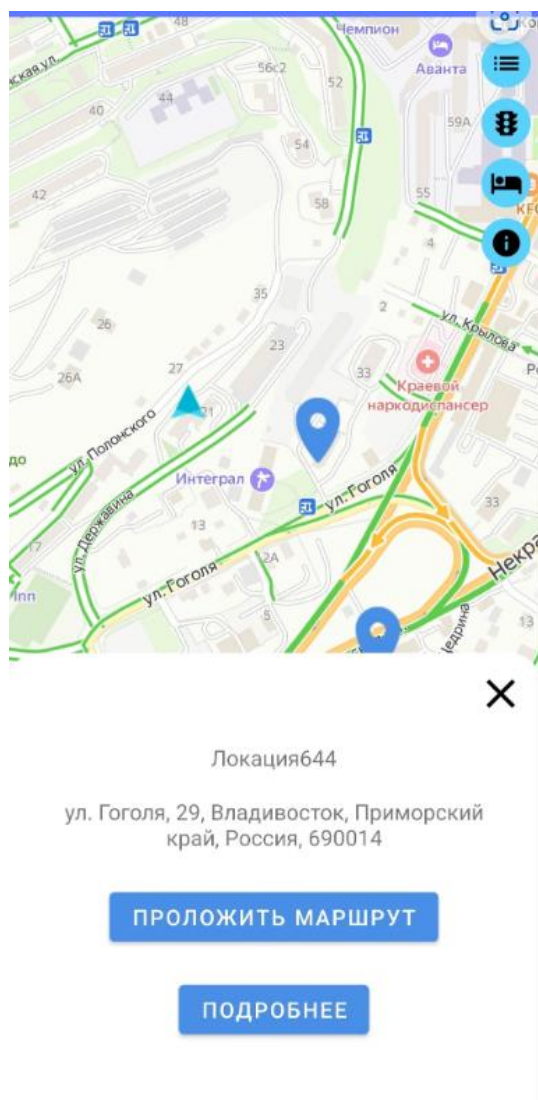


Рисунок 26 – Объекты города

При нажатии кнопки «Проложить маршрут», система пытается проложить маршрут относительно объектов, добавленных в список. В случае успеха маршрут отображается в главном меню, иначе сообщение об ошибке

«Невозможно построить маршрут».

При нажатии кнопки «Назад» открывается главное меню.

При нажатии кнопки «Отель», отображаются метки с близко находящимися отелями.

Кнопка «Избранное» - в данной версии приложения функционал не доступен.

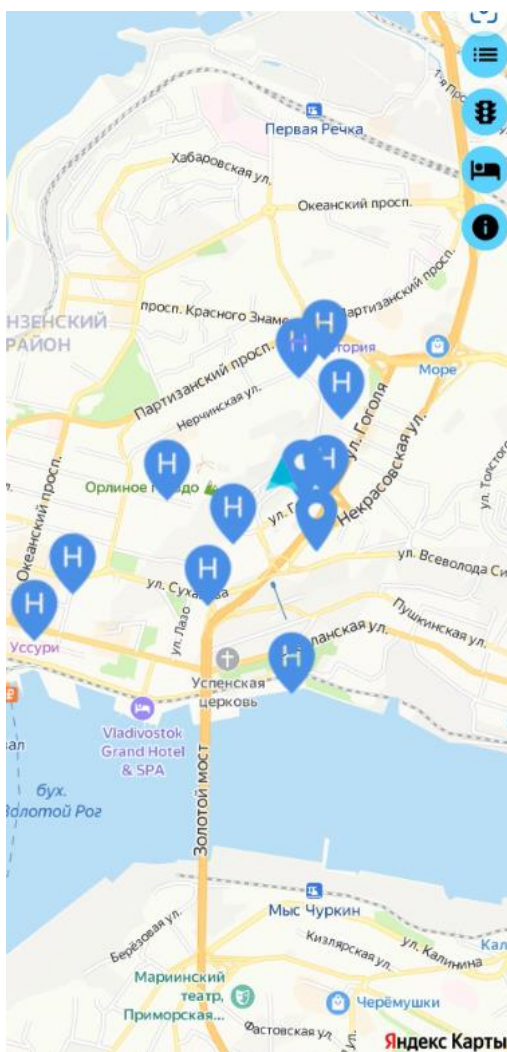


Рисунок 27 – Отображение ближайших отелей

При нажатии на объекты отображается информация о отеле с номером телефона, рисунок 28.

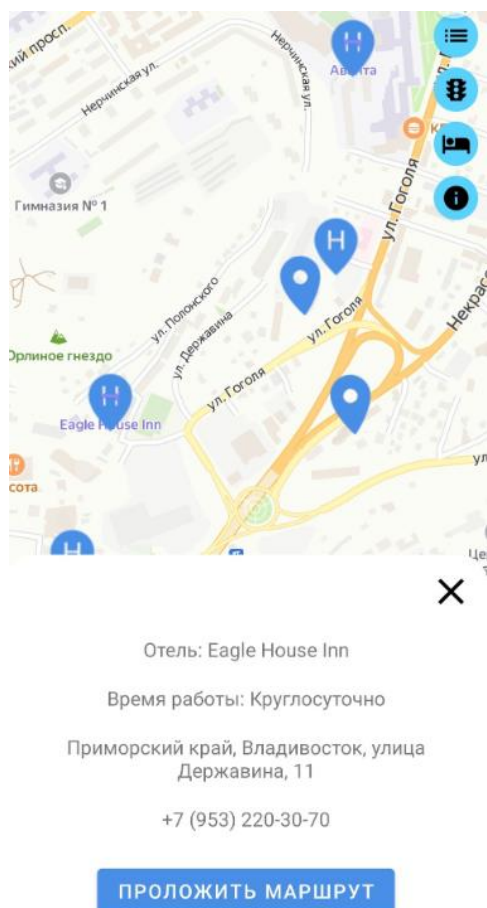


Рисунок 28 – Информация об отеле

При нажатии на кнопку «Назад» открывается окно выбора типа объекта.

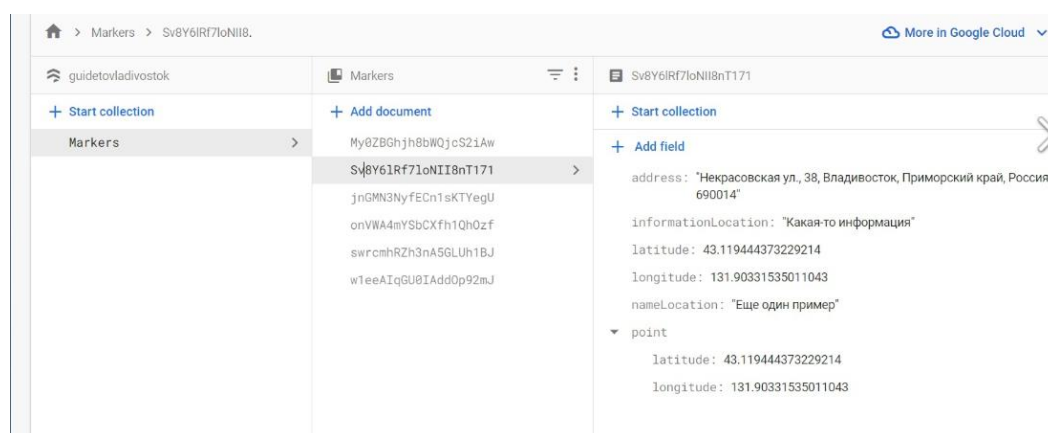
Глава 4 Реализация и тестирования

4.1 Реализация серверной части

Для реализации серверной части используется Firebase и Firestore. Firebase поддерживает данные JSON, и все подключенные к ней пользователи получают оперативные обновления после каждого изменения. Firestore — это размещенная в облаке база данных NoSQL, к которой ваши Apple, Android и веб-приложения могут обращаться напрямую через

собственные SDK. Cloud Firestore также доступен в собственных Node.js, Java, Python, Unity, C++ и Go SDK, в дополнение к REST и RPC API. Следуя модели данных NoSQL Cloud Firestore, вы храните данные в документах, содержащих поля, сопоставленные со значениями. Эти документы хранятся в коллекциях, которые являются контейнерами для ваших документов, которые вы можете использовать для организации своих данных и создания запросов.[10]

Теперь чтобы получить данные с нашего сервера в виде JSON объекта заходим в папку Markers и видим следующие данные:



4.2 Реализация клиентской части

Для реализации клиентской части был выбран язык kotlin [11] и облачного сервера Firebase[12].

Kotlin представляет современный, статически типизированный и один из самых быстроразвивающихся языков программирования, созданный и развиваемый компанией JetBrains. Kotlin можно использовать для создания самых разных приложений. Это и приложения для мобильных устройств - Android, iOS. Причем Kotlin позволяет писать кроссплатформенный код, который будет применяться на всех платформах. Это и веб-приложения, причем как серверные приложения, которые отработывают на стороне на стороне сервера - бекэнда, так и браузерные клиентские приложения - фронтенд. Kotlin также можно применять для создания десктопных приложений, для Data Science и так далее.

Таким образом, круг платформ, для которых можно создавать приложения на Kotlin, чрезвычайно широк - Windows, Linux, Mac OS, iOS, Android. Для реализации в приложении карт, были выбраны карты от Яндекс – Яндекс.МарKit [13].

При разработке клиента был использован паттерн MVP[14], рисунок 16. MVP, или Model-View-Presenter, это один из нескольких паттернов, который способствует разделению ответственности при реализации пользовательского интерфейса

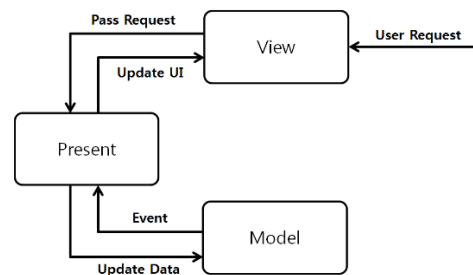


Рисунок 16 - Модель MVP

Сопоставим каждому классу его тип в паттерне MVP:View:

1. AboutProgramActivity.kt
2. AddPointFragment.kt
3. CreateOrChangeLocationFragment.kt
4. DeleteDialogFragment.kt
5. InfoHotelFragment.kt
6. InformationLocationFragment.kt
7. KeyValue.kt
8. ListLocationActivity.kt
9. MapActivity.kt

10. MapKitInitializer.kt

11. RouteFragment.kt

Presenter:

1. ChangeLocationPresenter.kt

2. CreateLocationPresenter.kt

3. DeleteLocationPresenter.kt

4. ListLocationPresenter.kt

5. MapPresenter.kt

Model:

1. CallBackHandler.kt

2. CallBackOnClose.kt

3. ChangeLocationModel.kt

4. CreateLocationModel.kt

5. DeleteLocationModel.kt

6. ListLocationModel.kt

7. HotelEntity.kt

8. MarkerEntity.kt

9. ChangeLocationRepository.kt

10. CreateLocationRepository.kt

11. Database.kt

12. DeleteLocationRepository.kt

13. GetListLocationRepository.kt

При создании приложения использовалась android sdk. Android SDK— универсальное средство разработки мобильных приложений для операционной системы Android. Отличительной чертой от обычных редакторов для написания кодов является наличие широких функциональных возможностей, позволяющих запускать тестирование и отладку исходных кодов, оценивать работу приложения в режиме совместимости с различными версиями ОС Android и наблюдать результат в реальном времени [16].

4.3 Характеристика прототипа

Прототип программного средства «Гид по достопримечательностям Владивостока» состоит из прототипов двух подсистем: клиент и сервер.

Прототип сервера представляет программу, написанную с использованием программной платформы Firebase . Программа запускается через облачный сервер.

Объем дискового пространства, занимаемого сервером – 42 Мб.

Поддерживаемые операционные системы – системы семейства Windows, Ubuntu, Mac OS.

Для реализации интерфейса использовался язык xml

Прототип клиента представляет мобильное приложение, написанное на языке kotlin

Объем дискового пространства, занимаемого приложением на устройстве – 110 Мб.

Поддерживаемые системы - совместимые устройства с Android 8 и выше

4.4 Тестирование и эксперименты

4.4.1 Цель тестирования

Целью проведения тестирования является проверка соответствия программной системы с разработанными функциональными требованиями, а также выявления ошибок, допущенных при разработке.

4.4.2 Цель экспериментальных исследований

Целью проведения экспериментальных исследований является проверка правильности решения программой прикладных задач предметной области и оценка пользователем удобства работы с программой.

Эксперименты проводились на основе ситуаций предметной области (см. п. 2.4 **Ошибка! Источник ссылки не найден.**).

4.4.3 Описание среды выполнения тестовых сценариев и проведение экспериментов

Экспериментальные проверки для сервера проводились на компьютере со следующими параметрами:

1. ОС: Windows 10
2. Процессор: AMD Ryzen 5 2,1 ГГц
3. ОЗУ: 8ГБ 2400 МГц DDR4

Экспериментальные проверки для клиента проводились на мобильном устройстве со следующими параметрами:

1. Модель устройства: vivo

y31

2. ОС: Android 13

4.4.4 Описание тестовых ситуаций и результаты тестирования

Добавление объекта города

1. Добавление объекта города со всеми правильными входными данными
2. Добавление объекта города с неправильными входными данными
 - 2.1. Не указан тип объекта
 - 2.2. Не введено имя объекта
 - 2.3. Не введено описание объекта
 - 2.4. Не введена долгота
 - 2.5. Не введена широта
3. Добавление пустого объекта города

Тестовые ситуации для клиента:

Связь с сервером для обновления данных

1. Связь с сервером при отсутствии сети интернет
2. Связь с сервером при наличии сети интернет

Обновление данных об объектах города на устройстве

1. С сервера пришла новая информация об объектах города

Просмотр данных об объектах города на устройстве

1. База данных об объектах города пуста
2. База данных об объектах города имеет данные

Построение маршрута на карте

1. Турист строит маршрут из пустого списка объектов города
2. Турист строит маршрут из двух объектов
3. Турист строит маршрут более чем из двух объектов
4. Турист добавляет в маршрут новый объект города
5. Турист очищает список объектов в маршруте

Заключение

В ходе дипломной работы был проведен анализ предметной области «Гид по достопримечательностям Владивостока». Из проведенного анализа были выявлены основные задачи для данной предметной области: добавление данных об объектах города, хранение данных об объектах города, отображение данных о городах, построение маршрута. Так же в ходе анализа были выделены методики решения данных задач. После проведения анализа была построена математическая модель предметной области и методов решения задач.

Помимо анализа и математической модели, были сформулированы требования к программным компонентам системы: клиент и сервер. Построена архитектура для каждой из подсистем, выполнена программная реализация данных систем.

Цель дипломной работы заключалась в разработке прототипа гида по достопримечательностям Владивостока, позволяющего туристам, используя мобильное приложение, просматривать информацию о городе и на ее основе строить маршрут. Опираясь на полученные в ходе выполнения дипломной работы результаты, можно сделать вывод о том, что цель дипломной работы была выполнена.

Список литературы

1. Приморье. 60 лет. 1938 – 1998 / Администрация Приморского края. – Владивосток: Изд-во "Уссури", 1998. – 100 с.
2. Дальний Восток России: исторический опыт и пути развития региона (Первые Крушановские чтения, 1998 г.). – Владивосток: Изд-во Дальнаука, 2001. – 352 с.
3. Мировой рынок смартфонов на 2023.
<https://nsysgroup.com/ru/blog/global-mobile-market-today/?ysclid=lpv0pxtbv293116305>
4. Государственная программа Приморского края «Развитие туризма в Приморском крае» на 2013-2021 годы.
<http://www.primorsky.ru/authorities/executive-agencies/departments/tourism-pk/gosudarstvennaya-programma-primorskogo-kraya-razvitie-turizma-v-primorskom-krae-na-2013-2017-gody-.php>
5. Артемьева И.Л. Сложно-структурированные предметные области. Построение многоуровневых онтологий. // Информационные технологии, 2009, № 1, с. 16-21
6. Карты Google
<https://itunes.apple.com/ru/app/google-karty-golosovaa-gps/id585027354?mt=8>
7. ForeverMap 2
<https://itunes.apple.com/us/app/forevermap-2-worldwide-offline/id424183595?mt=8>
8. AroundMe
<https://itunes.apple.com/ru/app/aroundme/id290051590?mt=8>
9. TripAdvisor
<https://itunes.apple.com/us/app/tripadvisor-hotels-flights/id284876795?mt=8>

10. Firestore

<https://firebase.google.com/docs/firestore?hl=ru>

11. Kotlin

<https://kotlinlang.org/docs/home.html>

12. Firebase

<https://firebase.google.com/products-build>

13. Яндекс. MapKit

<https://yandex.ru/dev/mapkit/doc/en/?ysclid=ljpvvf9ino693075248>

14. MVP

<https://lpgenerator.ru/blog/2015/03/02/rukovodstvo-po-sozdaniyu-mvp-ili-minimalno-zhiznesposobnogo-produkta/?ysclid=ljpvypfwfc778638979>

15. Android SDK

<https://documentation.idmission.com/identity/Android-SDK-1/>