

Министерство образования и науки Российской Федерации
ФГБОУ ВО РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
(РГГМУ)

Институт Информационных систем и геотехнологий
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

БАКАЛАВРСКАЯ РАБОТА

На тему _____ Разработка ГИС по утилизации мусора _____

Исполнитель _____ Цветков Алексей Игоревич _____
(фамилия, имя, отчество)

Руководитель _____ доктор географических наук, профессор _____
(ученая степень, ученое звание)

_____ Шилин Михаил Борисович _____
(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой _____
(подпись)

_____ доктор технических наук, профессор _____
(ученая степень, ученое звание)

_____ Истомин Евгений Петрович _____
(фамилия, имя, отчество)

«__» _____ 20__ г

Санкт–Петербург
2022

СОДЕРЖАНИЕ

Список сокращений и специальных терминов	4
ВВЕДЕНИЕ	5
Глава 1 Аналитическая часть	8
1.1 Анализ предметной области	8
1.2 Анализ статистики ключевых запросов	10
1.3 Анализ аналогов	13
1.4 Анализ функций приложения	17
1.5 Анализ объекта исследований	18
Глава 2 Проектирование ИС	22
2.1 Концептуальное проектирование	22
2.2 Функциональное проектирование	23
2.3 Объектно-ориентированное проектирование	24
Глава 3 Реализация	31
3.1 Обоснование выбора инструментальных и программных средств	31
3.1.1 Среда разработки	31
3.1.2 Язык программирования	33
3.1.3 База данных	35
3.2 Интерфейс мобильного приложения «PollutionHelper»	37
3.3 Система хранения данных	54
3.3.1 Authentication	54
3.3.2 RealtimeDatabase	56
3.3.3 Storage	59
3.3.4 Схема восстановления и сохранения информации	60
3.4 Надёжность системы	62
3.4.1 Расчёт структурной схемы надёжности	63
3.4.2 Расчёт надёжности программного обеспечения	67
ЗАКЛЮЧЕНИЕ	70

СПИСОК ЛИТЕРАТУРЫ	73
ПРИЛОЖЕНИЕ А	76
ПРИЛОЖЕНИЕ Б	79
ПРИЛОЖЕНИЕ В	84

Список сокращений и специальных терминов

BPMN	– Business Process Model and Notation, нотация и модель бизнес-процессов;
IDE	– интегрированная среда разработки;
IOS	– iPhoneOS, операционная система Apple;
SWOT	– strengths (сильные стороны), weaknesses (слабые места), opportunities (возможности), threats (угрозы);
UML	– унифицированный язык программирования;
БД	– база данных;
ВКР	– выпускная квалификационная работа;
ГИС	– геоинформационная система;
Девайс	– устройство;
ИС	– информационная система;
ИТ	– информационная технология;
ООП	– объектно-ориентированного программирования;
ОС	– операционная система;
Рефакторинг	– переработка;
ТБО	– твёрдые бытовые отходы;
Тutorial	– обучающее пособие;
Фреймворк	– шаблон.

ВВЕДЕНИЕ

Современная экологическая обстановка характеризуется следующими проблемами [21]:

- парниковый эффект;
- загрязнение воды, почвы, воздуха;
- истощение озонового слоя Земли;
- природные катаклизмы;
- гибель флоры и фауны.

Все обозначенные выше проблемы прямо или косвенно связаны с производимыми человечеством отходами жизнедеятельности.

На сегодняшний день развитые и развивающиеся страны используют три способа утилизации отходов [22].

- Захоронение отходов – изоляция угрожающих здоровью человека промышленных остатков производства на специально отведенных для этого площадях. Представителями данного способа выступают Россия и Китай.
- Извлечение энергии – процесс выработки электрической и тепловой энергии в результате мусоросжигания. Данный способ наиболее распространён в Северной Америке.
- Сортировка – это процесс разделения отходов на различные элементы. Подход на основе сортировки используется в Японии и Северной Европе.

Страны Европы давно практикуют отдельный сбор мусора. Как результат – отличная экологическая обстановка, дешёвое вторсырьё, сохранение природных богатств.

В России, в настоящее время, управление отходами начинает активно развиваться. Как в любой развивающейся отрасли на этапе становления

существует большое количество проблем и сложностей – люди не знают, и часто даже не задумываются над тем, как правильно разделять мусор, где находятся пункты по сбору мусора и для чего требуется отдельный сбор. Между тем, правильная утилизация и сортировка отходов способна решить множество экологических и финансовых задач[23]:

- снижение потребления природных ресурсов из-за использования вторсырья;
- сокращение объема отходов, складываемых на полигонах ТБО и несанкционированных свалках;
- улучшение экологической обстановки;
- сокращение расходов на повторную переработку.

Одним из вариантов решения проблемы сортировки мусора в России является разработка систем/ приложений, позволяющих получить информацию о вариантах отдельного сбора мусора и местах, где этот сбор производится.

Объектом исследований настоящей ВКР выступает экологическая отрасль, связанная с отдельным сбором мусора.

Предметом исследования является проектировка и разработка мобильного приложения с элементами геоинформационной системы для поиска мест отдельного сбора мусора.

Цель ВКР – решение задачи управления мусорными потоками при отдельном сборе, путём разработки мобильного геоинформационного приложения.

Для достижения цели необходимо выполнить следующие задачи.

1. Провести анализ предметной области;
2. Рассмотреть аналоги информационных систем;
3. Определить основные функции проектируемой информационной системы;
4. Спроектировать информационную систему «PollutionHelper» с учётом проведённого анализа;

- 5.Реализовать информационную систему «PollutionHelper»;
- 6.Рассчитать надёжности системы;
- 7.Сделать выводы относительно проделанной работы и рассмотреть возможности улучшения и развития системы.

Методы, стандарты и технологии, примененные в ВКР:

- сравнительный анализ типовых проектных решений;
- планирования разработки систем SWOT, VCM и ISA;
- методология моделирования бизнес-процессов BPMN;
- методы графического описания UML для объектного моделирования в области разработки программного обеспечения.

В ходе выполнения практической части ВКР, будут использованы следующие вспомогательные инструменты:

- пакет программ Microsoft Office;
- облачная база данных FirebaseDatabase;
- интегрированная среда разработки Android Studio;
- язык программирования Java.

В результате выполнения ВКР будет разработано мобильное приложение на операционной системе Android, отображающее места раздельного сбора мусора на карте.

Глава 1 Аналитическая часть

1.1 Анализ предметной области

На сегодняшний день существуют и используются несколько технологий переработки мусора: механическая, термическая, а также методы химического воздействия. Эти технологии позволяют повторно использовать многие виды мусора после обработки [24]:

- пластик – он несёт особую угрозу окружающей среде, ведь только 2,5% пластиковых отходов перерабатываются, а оставшаяся часть собирается на специально отведенных полигонах или разлагается в естественных условиях в течение от 100 до 400 лет;
- макулатура – она составляет более трети состава твёрдых бытовых отходов, так что использование бумаги и картона в качестве вторсырья позволит сохранять большие лесные массивы;
- металл – это вид материала, который уже сейчас активно подвергается переработке, а полученное сырьё помогает рационально использовать не возобновляемые ресурсы планеты;
- стекло – это материал, отличающийся низкой себестоимостью и длительным периодом разложения (до 1 миллиона лет), который можно перерабатывать неограниченное количество раз;
- текстиль – его доля на свалках составляет около 65%, а переработка позволяет получить различные волокна, вату, паклю, утепляющие и изоляционные материалы, и, кроме того, сокращает количество выбрасываемого мусора;
- вышедшая из строя электроника – при утилизации таких товаров используется метод сжигания (что негативно влияет на окружающую среду из-за вредоносных выделений, содержащих в себе пары ядов),

хотя такой вид мусора должен быть правильно утилизирован с соблюдением всех требований технологического процесса;

- батарейки – они содержат в себе такие химические элементы, как свинец, кадмий, ртуть, никель, цинк и щёлочь, попадающие в процессе горения в атмосферу, отравляя в дальнейшем почву и воду и требующие сдачи в соответствующие пункты приёма, которые переправляют их на заводы, для изготовления вторсырья.

Основной проблемой, препятствующей активному участию людей в решении задачи управления мусорными потоками, является неосведомленность, которая заключается в двух аспектах:

- люди не понимают важность сортировки отходов, цель которой заключается в предотвращении смешивания разных видов отходов, их повторной переработке и, как следствие, в предотвращении загрязнения окружающей среды;
- даже если человек понимает важность отдельного сбора мусора, он подчас не знает, куда может сдать отсортированные отходы.

На решение именно этой проблемы и направлено разрабатываемое мобильное приложение. «PollutionHelper» будет отображать информацию о ближайших местах отдельного сбора мусора относительно местоположения пользователя с фильтром по категориям.

Для удобства поиска информация будет отображаться на карте с представленными фотографиями мест отдельного сбора мусора и списком категорий принимаемых в этих местах отходов. В приложении также будет реализован чат, который позволит делиться своими замечаниями и мнением по поводу пункта сбора с другими пользователями.

Организационная структура приложения «PollutionHelper» представлена на рисунке 1.1.

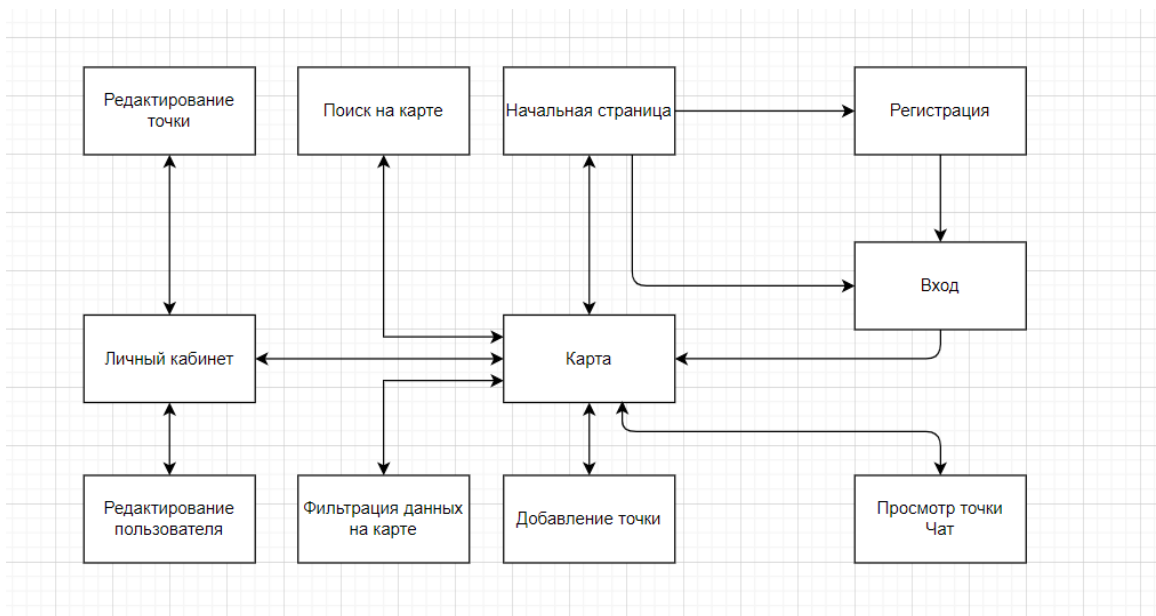


Рисунок 1.1 -Организационная структура приложения

1.2 Анализ статистики ключевых запросов

Для подтверждения обозначенной выше проблемы неосведомленности, был проведен дополнительный анализ запросов ключевых слов по обозначенной тематике в поисковых системах Яндекс и Google.

Со стороны Яндекс использован сервис «Яндекс.Wordstate». Данный продукт собирает статистику в поисковой системе «Яндекс» по ключевым словам в течение месяца. Полезной особенностью является сбор статистики не только по ключевому слову, но и по похожим запросам, это позволило увеличить точность анализа, так как запросы могут отличаться по содержанию, имея одну смысловую нагрузку. Статистические данные по запросам за месяц представлены на рисунках 1.2.

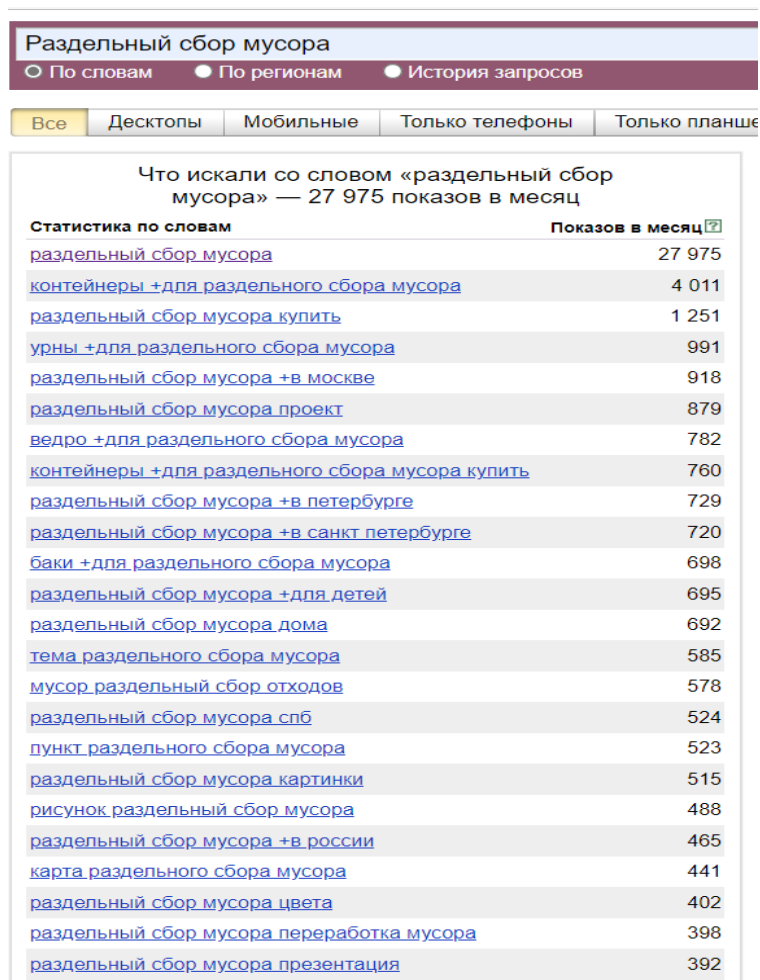


Рисунок 1.2 -Статистика запросов в Яндекс

Кроме того, в «Яндекс.Wordstate» есть возможность просмотра статистики за год в виде графика (рисунок 1.3). Такая функция полезна для просмотра картины в целом.

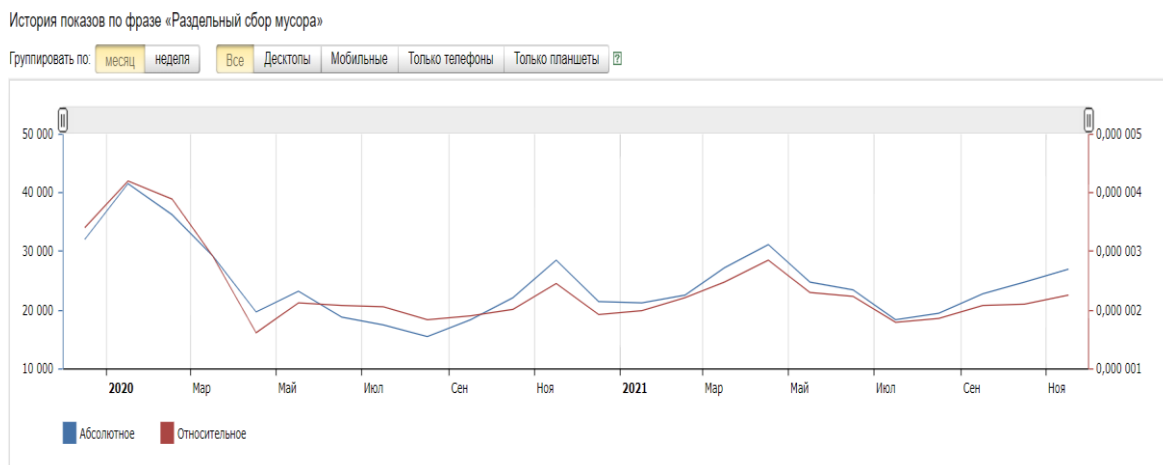


Рисунок 1.3 - Статистика запросов за год

Для анализа запросов в Google использован сервис «GoogleTrends» (рисунок 1.4). Это решение позволяет просматривать статистику по ключевому слову в поисковой системе «Google» за год. В отличие от аналога со стороны Яндекс, рассматриваемый сервис имеет существенный минус, так как не ведёт статистику по похожим запросам и, следовательно, не даёт увидеть чёткую статистическую картину по запросу.

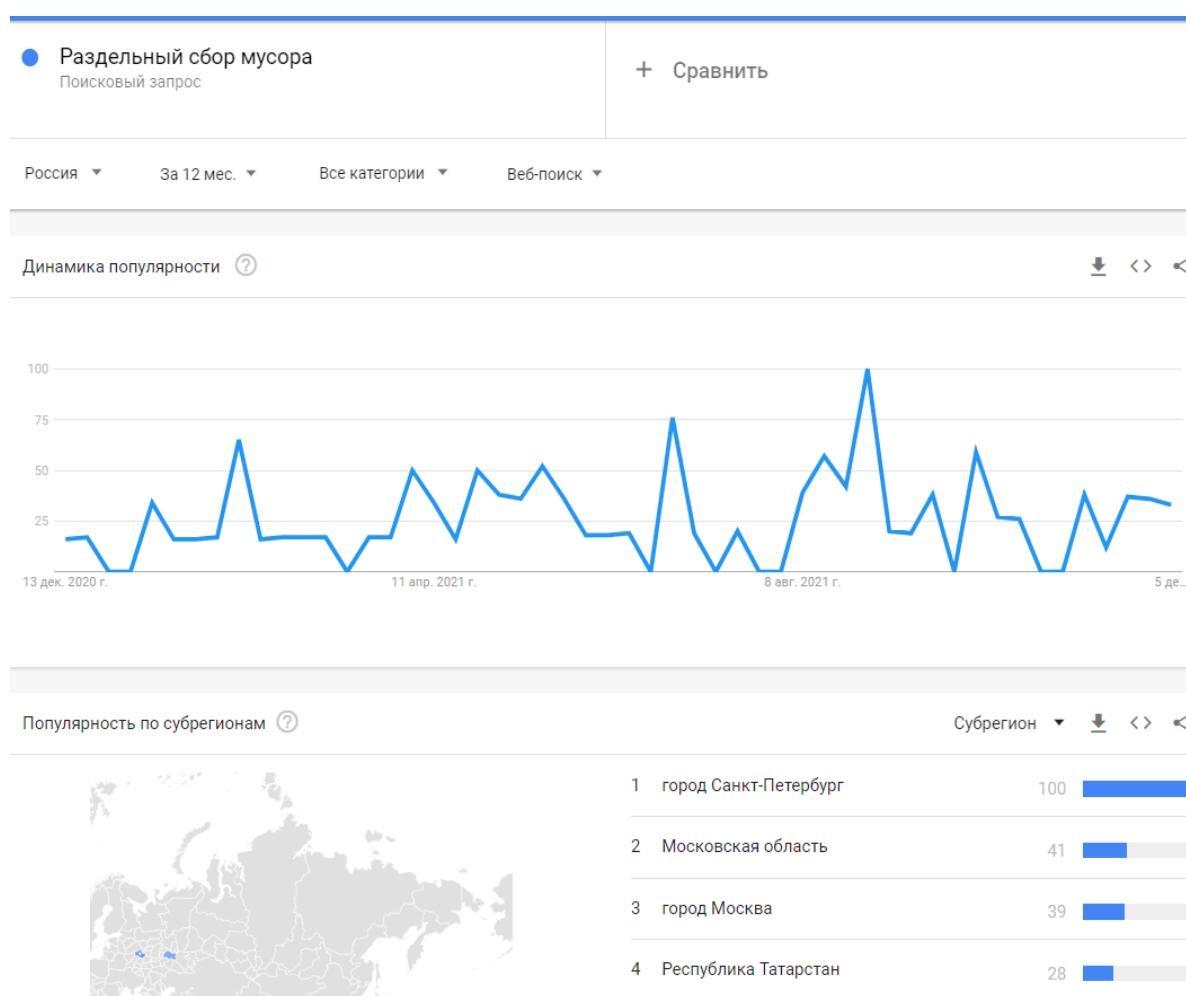


Рисунок 1.4 -Статистика по запросу в «GoogleTrends»

Как видно из проделанного анализа, существует большое количество запросов ежемесячно, около 28 тысяч. Основываясь на запросах в Яндекс, по обозначенной тематике, можно заключить, что люди заинтересованы данной темой. Спрос присущ в основном жителями больших городов, таких как

Москва и Санкт-Петербург. Это обусловлено «плотностью проблем» - большой численностью населения на единице площади.

Как итог, проведение данного анализа подтвердило существование проблемы сортировки мусора в России и необходимость её решать.

Как было отмечено выше, вариантом решения проблемы является разработка приложений, позволяющих получить онлайн информацию о местах раздельного сбора мусора.

Решением этой задачи и занимается разрабатываемое приложение «PollutionHelper».

1.3 Анализ аналогов

В интернете доступны к использованию несколько приложений по схожей тематике. Для разработки проекта был проведен анализ аналогичных продуктов, позволивший определить сильные и слабые стороны конкурентов, что, в свою очередь, позволило сформировать общие функции необходимые приложению и функции, которые будут выделять приложение среди других.

1. «Зелёная карта» (Рисунок 1.5) – Приложение для тех, кто хочет жить «экологично». Внутри: места сбора отходов и ненужных вещей на территории Беларуси, карта «экомест» - мест с хорошей экологической обстановкой, интерактивный помощник по сортировке отходов.

Плюсы данного приложения сводятся к удобности и понятности интерфейса для стабильной работы и большого спектра отображаемых услуг, которые в свою очередь, достаточно удобно фильтруются.

Основной минус состоит в том, что приложение поддерживается только на территории Беларуси. Еще одним минусом является, отсутствует возможность пользователю самому ставить точки на карте: этим занимается администратор непосредственно через базу данных.

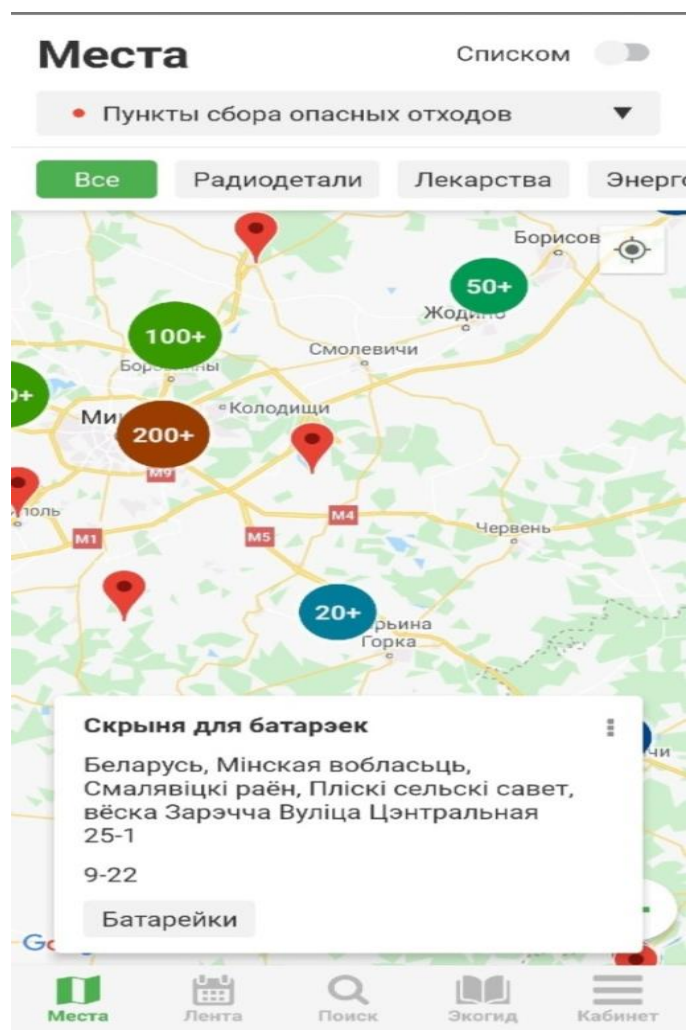


Рисунок 1.5-Приложение «Зелёная карат»

2. «+1Город» (Рисунок 1.6) – первый веб-навигатор и мобильное приложение с интегрированной программой лояльности, включающие в себя социальные и экологические сервисы и предложения для сознательных горожан.

Главным плюсом этого приложения является система поощрения пользователей, достаточно широкий объём предложений от спонсоров, таких как купоны в местах общепита и скидки в определённых магазинах.

Основной минус заключается в том, что приложение сделано на основе сайта, и из-за этого работоспособность продукта крайне низкая.

Существенным минусом является невозможность взаимодействия с приложением без регистрации на начальном экране (регистрация обязательна).

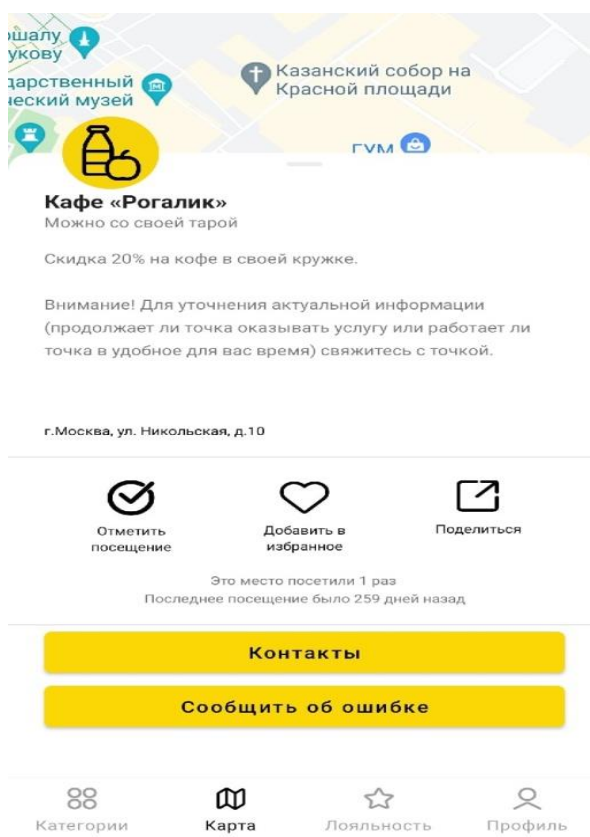


Рисунок 1.6-Приложение +1Город

3. «ЕcoHub» (Рисунок 1.7) – Приложение, показывающее места сбора мусора.

Плюсы продукта: простотой и понятный пользователю интерфейс, нет лишних кнопок, рекламных сообщений.

Минусы вытекают из его плюсов: в результате простоты проекта его функционал максимально урезан. Данные на карте не информативные, отсутствует какая-либо система поощрений.

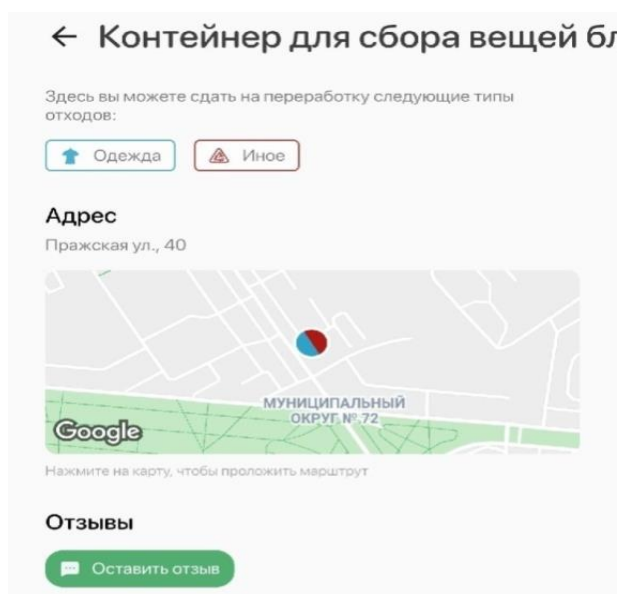


Рисунок 1.7 -Приложение EсоHub

Для наглядного представления результатов проведённого анализа аналогов данные сгруппированы в таблицу:

Таблица 1 – сравнение аналогов мобильных приложений

	Зелёная карта	+1Город	EсоHub	Разрабатываемое приложение «PollutionHelper»
Регистрация	Не обязательная	Принудительная	Не обязательная	Не обязательная
Карта с геолокацией	+	+	+	+
Работоспособность	Стабильная	Не стабильная	Стабильная	Стабильная
Построение маршрута	-	-	+-	+
Фильтр	+	+	+	+
Личный кабинет	+ -	+	-+	+
Добавление точки	-	+	+	+
Комментарии для других пользователей	-	-	-	+

Продолжение таблицы 1 – сравнение аналогов мобильных приложений

	Зелёная карта	+1Город	ЕcoHub	Разрабатываемое приложение «PollutionHelper»
Система поощрения пользователей	-	+	-	+
Информативность маркера на карте	Частичная	Частичная	Никакой	Полная
Страна	Беларусь	Россия	Россия	Россия с возможностью внедрения по всему миру

Плюсами (+) и минусами (-) обозначены достоинства и недостатки информационных систем.

В результате проведённого анализа были выявлены основные недостатки конкурентов, на которые стоит обратить внимание при проектировании проекта, что позволит приложению быть более востребованным в своём сегменте.

1.4 Анализ функций приложения

В результате проведённых выше анализов чётко вырисовывается набор основного и дополнительного функционала, который должно включать в себя мобильное приложение «PollutionHelper», а именно:

- регистрация пользователя в системе с помощью почты/ подтверждение принадлежности почты пользователю;
- авторизация пользователя;
- возможность входа в приложение без обязательной авторизации с ограниченным функционалом;
- просмотр карты;

- определение местоположения;
- поиск на карте;
- просмотр личного кабинета;
- изменение данных о пользователе;
- добавление точек на карту;
- редактирование данных о точках на карте;
- просмотр данных о точках на карте;
- пользовательский чат.

Кроме того, важными задачами являются:

- обеспечить для пользователя интуитивно понятный интерфейс, который не будет заставлять долго думать над возможностью его использования;
- организовать канал связи между пользователем и администрацией приложения для решения проблем, связанных с работой приложения.

1.5 Анализ объекта исследований

SWOT анализ

SWOT анализ [3] – это метод стратегического планирования, заключающийся в выявление внешних и внутренних плюсов и минусов организации. Этот подход позволяет идентифицировать, классифицировать, ранжировать и выбирать проекты по разработке ИС таким образом, чтобы они были увязаны с сильными и слабыми сторонами организации, а также возможностями и угрозами.

SWOT анализ позволил определить сильные и слабые стороны разрабатываемой системы «PollutionHelper», а так же дальнейшие пути модификации продукта. Результаты SWOT анализа представлены в таблице 2.

Таблица 2- SWOT анализ

	Достоинства	Недостатки
Внутренние	<ol style="list-style-type: none"> 1. Возможность анонимного использования 2. Возможность видеть мнение других пользователей 3. Информативная карта 	<ol style="list-style-type: none"> 1. Отсутствие спонсоров 2. Слабое стимулирование пользователей 3. Невозможность работы без интернета
Внешние	<ol style="list-style-type: none"> 1. Актуальность в современном мире 2. Отечественная разработка 3. Возможность выхода на международный рынок 	<ol style="list-style-type: none"> 1. Конкуренты 2. Слабо сформированный имидж организации 3. Отсутствие сознательности у людей

ISA анализ

ISA анализ [4] - этот подход предлагает нейтральную архитектурно-концептуальную схему для проектных решений по созданию ИС, которая может подходить для различных бизнес-стратегий.

Схема ISA представляет собой таблицу из тридцати ячеек, организованных в виде пяти строк и шести столбцов. Строки представляют различные точки зрения, используемые при конструировании сложных инженерных продуктов, таких как информационные системы.

Этот метод дает представление о функциональных требованиях и требованиях пользователей информационной системы «PollutionHelper». Результаты ISA анализа представлены в таблице 3.

Таблица 3 – Результаты ISA анализа

	А	В	С	Д	Е	Ф
1	Интерфейс	Поиск, добавление точек	Приложение	Пользователь, Администратор	В момент добавления точек	Продвижение идеи экологии
2	Количество пользователей	Регистрация пользователей	Приложение	Пользователь	В момент регистрации	Получение прибыли
3	Поток данных	Поиск, прокладывание маршрута, регистрация, авторизация	СУБД	Пользователь, администратор	Программирование, отладка, сопровождение	Функционирование системы
4	Документ ЖЦ	Оформление документов	Приложение	Пользователь, администратор	Документ сопровождения ЖЦ	Грамотно вывести ИС на рынок
5	Количество точек	Предоставляет точки	Приложение	Пользователь	При добавлении, редактировании точек	Оказание услуг

ВСМ анализ

Метод ценностных цепочек — VCM [5] — позволяет оценить конкурентные преимущества с помощью анализа всей цепочки видов деятельности в организации, начиная от получения сырья до конечной продукции, продаваемой и поставляемой потребителям.

Модель служит цели уяснения того, какая конфигурация цепочки ценности сулит наибольшие конкурентные преимущества. Проекты по разработке ИС могут впоследствии указать на те сегменты, операции, каналы распределения, маркетинговые подходы и т. д., которые позволяют завоевать наибольшее конкурентное преимущество.

Проведение VSM анализ позволил чётко сформулировать основную роль разрабатываемого мобильного приложения, выделить функционал, благодаря которому продукт будет конкурировать с аналогами, определить область для дальнейшего улучшения с целью получения выгоды, а именно:

- оценить информационную емкость продуктов и процессов (информационная емкость будет определяться количеством пользователей и точек на карте);

- оценить роль ИТ в отраслевой структуре (данная ИС будет играть большую роль в отрасли «экологическая безопасность», так как помогает пользователям отдельно утилизировать мусор);

- выявить и ранжировать способы, с помощью которых ИТ создает конкурентное преимущество (по сравнению с аналогами, разрабатываемая ИС будет обладать следующими преимуществами: использование без регистрации; возможность просмотра мнения других пользователей благодаря открытому чату; информативная карта; расширенный функционал);

- рассмотреть, каким образом ИТ может создать новое направление в бизнесе (так как разрабатываемая ИС будет ориентирована на места отдельного сбора мусора, должно найтись большое количество компаний, которым выгодно собирать отдельный мусор, следовательно, компаниям будет выгодно организовывать, добавлять свои точки сбора);

- разработать план, направленный на извлечение выгод от использования ИТ для владельца (интегрирование какого-либо вида направленной рекламы в приложение).

Проведенный анализ подтверждает актуальность разрабатываемой геоинформационной системы «PollutionHelper», которая позволяет решить задачу активного включения граждан в решение проблемы управления мусорными потоками, а также намечает пути ее дальнейшего развития с целью повышения эффективности и рентабельности.

Глава 2 Проектирование ИС

Этап проектирования очень важен для разработки геоинформационного продукта. На нём описывается теоретическая реализация всех заявленных функций, связь компонентов приложения между собой, определяются типы данных используемых базой. С учётом этапа проектирования строится прототип или даже рабочее приложение.

Проектирование информационных систем [6] – это упорядоченная совокупность методологий и средств создания или модернизации информационных систем. Результатом проектирования является целостная совокупность моделей, описанных в форме, пригодной для реализации системы.

Ниже рассмотрены три методологии проектирования информационных систем, на основе которых разработаны соответствующие модели.

2.1 Концептуальное проектирование

Концептуальное проектирование систем [7] — начальная стадия проектирования, на которой принимаются решения, определяющие последующий облик системы, и проводится исследование и согласование параметров созданных технических решений с возможной их организацией.

Результат этой методологии проектирования дает разрабатываемому мобильному приложению общее представление об облике будущей системы.

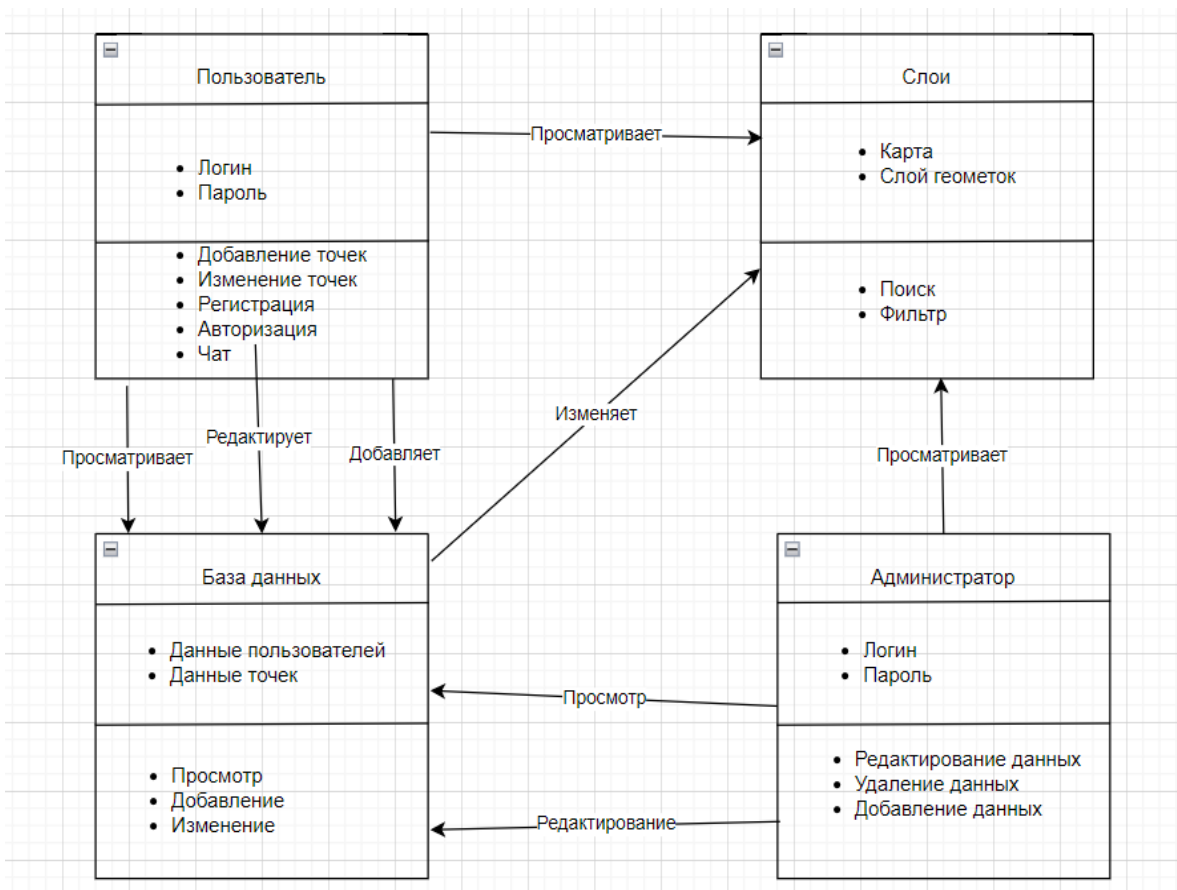


Рисунок 2.1 -Концептуальное представление системы

2.2 Функциональное проектирование

Функциональное проектирование [8] – методология, основанная на взаимодействии с функциями, то есть процессами, описывающими связь между входными и выходными параметрами.

Для построения функциональной модели системы использовалась нотация BPMN – система условных обозначений и их описания для моделирования бизнес-процессов. Часть схем разрабатываемой ИС «PollutionHelper» представлена на рисунке 2.2.1 – 2.2.2. С общим видом BPMN-схемы можно ознакомиться в приложение А.

Этот вид схем наглядно и подробно демонстрирует последовательность рабочих действий и перемещение информационных потоков, необходимых для выполнения работы приложения «PollutionHelper».

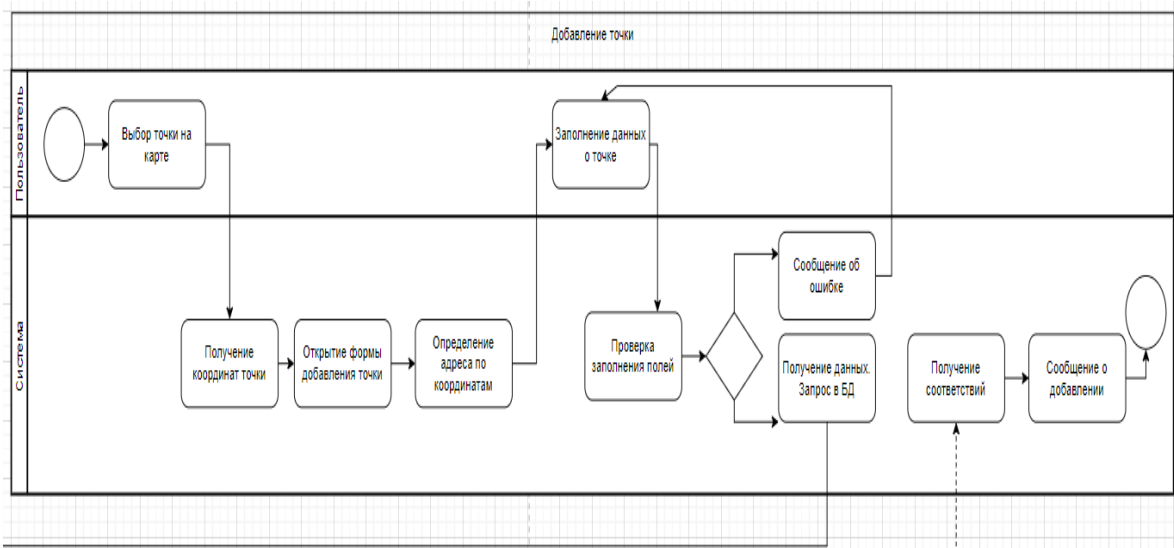


Рисунок 2.2.1 -BPMN-схема Добавление точки

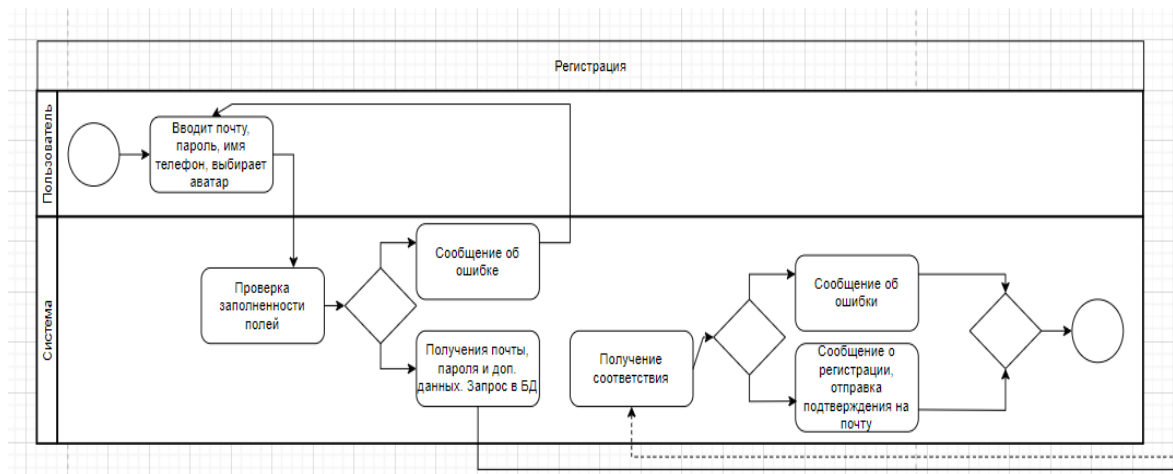


Рисунок 2.2.2 -BPMN-схема Регистрация

2.3 Объектно-ориентированное проектирование

Объектно-ориентированное проектирование [9] – методология, основанная на выделении классов объектов, установлении их характерных свойств и методов обработки. К объектно-ориентированному проектированию относят диаграммы языка моделирования UML, которые предназначены для статистического представления системы: диаграммы

вариантов использования, диаграммы классов, диаграммы последовательностей и другие.

Диаграмма вариантов использования

Диаграммы прецедентов используются для моделирования динамических аспектов ИС. Диаграммы этого типа позволяют достаточно четко описать и визуализировать поведение системы или ее части с точки зрения способа их использования. В результате, с одной стороны, пользователи системы понимают, как использовать некоторые элементы, а разработчики — как их реализовать. Диаграммы данного типа облегчают понимание системы и ее частей, представляя взгляд извне на то, как данные элементы могут быть использованы в соответствующем контексте. При этом достигается высокий уровень понимания функционирования всей системы в целом.

Конкретно для разрабатываемого приложения «PollutionHelper» данная диаграмма дает наглядное представление о пользователях системы и о функциях, которые они могут использовать.

Графическое представление Диаграммы вариантов использования (прецедентов) изображено на Рисунке 2.3.

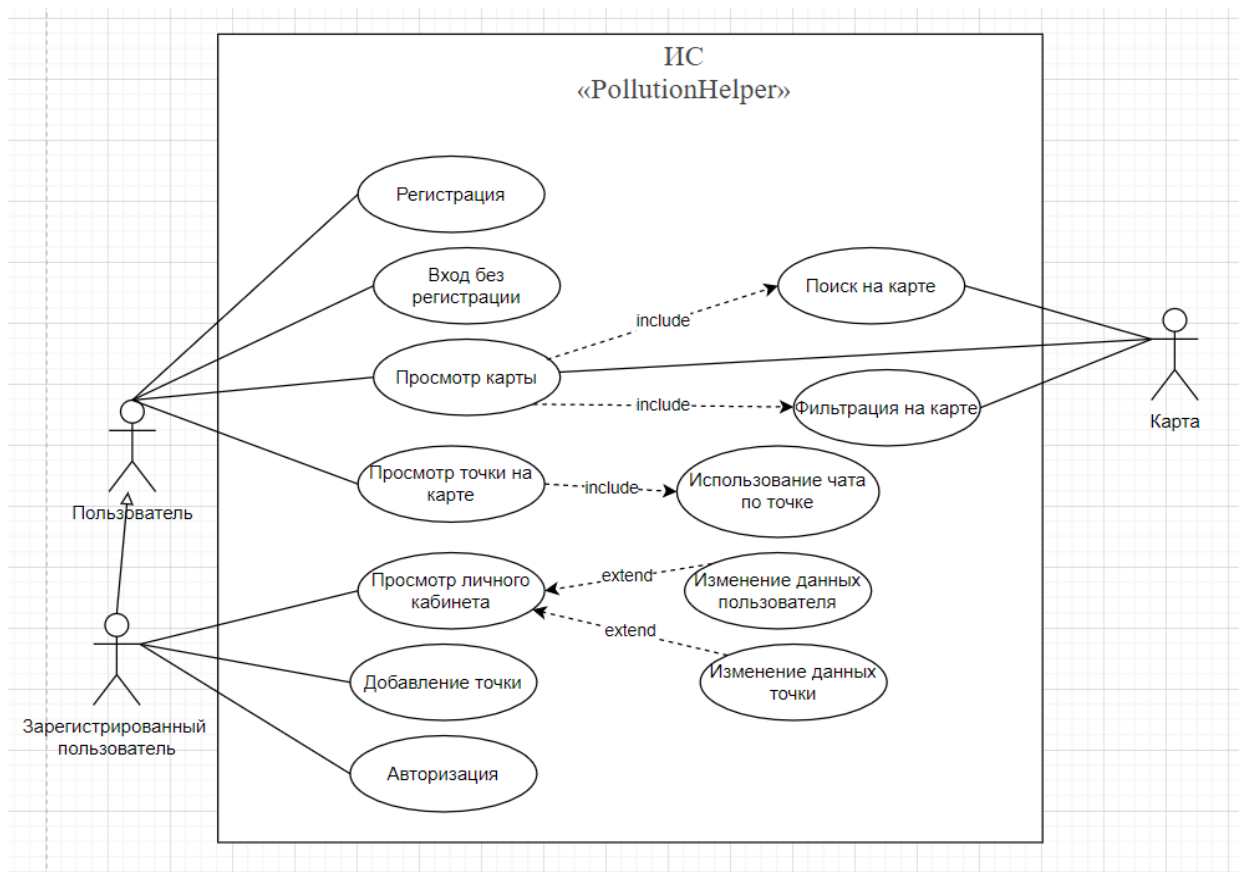


Рисунок 2.3 - Диаграмма вариантов использования

Диаграмма последовательностей

Диаграммы последовательностей моделируют взаимодействия между объектами в едином сценарии использования.

Построение диаграмм последовательностей даёт пошаговое представление о взаимодействии пользователя приложения «PollutionHelper» с системой на каждое действие, а также показывает реакцию системы на возникновение исключений.

Часть диаграмм последовательностей представлена на рисунках 2.4.1 – 2.4.3, полный перечень диаграмм находится в приложении Б.

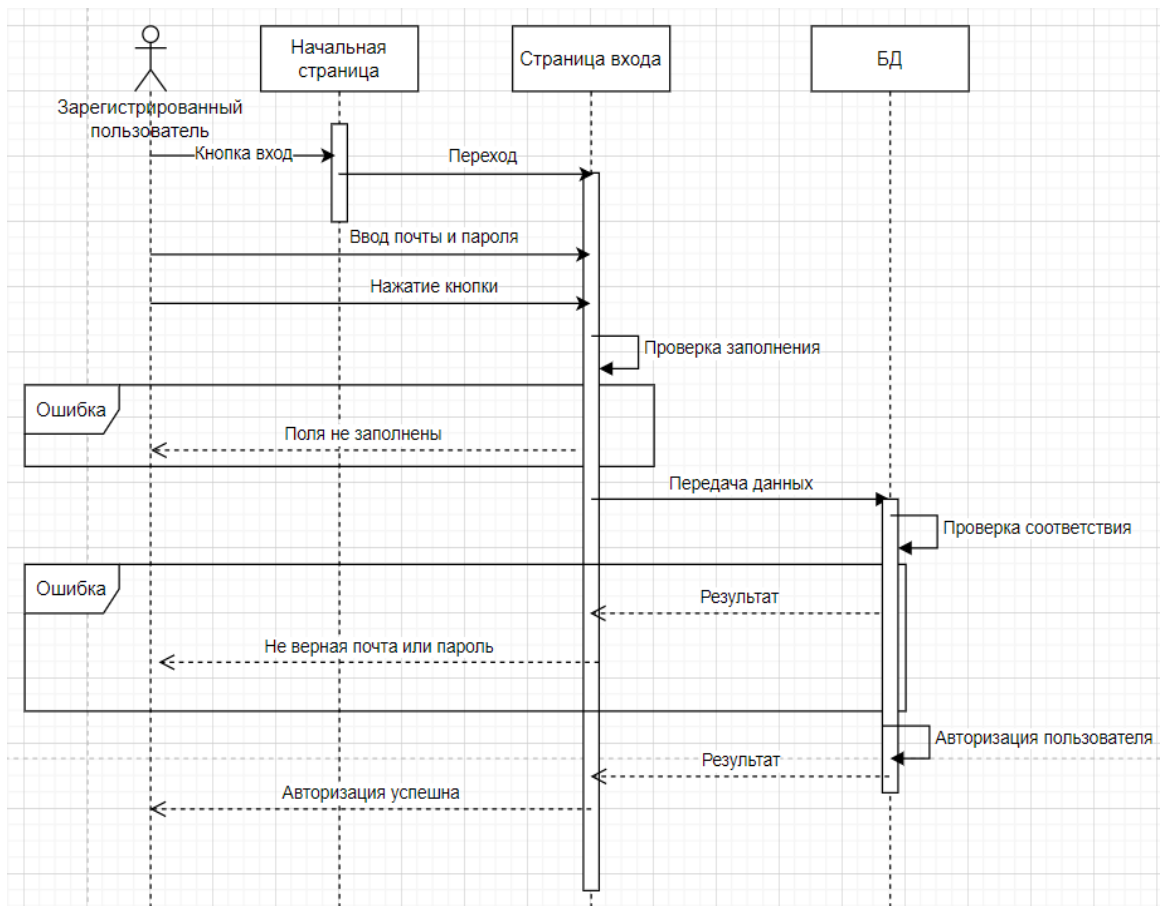


Рисунок 2.4.1 - Диаграмма последовательности, вход

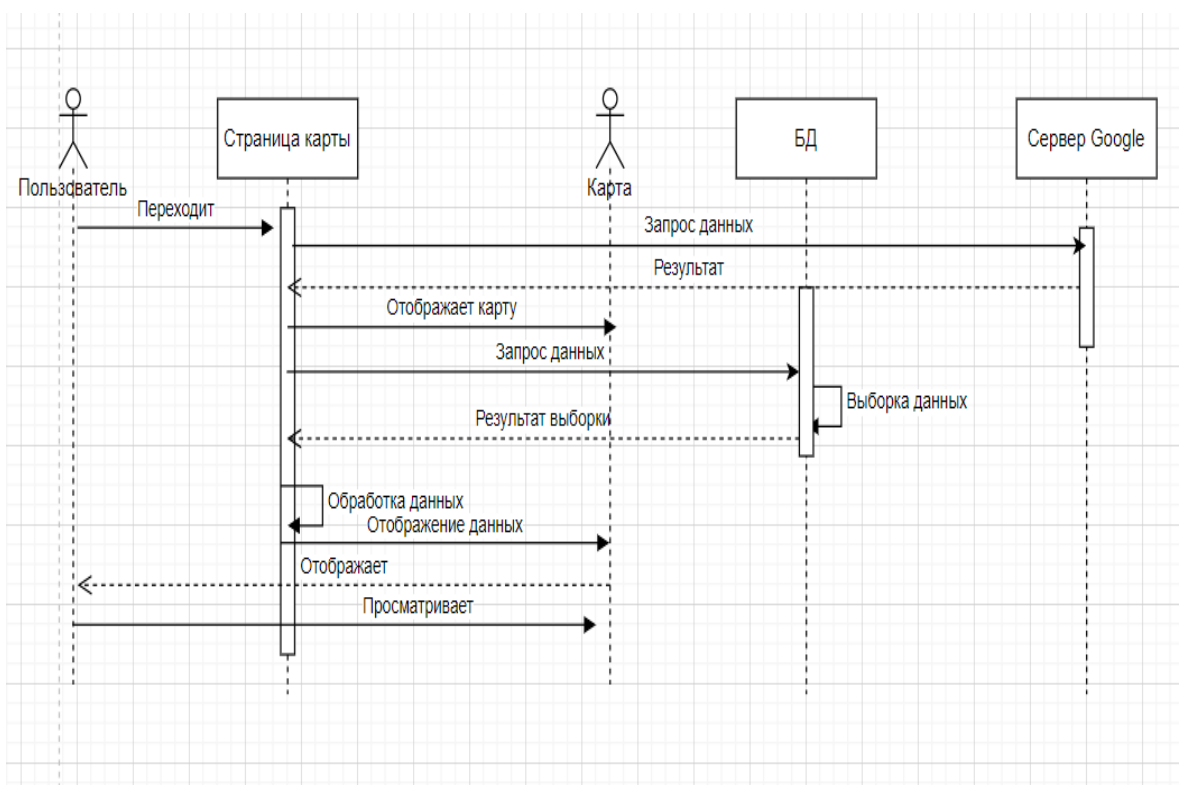


Рисунок 2.4.2-Диаграмма последовательности, карта

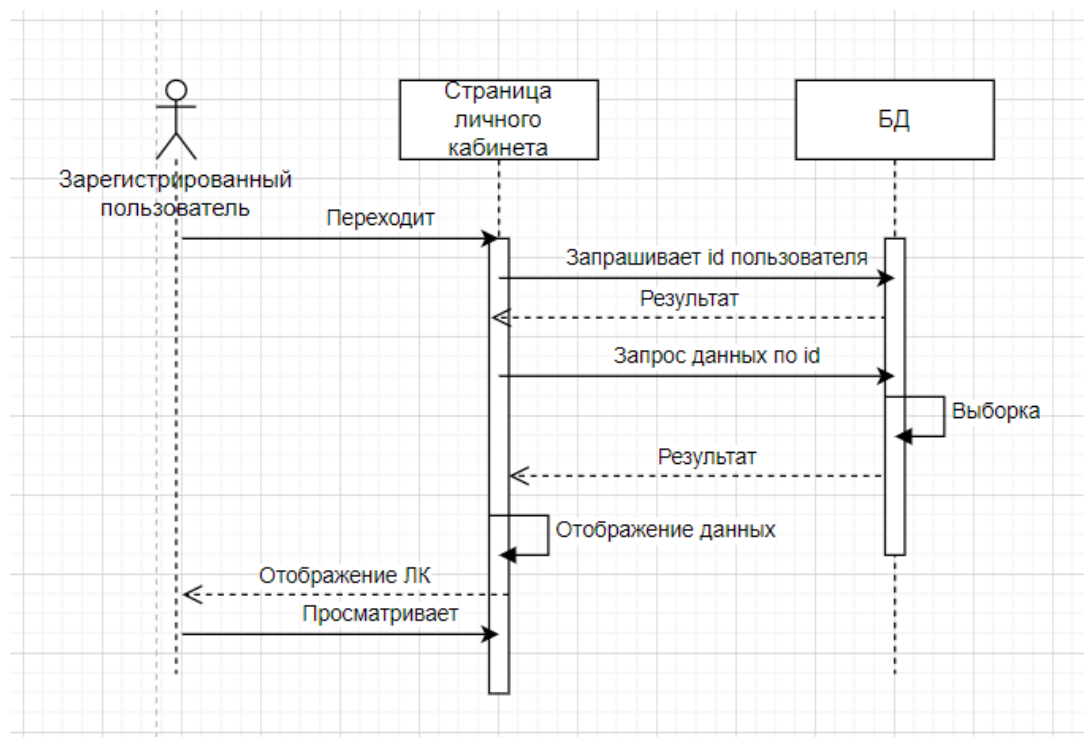


Рисунок 2.4.3 - Диаграмма последовательности, личный кабинет

Диаграмма классов

Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Диаграмма широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Данный вид диаграммы использован в проектирование приложения «PollutionHelper» для визуального представления иерархии классов системы, определения полей, интерфейсов и методов, диаграмма классов по частям представлена на рисунках 2.5.1 и 2.5.2, в общем виде находится в приложение В.

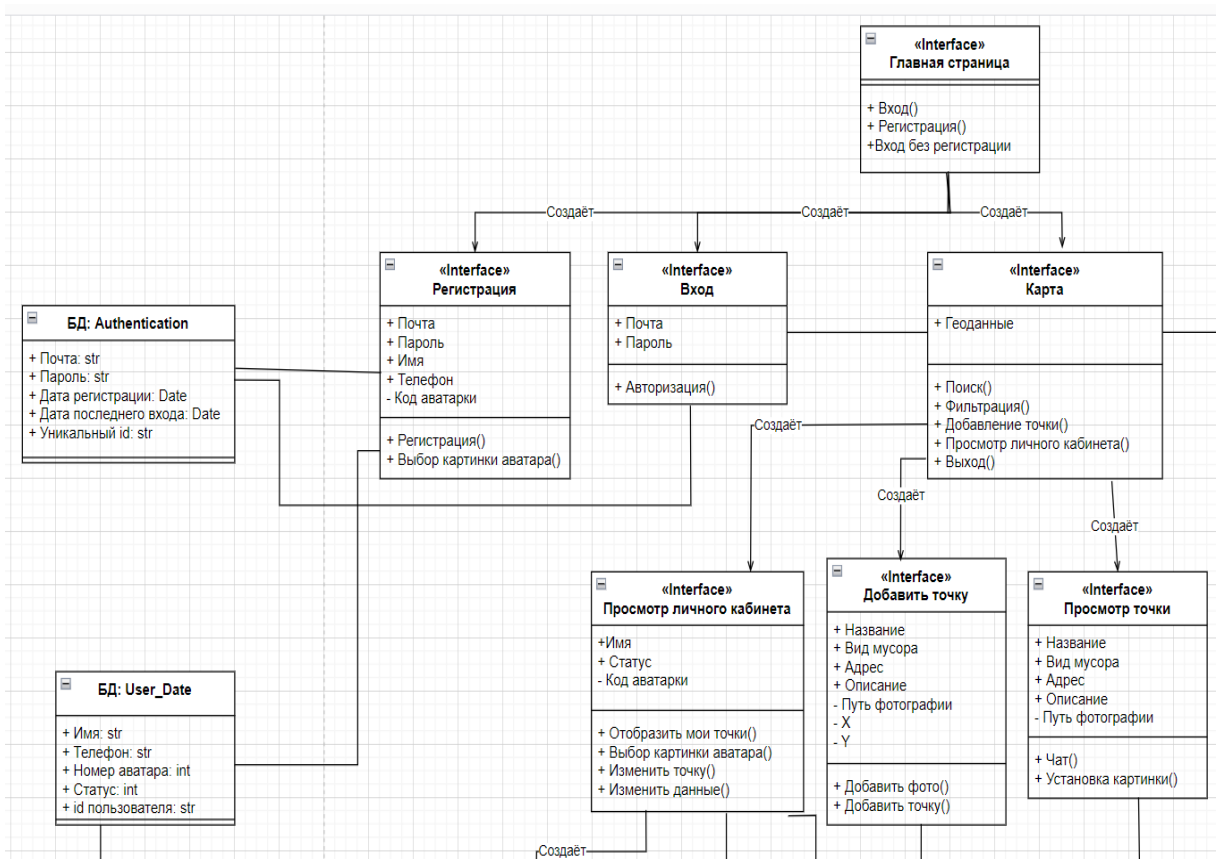


Рисунок 2.5.1 - Диаграмма классов для приложения «PollutionHelper»

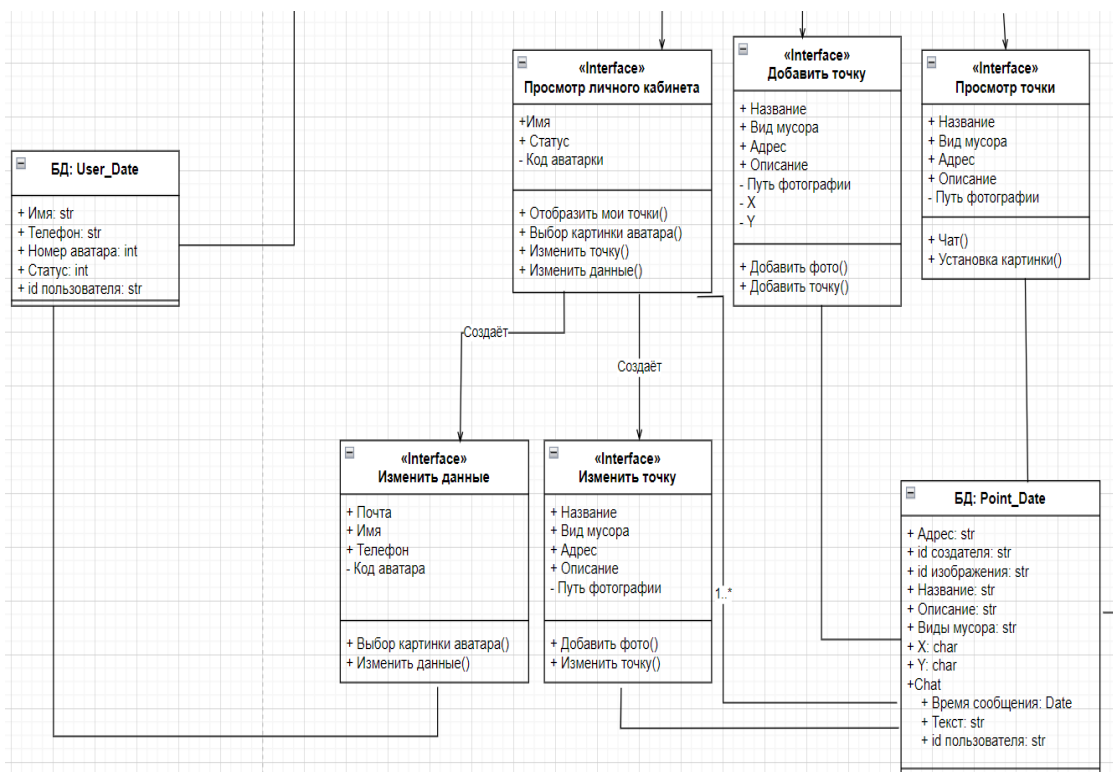


Рисунок 2.5.2 - Диаграмма классов для приложения «PollutionHelper»

Диаграмма развертывания

Данный вид диаграммы предназначен для визуального представления архитектуры исполняемой системы. Используя его можно понять, как система будет физически развернута на аппаратном обеспечении.

Для разрабатываемого мобильного приложения диаграмма развертывания является заключительным шагом проектирования и позволяет определить аппаратную топологию системы.

Диаграмма представлена на рисунке 2.6.

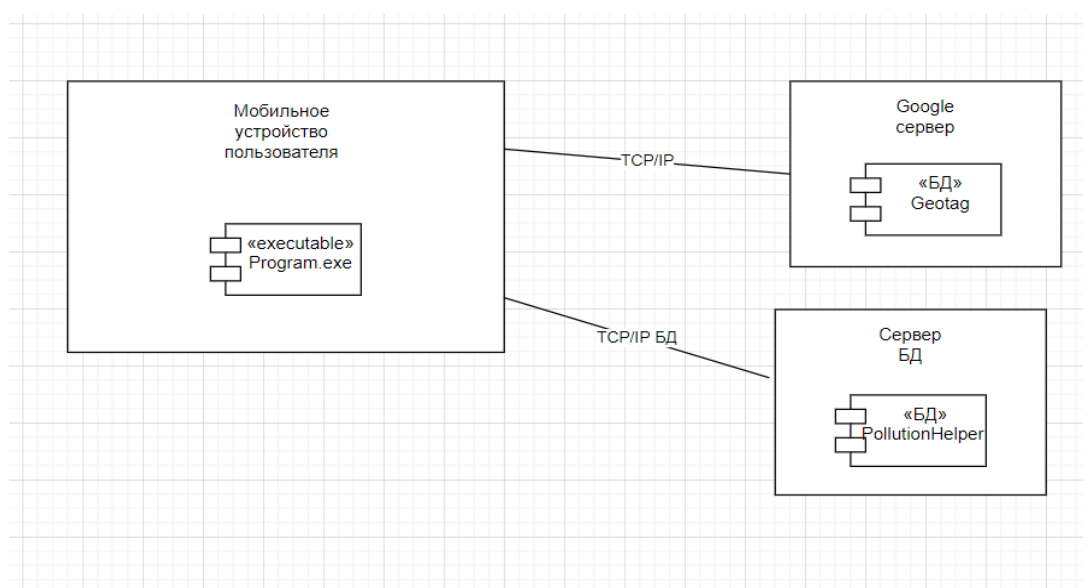


Рисунок 2.6 -Диаграмма развёртывания

Использование методологий проектирования (концептуальная, функциональная, объектно-ориентированная) позволило разработать схему проектирования геоинформационной системы и определить последовательность и методы реализации проекта.

Глава 3 Реализация

В главе приведено описание процесса создания рабочего прототипа мобильного приложения «PollutionHelper», с реализацией всего запланированного функционала, выбор которого теоретически обоснован в предыдущих главах.

3.1 Обоснование выбора инструментальных и программных средств

Раздел посвящен описанию средств разработки проекта:

- среда разработки;
- язык программирования;
- используемые для хранения информации базы данных.

и обоснованию использования выбранных средств.

3.1.1 Среда разработки

В качестве среды разработки мобильного приложения выбрана интегрированная среда разработки AndroidStudio. Это официальная IDE, которую поддерживает Google. Она позволяет разрабатывать приложения для смартфонов, планшетов, часов, телевизоров и других устройств на ОС Android.

AndroidStudio является бесплатно распространяемым программным продуктом, который может быть загружен с официального сайта разработчика. Данная среда разработки доступна для работы под наиболее распространенными операционными системами: Windows, macOS и GNU/Linux (русские ОС разрабатываются на ядре Linux).

Недостатком AndroidStudio для российского пользователя является отсутствие адаптации под русский язык (для работы необходимо знание английского языка).

Немаловажным преимуществом выбранной среды разработки является наличие встроенных эмуляторов Android девайсов, которые позволяют проверять и тестировать корректную работу разрабатываемого приложения на персональном компьютере со штатной операционной системой без использования устройств с операционной системой Android.

Основные преимущества AndroidStudio заключаются в следующем:

- официально поддерживаемая среда разработки;
- бесплатное распространение в интернете;
- доступность для Windows, macOS и GNU/Linux;
- удобный в работе редактор кода;
- возможность разработки приложений не только для смартфонов/планшетов, а также для телевизоров AndroidTV, устройств на операционной системе AndroidWear, новомодных мобильных устройств с необычным соотношением сторон экрана;
- тестирование корректности работы новых приложений, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе;
- рефакторинг уже готового кода;
- достаточно большая библиотека с готовыми шаблонами и компонентами для разработки приложений;
- разработка приложений для Android N – самой последней версии операционной системы;
- возможность предварительной проверки/тестирования уже созданного приложения на предмет ошибок в нем;
- большой набор инструментальных средств, для тестирования каждого элемента приложения;

- для неопытных/начинающих разработчиков специально создано руководство по использованию AndroidStudio, размещенное на официальном сайте разработчика.

Необходимо отметить заметный недостаток AndroidStudio.

Как известно, в мире доминируют две операционные системы для мобильных устройств: ОС Android с любыми оболочками и IOS. По статистике на 2021 год 71% мирового сообщества используют Android смартфоны, 29% предпочитают IOS смартфоны. AndroidStudio не поддерживает разработку приложений под IOS, и это сокращает область реализации разработанного приложения почти на 30%.

Проанализировав достоинства и недостатки среды разработки AndroidStudio можно с уверенностью утверждать, что она полностью подходит для реализации проекта, который изначально предполагал работу только под одной операционной системой. Наличие же большого объёма справочной информации по работе с AndroidStudio позволяет оперативно решать возникающие проблемы и исправлять всплывающие ошибки.

3.1.2 Язык программирования

Среда разработки AndroidStudio предлагает возможность использования для работы двух языков программирования: Java или Kotlin.

Язык программирования Java – мультифункциональный объектно-ориентированный язык со строгой типизацией, один из самых популярных в мире языков программирования: он был создан в 1995 году, быстро набрал популярность и уже много лет её не теряет.

Основные плюсы языка Java:

- независимость — код будет работать на любой платформе, которая поддерживает Java;

- надёжность — в немалой мере достигается благодаря строгой статической типизации;
- мультифункциональность;
- сравнительно простой синтаксис;
- Java — основной язык для Android-разработки.

Основные минусы языка Java:

- низкая скорость;
- требует большой объем памяти;
- нет поддержки низкоуровневого программирования;
- для ООП нужен опыт, а планирование новой программы занимает много времени.

Язык программирования Kotlin - это статически типизированный язык программирования, первая официальная версия которого была выпущена в 2016 году. В отличие от Java, где программы строятся на классах, основным строительным блоком программы на Kotlin является функция. Однако Kotlin также поддерживает объектно-ориентированный подход к программированию.

Основные плюсы языка Kotlin:

- Kotlin полностью совместим с Java, что означает возможность вызывать функции, написанные в Java-коде, прямо из кода Kotlin и наоборот, и, в том числе, пользоваться большим количеством уже существующих библиотек на Java;
- в программе на Kotlin можно использовать любые Java-фреймворк;
- лаконичность;
- Kotlin-безопасен, то есть создатели языка сделали всё, чтобы NullPointerException остались в прошлом (NullPointerException одни из самых часто встречающихся исключений, и их почти невозможно отследить во время компиляции, а это значит, что ошибки будут возникать только во время работы программы);

- наличие функций-расширения — позволяют создавать видимость внедрения в существующий класс нового метода;
- Kotlin имеет открытый исходный код.
Основные минусы языка Kotlin:
 - часто возникающие проблемы со скоростью компиляции кода;
 - малое количество туториалов. Сообщество разработчиков на языке программирования Kotlin еще молодое и статей по разработке не много;
 - небольшое количество библиотек на Kotlin.

При отсутствии опыта работы с Java и Kotlin большое значение имеет объём справочной информации по языку на интернет ресурсах и в видеоуроках. Как указано выше, язык программирования Kotlin относительно молодой, поэтому объём справочной информации по языку Kotlin значительно уступает языку Java.

Сравнение языков программирования Java и Kotlin позволило принять решение об использовании при реализации проекта языка Java. Так как этот язык обладает рядом преимуществ, для решения поставленной в ВКР задачи, таких как: надёжность; мультифункциональность; общепризнанность.

3.1.3 База данных

На выбор базы данных оказал влияние выбор среды разработки.

Для среды разработки AndroidStudio имеется встроенная база данных Firebase.

FirebaseDatabase - это облачный сервис, который предоставляет доступ к NoSQL базе данных, а также имеет удобные средства и методы взаимодействия с ней.

Firebase хранит текстовые данные в JSON формате и предоставляет удобные методы для чтения, обновления и извлечения данных. Также,

Firebase предоставляет функционал для регистрации и авторизации пользователей, хранения сессий, медиафайлов, к которым с легкостью предоставляет доступ благодаря CloudStorage.

Являясь встроенной базой для AndroidStudio, Firebase полностью совместим с этой средой разработки.

Основные плюсы базы данных Firebase:

- бесплатный начальный план;
- высокая скорость разработки;
- сквозная платформа для разработки приложений;
- работает на платформе Google;
- не требует использования сервера;
- мониторинг ошибок;
- безопасность.

Основные минусы базы данных Firebase:

- не имеет открытый исходный код;
- доступны только базы данных NoSQL;
- медленные запросы;
- не все службы работают бесплатно на базовом тарифе;
- это не дешевая платформа со сложно прогнозируемой ценой;
- работает только в Google Cloud.

Проанализировав достоинства и недостатки FirebaseDatabase можно сделать вывод, что она удовлетворяет требованиям проекта, а ее недостатки не оказывают существенного влияния на его работу. Простая интеграция в проект облегчает разработку, а наличие большого количества справочных материалов на форумах по данной тематике помогает справиться с подключением базы данных разработчику, никогда ранее не занимавшегося этой задачей.

3.2 Интерфейс мобильного приложения «PollutionHelper»

Раздел посвящён детальному описанию следующих компонентов разработанной системы:

- визуальные компоненты;
- элементы пользовательского интерфейса;
- функции, реализуемые системой;
- инструкции по работе с системой.

После загрузки приложения «PollutionHelper» открывается начальная страница приложения, вид которой зависит от статуса пользователя:

- пользователь авторизован (режим 1);
- пользователь не авторизован (режим 2).

Режим 1 (Рисунок 3.1.1) отображения начальной страницы активируется для пользователя, впервые вошедшего в приложение или сделавшего выход (LogOut) при последнем завершение сессии.

В этом режиме на начальной странице отображаются три функциональные кнопки:

- «Войти» – нажатие кнопки переводит пользователя на страницу «Вход»;
- «Регистрация» – нажатие кнопки переводит пользователя на страницу «Регистрация»;
- «Войти без авторизации» – нажатие кнопки позволяет пользователю перейти на страницу с картой и пользоваться ограниченными функциями приложения без авторизации в системе.



Рисунок 3.1.1 - Начальная страница, режим 1

Режим 2 (Рисунок 3.1.2) отображения начальной страницы активируется для пользователя, ранее авторизованного в приложении и не производившего отказ от авторизации (LogOut).

В этом режиме на начальной странице отображаются две функциональные кнопки:

- «Начать» – нажатие кнопки позволяет пользователю продолжить работать в системе с данными последней авторизации;

- «Выход» – нажатие кнопки позволяет выполнить отказ от авторизации (LogOut) в системе, что приведет к переходу начальной страницы к отображению в режиме 1.

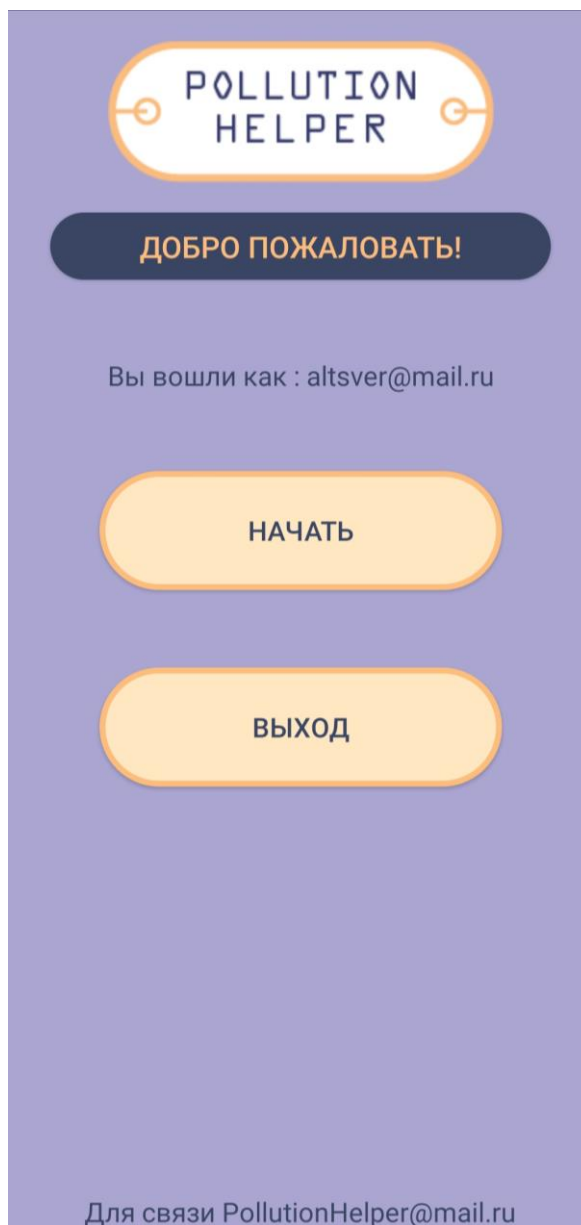


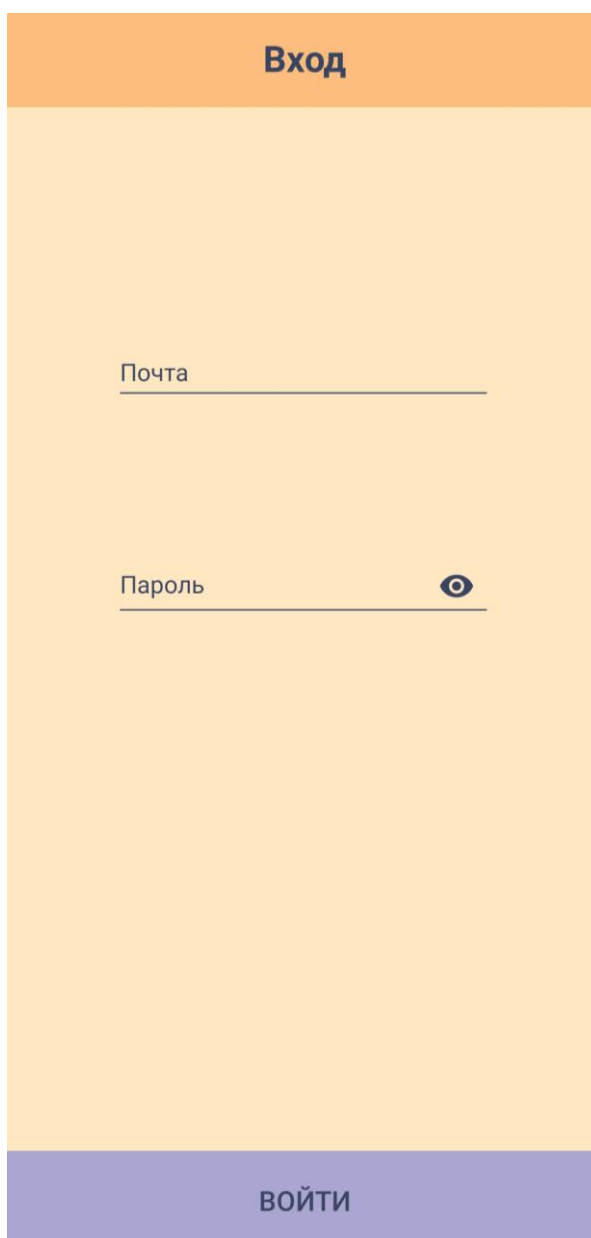
Рисунок 3.1.2 - Начальная страница, режим 2

Страница «Вход» (Рисунок 3.2) предназначена для авторизации пользователя в системе.

На странице отображаются:

- два поля ввода: «Почта» и «Пароль»;
- функциональная кнопка «Войти».

Нажатие кнопки «Войти» инициирует процесс авторизации пользователя в базе пользователей. После нажатия кнопки проводится проверка заполнения полей ввода. При некорректности введенных данных будет выдано сообщение об ошибке заполнения поля/полей ввода и запрос на авторизацию выполняться не будет. Если поля заполнены корректно, то система формирует запрос на авторизацию. При возникновении после обработки запроса в базе данных ошибки авторизации, будет выдано сообщение об ошибке, с предложением проверить вводимые данные. При успешной авторизации производится автоматический переход на страницу с картой.



Вход

Почта

Пароль

ВОЙТИ

Рисунок 3.2 -Страница входа

Страница «Регистрация» (Рисунок 3.3) предназначена для регистрации пользователя в системе.

На странице отображаются:

- четыре обязательных для заполнения поля ввода: «Почта», «Пароль», «Имя пользователя», «Телефон»;
- выпадающий список «Аватарка»;
- функциональная кнопка «Регистрация».

Нажатие кнопки «Регистрация» инициирует процесс регистрации пользователя в базе пользователей. После нажатия кнопки проводится проверка заполнения полей ввода. При некорректности введенных данных будет выдано сообщение об ошибке заполнения поля/полей ввода и запрос на регистрацию в системе выполняться не будет. Если поля заполнены корректно, то формируется запрос на регистрацию в системе. При совпадении данных почты с почтой ранее зарегистрированного в системе пользователя, будет выдано сообщение об ошибке регистрации, с предложением откорректировать данные. Если данные корректны, осуществляется автоматический переход на начальную страницу, а на указанную почту высылается письмо для подтверждения регистрации, информация об этом выводится во всплывающем сообщении. После ответа на письмо процедура регистрации завершается, и пользователь может осуществить вход в систему с авторизацией.

Регистрация

Аватарка ▼

Почта

Пароль 

Имя

Телефон 0/11

РЕГИСТРАЦИЯ

Рисунок 3.3 -Страница регистрации

Страница с картой (Рисунок 3.4.1) является главной для приложения «PollutionHelper». К ней привязан основной функционал системы.

В нижней части страницы отображаются четыре функциональные кнопки:

- «Личный кабинет» – нажатие кнопки переводит пользователя на страницу личного кабинета;
- «Фильтр» – нажатие кнопки приводит к открытию окна фильтрации на карте (Рисунок 3.4.2), позволяющей отобразить на карте маркеры по

интересующему виду мусора, путём выбора соответствующего элемента из списка и нажатия кнопки «Фильтровать»;

- «Поиск» – нажатие кнопки позволяет выполнить поиск места на карте, путём ввода интересующего адреса в появившееся поле ввода и повторного нажатия кнопки поиска, при этом в найденную точку выставляется маркер, цвет которого отличается от цвета остальных маркеров (Рисунок 3.4.3);
- «Выход» – нажатие кнопки позволяет выполнить отказ от авторизации (LogOut) в системе, что приводит к переходу на начальную страницу приложения в режиме 1.

В центральной области страницы расположен элемент fragment, к которому подключена Google карта. На карте отображаются маркеры, отмечающие места сбора мусора.

В функционал карты входит:

- определение места положения пользователя с помощью геолокации с помощью кнопки, расположенной в правом верхнем углу области отображения карты (Рисунок 3.4.1);
- определения направления на север с помощью кнопки, расположенной в левом верхнем углу области отображения карты (Рисунок 3.4.1);
- изменение масштаба карты с помощью специальных кнопок в правом нижнем углу области отображения карты (Рисунок 3.4.1) или стандартным для устройства образом;
- добавление маркера на карту.

Добавление маркера на карту производится длительным нажатием на выбранную область. Отображение маркера на карте может быть произведено в двух режимах:

- Режим 1 (Рисунок 3.4.4). Маркер отображается на карте из базы данных. В этом случае, при выборе маркера, над ним открывается информационное окно, отображающее название места по сбору мусора

и вид мусора, который собирается в этом месте. При нажатии на открывшееся окно осуществляется автоматический переход на страницу информации о точке, для более детального ознакомления с местом по сбору мусора с возможностью использования чата.

- Режим 2 (Рисунок 3.4.5). Новый маркер добавлен на карту пользователем. В этом случае, при выборе маркера на карте, открывается информационное окно, в котором отображается наименование «Новый маркер» и координаты точки. При нажатии на открывшееся окно осуществляется автоматический переход на страницу «Добавление», для добавления маркера в базу маркеров, что позволит другим пользователям системы пользоваться введенной информацией.

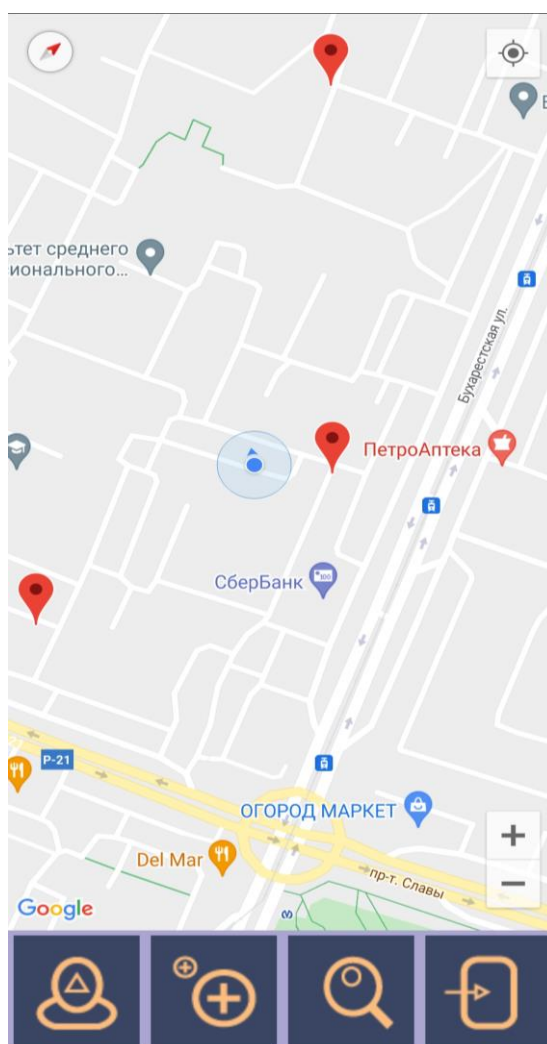


Рисунок 3.4.1 -Общий вид страницы с картой

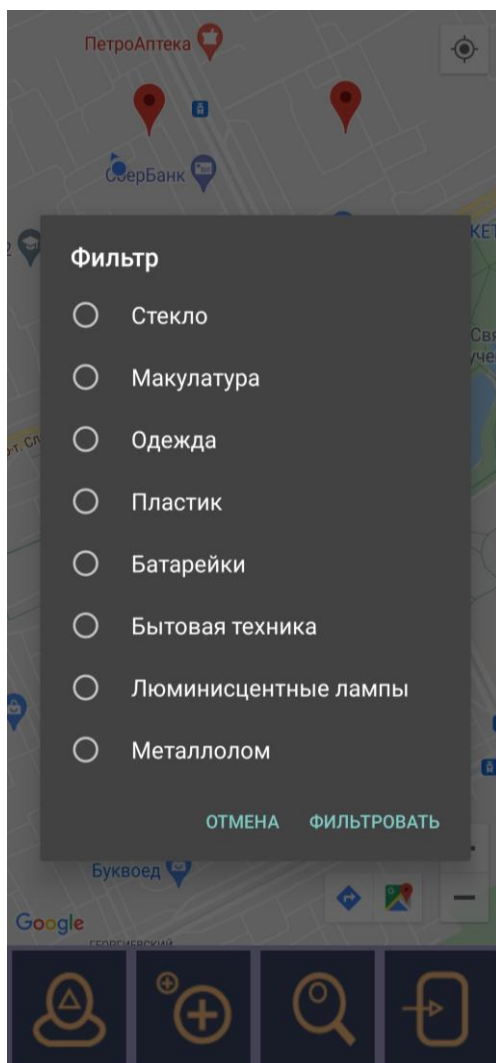


Рисунок 3.4.2 -Окно фильтра на карте

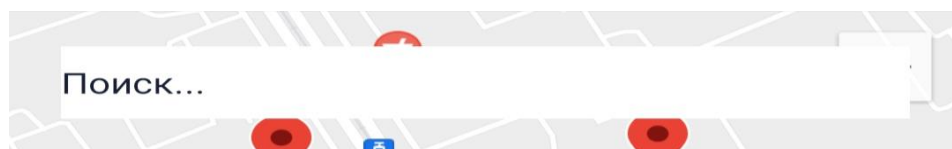


Рисунок 3.4.3 -Поиск на карте

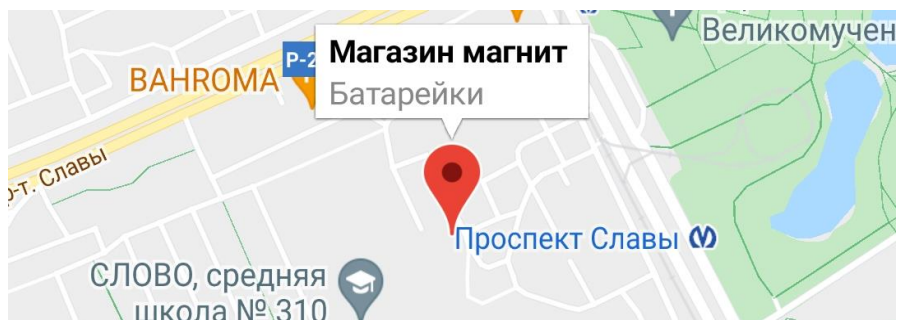


Рисунок 3.4.4 - Окно маркера, режим 1

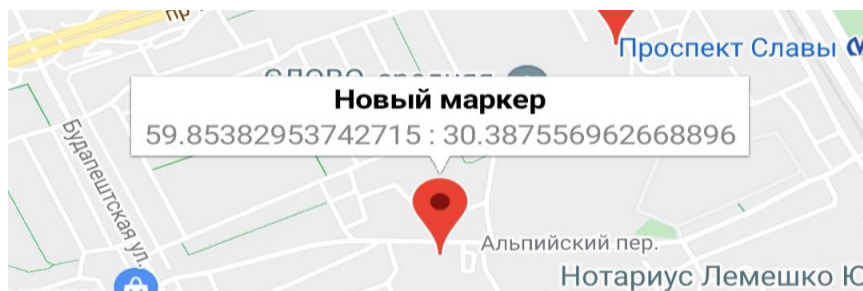


Рисунок 3.4.5 - Окно маркера, режим 2

Страница личного кабинета (Рисунок 3.5), которая доступна только для авторизованных пользователей, предназначена для отображения основных данных пользователя. На странице отображаются:

- картинка аватара, которая может быть выбрана/изменена на страницах «Регистрация» и «Изменение». Картинка является кнопкой, при нажатии на которую осуществляется автоматический переход на страницу «Изменение»;
- две текстовые строки, содержащие информацию об имени, под которым зарегистрирован пользователь и статусе пользователя, который зависит от количества добавленных в базу маркеров точек сбора мусора;
- список «Мои отметки» добавленных пользователем в базу маркеров точек сбора мусора, при нажатии на элемент списка осуществляется автоматический переход на страницу «Изменение точки».

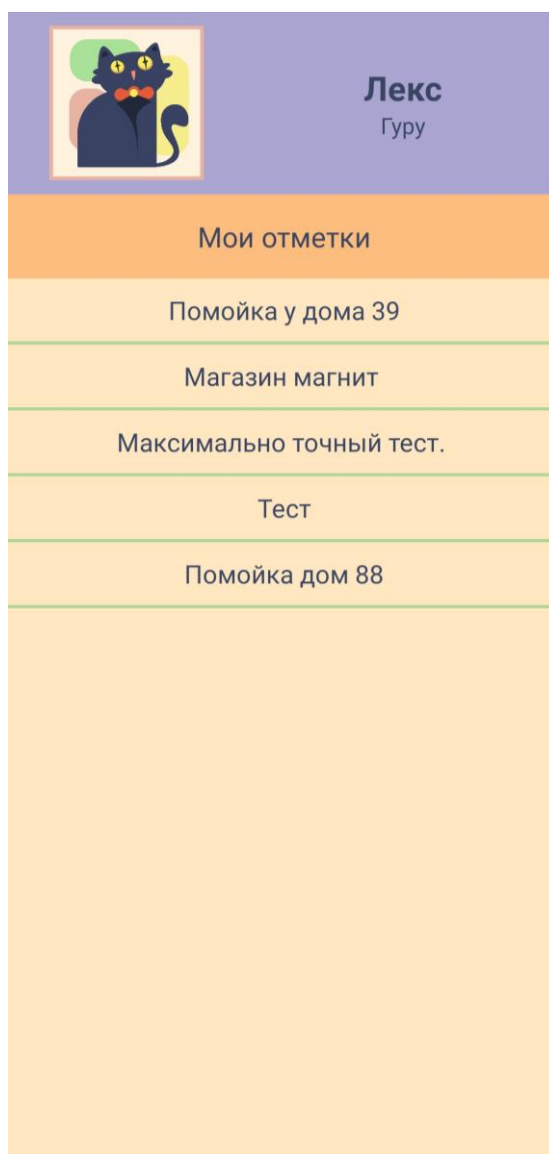


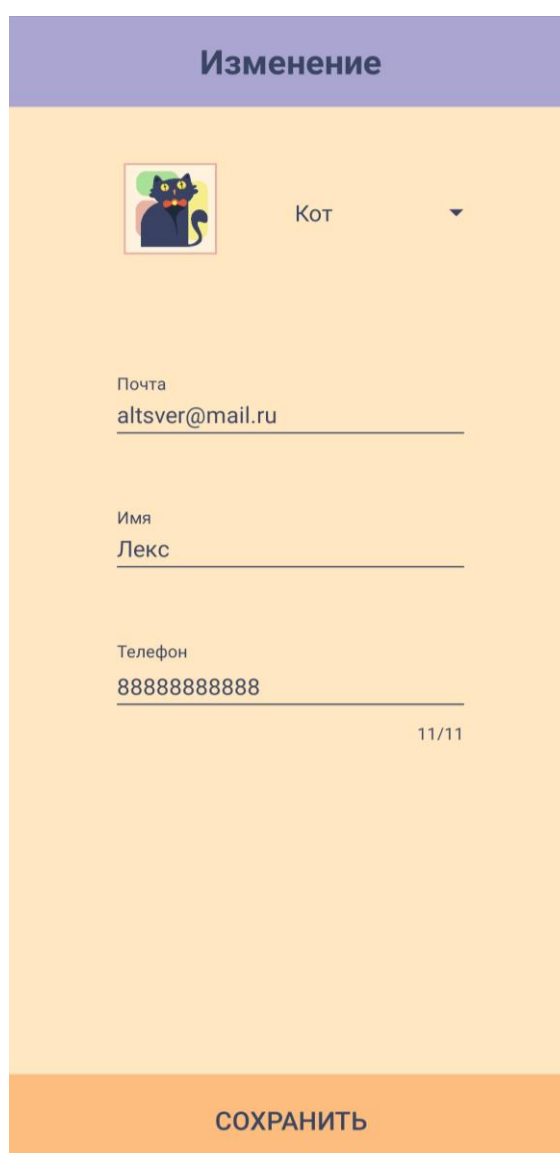
Рисунок 3.5 -Страница личного кабинета

Страница «Изменения» (Рисунок 3.6) предназначена для изменения личных данных пользователя, зарегистрированного в системе. На странице отображаются:


- картинка аватара, актуальная на момент перехода на страницу;
- выпадающий список выбора аватара, число элементов которого напрямую зависит от статуса пользователя (при повышении статуса открываются новые элементы выбора);
- информационная строка «Почта», содержащая информацию о почте, использованной для регистрации пользователя в системе;
- два поля ввода «Имя», «Телефон», которые могут быть изменены;

- функциональная кнопка «Сохранить».

Нажатие кнопки «Сохранить» инициирует процесс сохранения внесенных изменений в базе пользователей. После нажатия кнопки проводится проверка заполнения полей ввода. При некорректности введенных данных будет выдано сообщение об ошибке заполнения поля/полей ввода и сохранение изменений выполняться не будет. Если поля заполнены корректно, то производится сохранение изменений в базе пользователей и автоматический переход на страницу личного кабинета.



Изменение

 Кот ▾

Почта
altsver@mail.ru

Имя
Лекс

Телефон
8888888888 11/11

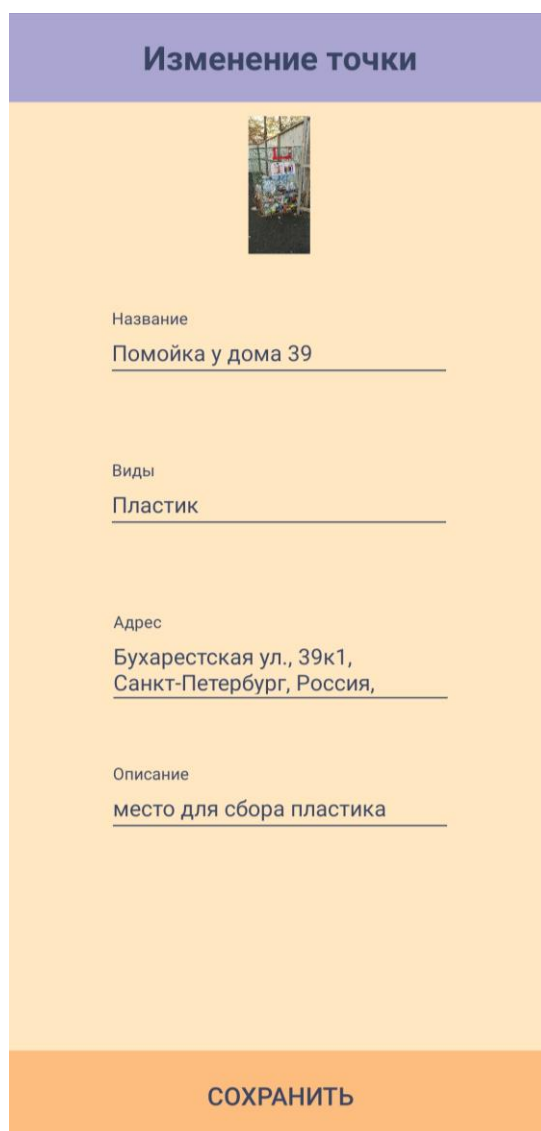
СОХРАНИТЬ

Рисунок 3.6 Страница изменения данных


Страница «Изменение точки» (Рисунок 3.7) предназначена для изменения данных добавленной пользователем точки. На странице отображаются:

- изображение точки сбора мусора, загруженное пользователем. Изображение является кнопкой, при нажатии на которую инициируется стандартный для устройства процесс загрузки изображения;
- информационная строка «Виды»;
- три поля ввода «Название», «Адрес» и «Описание»;
- функциональная кнопка «Сохранить».

Нажатие кнопки «Сохранить» инициирует процесс сохранения внесенных изменений в базе маркеров. После нажатия кнопки проводится проверка заполнения полей ввода. При некорректности введенных данных будет выдано сообщение об ошибке заполнения поля/полей ввода и сохранение изменений выполняться не будет. Если поля заполнены корректно, то производится сохранение изменений в базе маркеров и автоматический переход на страницу с картой.



Изменение точки



Название
Помойка у дома 39

Виды
Пластик

Адрес
Бухарестская ул., 39к1,
Санкт-Петербург, Россия,

Описание
место для сбора пластика

СОХРАНИТЬ

Рисунок 3.7 -Страница изменение данных точки

Страница «Добавление» (Рисунок 3.8), доступная только для авторизованных пользователей, предназначена для добавления точки в базу маркеров, с целью дальнейшего её отображения на карте. На странице отображаются:

- три поля ввода: «Название», «Адрес» – адрес точки в текстовом представлении (изначально определяется автоматически по координатам с помощью сервиса Google), «Описание» – дополнительная информация о точке;
- группа независимых переключателей для выбора видов обрабатываемого мусора, собираемого в точке;

- кнопка для загрузки изображения точки сбора мусора с устройства пользователя. Нажатие кнопки инициирует стандартный для устройства процесс загрузки изображения;
- функциональная кнопка «Добавить».

Нажатие кнопки «Добавить» инициирует процесс сохранения данных новой точки сбора мусора в базе маркеров. После нажатия кнопки проводится проверка заполнения полей ввода. При некорректности введенных данных будет выдано сообщение об ошибке заполнения поля/полей ввода и сохранение изменений выполняться не будет. Если поля заполнены корректно, то производится сохранение изменений в базе маркеров и автоматический переход на страницу с картой.

Добавление

Название

Стекло Батарейки

Макулатура Бытовая техника

Одежда Люминисцентные лампы

Пластик Металлолом

Адрес
Альпийский пер., 33-1,
Санкт-Петербург, Россия,
192286

Описание

ДОБАВИТЬ

Рисунок 3.8 -Страница добавление точки

Страница информации о точке (Рисунок 3.9), предназначена для отображения информации о точке сбора мусора для всех пользователей приложения и для отправки сообщений зарегистрированным пользователям (пользовательский чат). Отправка сообщения разрешена только авторизованным пользователям. Просмотр сообщений чата разрешен всем пользователям. Чат предназначен для предложений по корректировке информации о точке сбора мусора.

Страница информации о точке состоит из двух областей:

- область отображения полной информации о точке;
- область пользовательского чата.

В заголовке страницы отображается название точки сбора мусора.

В области отображения полной информации о точке отображаются:

- изображение точки сбора мусора, загруженное пользователем;
- информационная строка «Сбор», информирующая о видах мусора, собираемых на точке;
- информационная строка «Адрес», содержащая адрес точки сбора мусора;
- информационная строка «Описание», содержащая дополнительное описание точки.

В области пользовательского чата отображаются:

- список сообщений пользователей, относящихся к текущей точке, в котором каждый элемент состоит из следующей информации: статуса пользователя, имени пользователя, даты/времени написания сообщения и самого сообщения пользователя;
- поле ввода для ввода текста сообщения;
- функциональная кнопка «Отправить».

Нажатие кнопки «Отправить» инициирует процесс сохранения сообщения в базе маркеров. После нажатия кнопки производится сохранение

изменений в базе маркеров, и сообщение отображается в списке сообщений, относящихся к текущей точке.

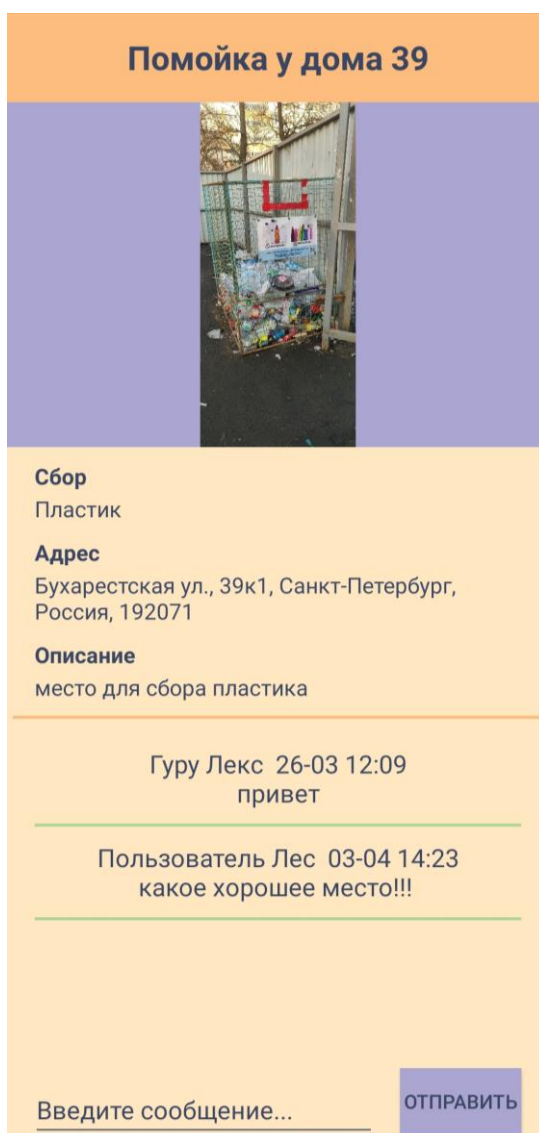


Рисунок 3.9 -Страница информация о точке

3.3 Система хранения данных

Для разработки приложения «PollutionHelper» была выбрана база данных Firebase, которая является комплексной платформой для разработки мобильных и веб-приложений. Платформа обладает набором определённых сервисов и служб, которые включены в продукт Firebase. В рамках разработки мобильного приложения были использованы следующие сервисы и службы.

1. Authentication – отвечает за регистрацию и авторизацию пользователей.
2. RealtimeDatabase – документоориентированная база данных, используется для хранения данных.
3. Storage – хранилище, встроенное в Firebase, предназначено для хранения и управления медиафайлами.

Реализация каждой службы в приложение «PollutionHelper» подробно описана ниже.

3.3.1 Authentication

Служба Authentication позволяет выбрать один и более способов регистрации нового пользователя (Рисунок 3.10).

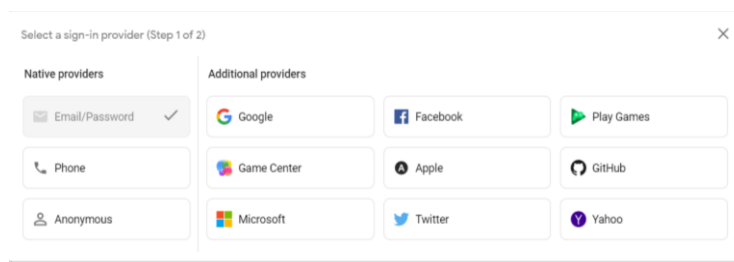
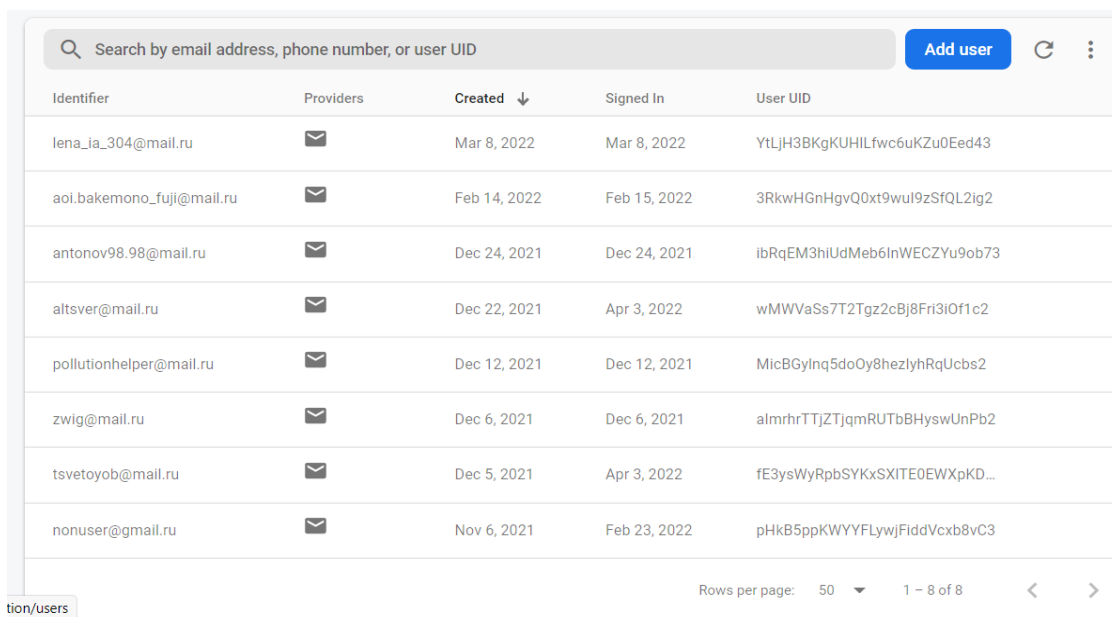


Рисунок 3.10 -Способы регистрации пользователя

В качестве способа регистрации для приложения «PollutionHelper» был выбран вариант Почта/Пароль, означающий, что пользователь для регистрации и авторизации использует почту и пароль.

Общий вид базы данных зарегистрированных пользователей приведен на рисунке 3.11.



Identifier	Providers	Created ↓	Signed In	User UID
lena_ia_304@mail.ru	✉	Mar 8, 2022	Mar 8, 2022	YtLjH3BKgKUHILfwc6uKZu0Eed43
aoi.bakemono_fujl@mail.ru	✉	Feb 14, 2022	Feb 15, 2022	3RkwHGnHgvQ0xt9wul9zSfQL2ig2
antonov98.98@mail.ru	✉	Dec 24, 2021	Dec 24, 2021	ibRqEM3hiUdMeb6InWECZYu9ob73
altsver@mail.ru	✉	Dec 22, 2021	Apr 3, 2022	wMwVaSs7T2Tgz2cBj8Fri3iO1fc2
pollutionhelper@mail.ru	✉	Dec 12, 2021	Dec 12, 2021	MicBGylnq5doOy8hezlyhRqUcbs2
zwig@mail.ru	✉	Dec 6, 2021	Dec 6, 2021	almrhrTTJZTjqmRUTbBHysUnPb2
tsvetoyob@mail.ru	✉	Dec 5, 2021	Apr 3, 2022	fE3ysWyRpbSYKxSXITE0EWXpKD...
nonuser@gmail.ru	✉	Nov 6, 2021	Feb 23, 2022	pHkB5ppKWYYFLywjFiddVcxb8vC3

Рисунок 3.11 - Таблица зарегистрированных пользователей

Как видно из таблицы, в базе данных приложения «PollutionHelper» хранится следующая информация:

- Identifier—при выбранном в проекте способе регистрации поле содержит данные о почте пользователя;
- Providers— поле содержит данные о способе регистрации;
- Created – поле содержит данные о дате регистрации пользователя;
- SignedIn – поле содержит данные о дате последней авторизации пользователя;
- UserUID – поле содержит индивидуальный идентификатор пользователя

Из состава информации базы данных видно, что пароль пользователя не хранится явно. Пароль сохраняется в хэшированном виде для защиты личных данных пользователя.

3.3.2 RealtimeDatabase

Служба RealtimeDatabase - основная служба для хранения данных в формате JSON. При таком хранении документоориентированные базы данных совместно используют базовую семантику доступа и поиск хранилищ ключей и значений. Для уникальной идентификации данных БД используют ключ.

Структура базы данных, используемой в приложении, представлена на Рисунках 3.12 – 3.14.

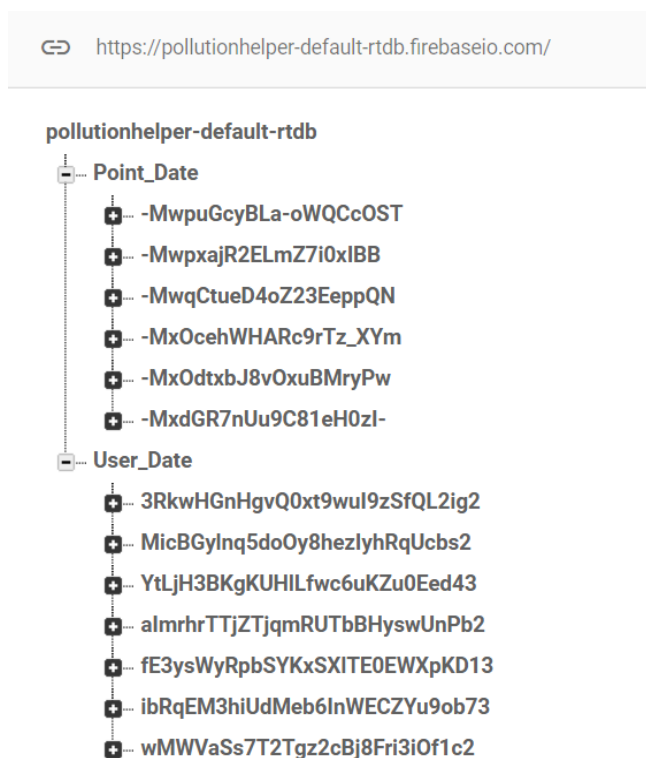


Рисунок 3.12 -Верхний уровень иерархии в БД

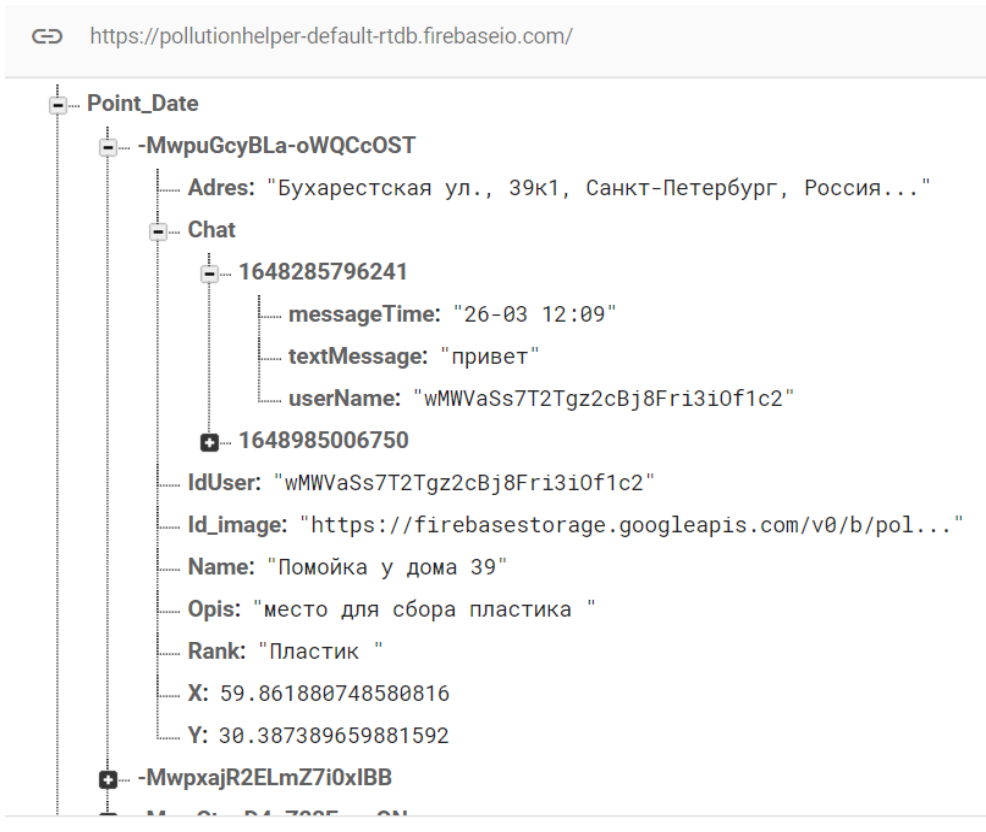


Рисунок 3.13 -Представление элемента раздела Point_Date



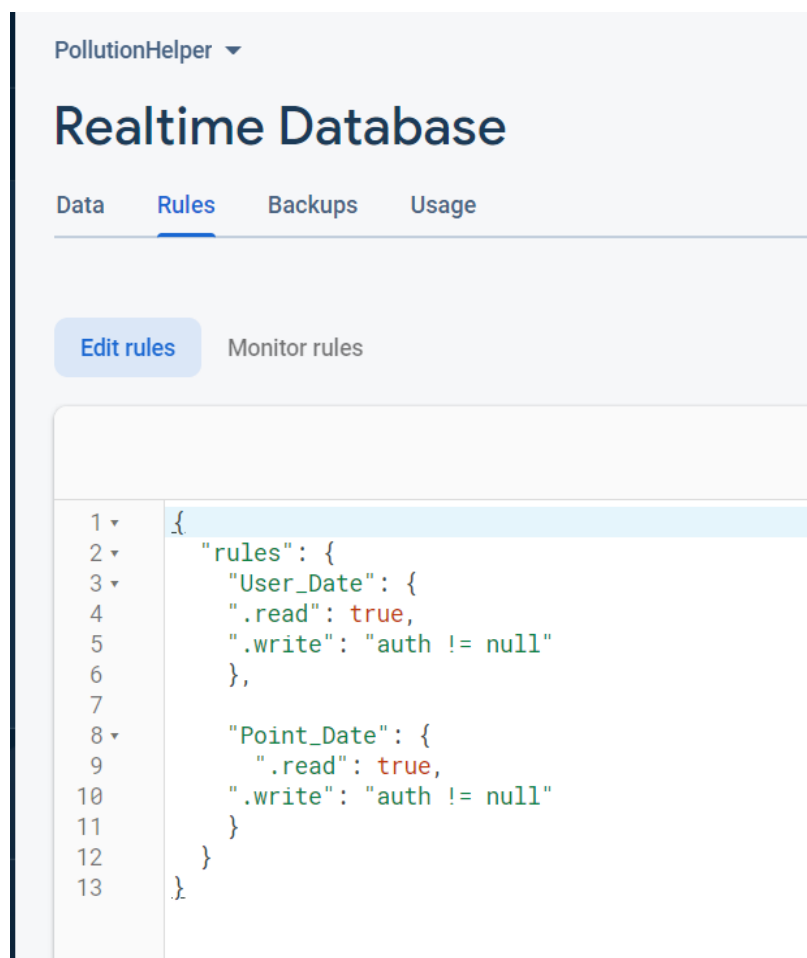
Рисунок 3.14 -Представления элемента раздела User_Date

Как видно из представленной информации, база данных приложения «PollutionHelper» состоит из двух главных разделов.

Point_Date (Раздел 1) – содержит данные о точке сбора мусора: адрес, название, описание, вид мусора, координаты, путь к фотографии, а так же пользовательский чат, относящийся к точке.

User_Date (Раздел 2) – содержит данные о пользователе: имя, телефон, данные об аватаре пользователя, статус пользователя, а так же id добавленных пользователем точек.

Для разграничения прав доступа (авторизованные и не авторизованные пользователи) необходима настройка правил использования базы. Для такой настройки в RealtimeDatabase существует специальный раздел - Rules, в котором объявляются правила работы с БД (Рисунок 3.15).



```
1 {
2   "rules": {
3     "User_Date": {
4       ".read": true,
5       ".write": "auth != null"
6     },
7
8     "Point_Date": {
9       ".read": true,
10      ".write": "auth != null"
11    }
12  }
13 }
```

Рисунок 3.15 -Правила базы данных

Для раздела User_Date приложения «PollutionHelper» действуют правила:

- чтение данных разрешено всем пользователям;
- запись данных разрешена только авторизованным пользователям.

Это позволяет пользователям, входящим без авторизации, видеть данные о других пользователях, например имя в чате, но запрещает вход в личный кабинет.

Для раздела Point_Date приложения «PollutionHelper» действуют аналогичные правила:

- чтение данных разрешено всем пользователям;
- запись данных разрешена только авторизованным пользователям

Это позволяет пользователям, входящим без авторизации, видеть точки на карте, производить фильтрацию и просматривать подробную информацию, но запрещает самим добавлять точки на карту и писать в чат.

Задание правил доступа приложения «PollutionHelper» хорошо иллюстрирует рисунок 3.15.

3.3.3 Storage

Сервис FirebaseStorage – хранилище используется для хранения медиафайлов. В приложении «PollutionHelper» сервис используется для хранения фотографий пользователя, загруженных с устройства (Рисунок 3.16). Благодаря записи пути к фотографии при загрузке и библиотеке Picasso, мобильное приложение может извлекать фотографии из хранилища и отображать их.

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	1645790441037my_image	951.82 KB	image/jpeg	Feb 25, 2022
<input type="checkbox"/>	1645794736520my_image	1.74 MB	image/jpeg	Feb 25, 2022
<input type="checkbox"/>	1645864967776my_image	713.63 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645872215776my_image	670.51 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645872363784my_image	670.51 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645872715768my_image	195.59 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645874436917my_image	195.59 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645876973158my_image	159.09 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645877828139my_image	195.59 KB	image/jpeg	Feb 26, 2022
<input type="checkbox"/>	1645882100519my_image	968.25 KB	image/jpeg	Feb 26, 2022

Рисунок 3.16 - Хранилище Firebase

3.3.4 Схема восстановления и сохранения информации

Firebase предоставляет возможность создавать резервную копию базы данных. Эта копия храниться в облаке, при необходимости может быть скачана и установлена. Так же Firebase создаёт копию правил использования базы данных (Рисунок 3.17), сохраняя при восстановлении базы права доступа.

Для приложения «PollutionHelper» настройка функции резервного копирования и восстановление БД доступны только администратору/разработчику приложения.

Database	Last backed up	Google Cloud Storage bucket	Daily
pollutionhelper-default-rtdb	0 minutes ago	pollutionhelper-default-rtdb-backups	–

Status	Files	Start time	Run time	Size
✓ Complete	2021-12-19T14:10:09Z_pollutionhelper-default-rtdb_data.json.gz 2021-12-19T14:10:09Z_pollutionhelper-default-rtdb_rules.json.gz	Dec 19, 2021, 5:10:09 PM	7s	459B
✓ Complete	2021-12-19T13:24:16Z_pollutionhelper-default-rtdb_data.json.gz 2021-12-19T13:24:29Z_pollutionhelper-default-rtdb_rules.json.gz	Dec 19, 2021, 4:24:16 PM	14s	459B

Рисунок 3.17 -Резервное копирование

По умолчанию, при первом подключение службы резервного копирования, оно осуществляется по следующим правилам:

- копирование осуществляется ежедневно;
- используется алгоритм сжатия Gzip для сокращения размера файла, что даёт возможность использовать для его хранения меньше пространства облака и быстрее передавать;
- копия удаляется через 30 дней для освобождения места в облаке.

Настройки могут быть изменены. Существует возможность создать резервную копию в ручном режиме. Данные хранятся в формате JSON.

Восстановление данных из резервной копии происходит по следующему алгоритму.

1. Скачать файл из облачного хранилища
2. Если файл сжат в Gzip необходимо его распаковать в JSON
3. В разделе Данные импортировать данный JSON файл

Общая схема восстановления и сохранения информации представлена на Рисунке 3.18.

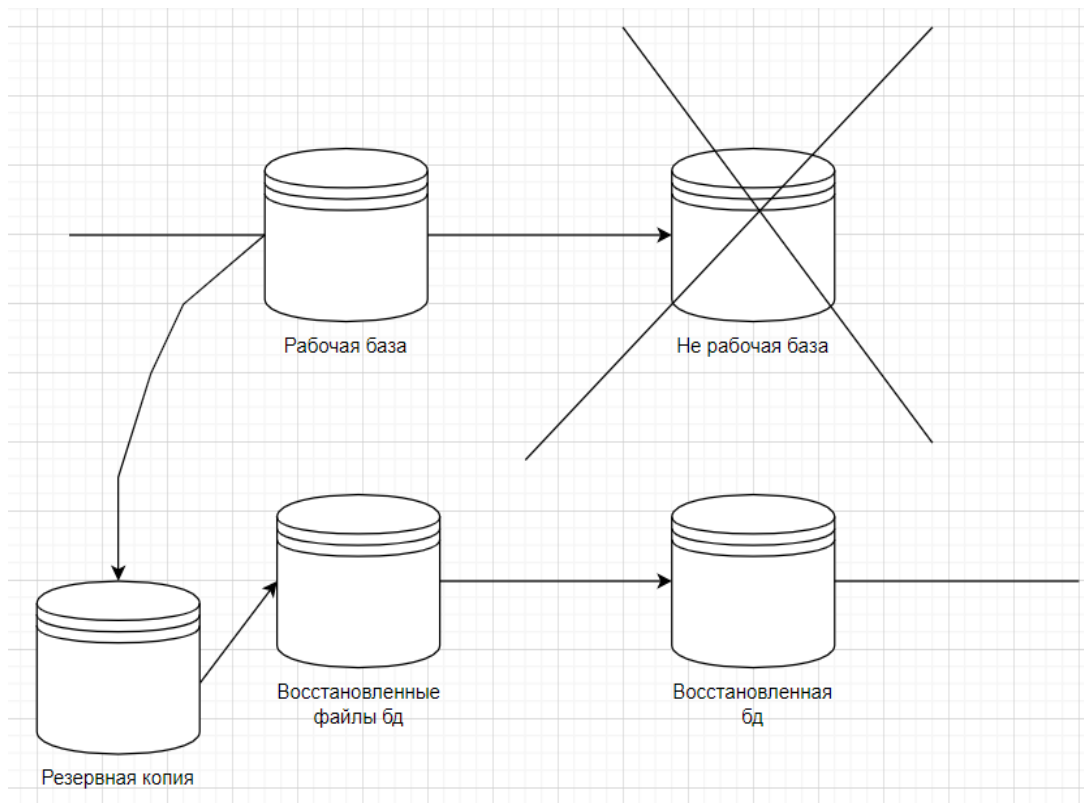


Рисунок 3.18 -Схема восстановления и сохранения информации

3.4 Надёжность системы

Теория надёжности относительно молодая наука. Возникла она тогда, когда стали строить сложные системы, состоящие из большого числа ненадёжных элементов [20].

Проблема обеспечения необходимой надёжности программных комплексов пока мало изучена, однако в настоящее время наблюдается устойчивая тенденция к расширению масштабов исследования в этой области. Даже приближенные методы позволяют получать ряд важных оценок, необходимых для принятия решения.

3.4.1 Расчёт структурной схемы надёжности

Структурная схема надёжности представляет собой условную запись или графическое изображение, позволяющее описать работоспособное состояние объекта через возможные состояния его составных частей, элементов с учетом их связей и функционального назначения.

Структурная схема системы «PollutionHelper» представлена на Рисунке 3.19.

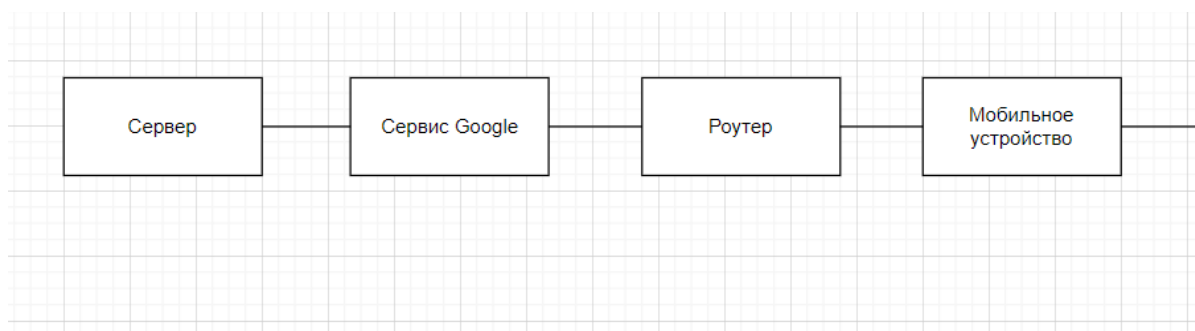


Рисунок 3.19 - Структурная схема системы

При выходе из строя любого элемента, система перестает работать, исходя из этого, схема представлена в виде последовательного соединения. Для расчёта надёжности системы, необходимо рассчитать надёжность каждого элемента системы.

Расчёт надёжности сервера

При расчёте надёжности принимаются следующие допущения:

- отказы устройств являются независимыми и случайными событиями;
- учитываются только устройства, входящие в сервер;
- вероятность безотказной работы подчиняется по экспоненциальному закону распределения;

Для элементов, используемых в сервере, приняты следующие интенсивности отказов (таблица 4):

Таблица 4 - Интенсивность отказов периферии сервера

Название элемента	Интенсивность отказа
Материнская плата	$\lambda_1=4.5 \times 10^{-8} \text{ ч}^{-1}$
Процессор	$\lambda_2=4.0 \times 10^{-7} \text{ ч}^{-1}$
Память	$\lambda_3=3.2 \times 10^{-7} \text{ ч}^{-1}$
Жесткий диск	$\lambda_4=8.3 \times 10^{-7} \text{ ч}^{-1}$
CD-ROM	$\lambda_5=0.1 \times 10^{-5} \text{ ч}^{-1}$
Дисковод 3.5"	$\lambda_6=0.04 \times 10^{-5} \text{ ч}^{-1}$
Сетевая карта	$\lambda_7=1.0 \times 10^{-7} \text{ ч}^{-1}$
Блок питания	$\lambda_8=2 \times 10^{-7} \text{ ч}^{-1}$

Исходя из этих значений, можно подсчитать суммарную интенсивность отказов всех устройств одного типа, а затем и для всех устройств сервера по формуле.

$$\lambda_{\text{общ}} = \sum_{i=1}^N \lambda_i, \text{ где} \quad (3.1)$$

N – количество элементов одного вида,

λ_i – интенсивность отказа i -того элемента.

Вероятность безотказной работы сервера:

$$P(t) = e^{-\lambda_{\text{общ}} t}, \text{ где} \quad (3.2)$$

t – время за которое рассматривается вероятность безотказной работы;

$\lambda_{\text{общ}}$ – общая интенсивность отказа элементов.

Подсчитаем суммарную интенсивность отказов всех устройств по формуле 3.1:

$$\lambda_{\text{общ}} = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 = 33.95 \times 10^{-7}$$

Вычисления вероятности безотказной работы сервера произведем по формуле 3.2. В рамках задания возьмем 1000 часов.

$$P(1000) = e^{-33.95 \times 10^{-7} \times 1000} = 0.9966;$$

Согласно произведенным расчетам, вероятность безотказной работы сервера в течение 1000 часов составляет 99.66%

Расчёт надёжности Роутера

Стандартный роутер обладает следующими элементами, представленными в таблице 5.

Таблица 5 - Интенсивность отказов периферии роутера

Название элемента	Интенсивность отказа
Центральный процессор	$\lambda_1=4.0 \times 10^{-7} \text{ ч}^{-1}$
Флэш-память	$\lambda_2=8.3 \times 10^{-7} \text{ ч}^{-1}$
ОЗУ	$\lambda_3=7.52 \times 10^{-6} \text{ ч}^{-1}$
Энергонезависимое ОЗУ	$\lambda_4=7.12 \times 10^{-6} \text{ ч}^{-1}$

Вычисление общей интенсивности отказа производится по формуле 3.1

$$\lambda_{\text{общ}} = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 15.87 \times 10^{-6}$$

Вычисление вероятности безотказной работы роутера производится по формуле 3.2. В рамках задания возьмем 1000 часов.

$$P(1000) = e^{-15.87 \times 10^{-6} \times 1000} = 0.9843$$

Согласно произведенным расчетам, вероятность безотказной работы роутера составляет в течение 1000 часов 98.43%

Расчёт надёжности мобильного устройства

В рамках исследования взяты усреднённые параметры и комплектующие мобильного устройства представленные в таблице 6.

Таблица 6 - Интенсивность отказов периферии смартфона

Название элемента	Интенсивность отказа
Процессор	$\lambda_1=4.0 \times 10^{-7} \text{ ч}^{-1}$
Материнская плата	$\lambda_2=4.5 \times 10^{-8} \text{ ч}^{-1}$
Основная память	$\lambda_3=8.3 \times 10^{-7} \text{ ч}^{-1}$
ОЗУ	$\lambda_4=7.52 \times 10^{-6} \text{ ч}^{-1}$
Батарея	$\lambda_5=7.2 \times 10^{-6} \text{ ч}^{-1}$
Модули	$\lambda_6=2.186 \times 10^{-6} \text{ ч}^{-1}$

Расчёт общей интенсивности отказа смартфона, с учётом трех модулей, производится по формуле 3.1.

$$\lambda_{\text{общ}} = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + 3 * \lambda_6 = 22.553 * 10^{-6}$$

Вычисления вероятности безотказной работы роутера производится по формуле 3.2. В рамках задания возьмем 1000 часов.

$$P(1000) = e^{-22.553 * 10^{-6} * 1000} = 0.9977$$

Согласно произведенным расчетам, вероятность безотказной работы мобильного устройства в течение 1000 часов составляет 99.77%.

Расчёт надёжности безотказной работы системы

В результате произведенных вычислений вероятности безотказной работы элементов схемы, были полученные следующие результаты:

- Вероятность безотказной работы сервера составляет 99.66%;
- Вероятность безотказной работы роутера составляет 98.43%;
- Вероятность безотказной работы смартфона составляет 99.77%;
- Вероятность безотказной работы Google сервиса будет принята за 99.99% в соответствии с официальными документами.

Структурная схема системы имеет вид последовательного соединения, для вычисления общей надёжности необходимо воспользоваться формулой:

$$P(t) = \prod_{i=1}^N P_i(t) = P_1(t) \times P_2(t) \times P_3(t) \times \dots, \text{ где} \quad (3.3)$$

t- время за которое рассматривается вероятность безотказной работы;

N - количество таких элементов;

P_i -вероятность безотказной работы i -го элемента.

$$P_{\text{системы}} = P_{\text{сервера}} * P_{\text{Google_сервиса}} * P_{\text{роутера}} * P_{\text{смартфона}} = 0.9786$$

В результате произведенных вычислений вероятность безотказной работы системы «PollutionHelper» в течение 1000 дней составляет 97.86%.

3.4.2 Расчёт надёжности программного обеспечения

Обеспечить надёжность работы программного обеспечения очень важная задача, для разработчика. Для того чтобы добиться высокой работоспособности программы, необходимо проводить тесты и заниматься отладкой. Это позволяет найти максимальное количество ошибок до официального выпуска программы. Методы расчёта надёжности играют большую роль в тестировании программы, так благодаря им, мы можем точно рассчитать надёжность различных систем.

В данной работе, для вычисления надёжности ПО, будет применён метод оценки надёжности основанный на модели Шумана [20].

Согласно методу, необходимо составить таблицу тестирования программы, где t_k - это длительность этапа тестирования, m_k - количество ошибок на этапе тестирования. Данные по проведённому тестированию представлены в таблице 7.

Таблица 7 -Тестирование приложения

Этапы	t, ч	m
1	10	3
2	15	1
3	15	1

Через метод подбора первоначальное количество ошибок $N=6$.

Коэффициент пропорциональности C рассчитывается по формуле:

$$C = \frac{\sum_{i=1}^k \frac{m_i}{N - n_{i-1}}}{\sum_{i=1}^k t_i}, \text{ где} \quad (3.4)$$

k – количество этапов тестирования;

N – первоначальное количество ошибок;

m_i – количество обнаруженных ошибок на i -том этапе тестирования;

n_{i-1} – количество ошибок обнаруженных на предыдущих этапах;

t_i – время проведения i -того этапа.

$$C = \frac{3/6 + 1/(3-6) + 1/(3-6-1)}{10 + 15 + 15}$$

$$C = 2.917 \cdot 10^{-2}$$

Интенсивность отказов рассчитывается по формуле:

$$\lambda = (N - n)C, \text{ где} \quad (3.5)$$

N – первоначальное количество ошибок;

n – количество найденных ошибок;

C – коэффициент пропорциональности.

$$\lambda = (6 - 3 - 1 - 1) \cdot 2.917 \cdot 10^{-2} = 2.917 \cdot 10^{-2}$$

Функция надёжности программы после завершения тестирования рассчитывается по формуле:

$$R(t) = e^{-\lambda t}, \text{ где} \quad (3.6)$$

t – время работы системы;

λ – интенсивность отказов.

Для мобильного приложения «PollutionHelper» функция надёжности имеет вид - $R(t) = e^{-2.917 \cdot 10^{-2} \cdot t}$. График функции во времени представлен на Рисунке 3.20. Он иллюстрирует, как изменяется вероятность безотказной работы приложения с течением времени.

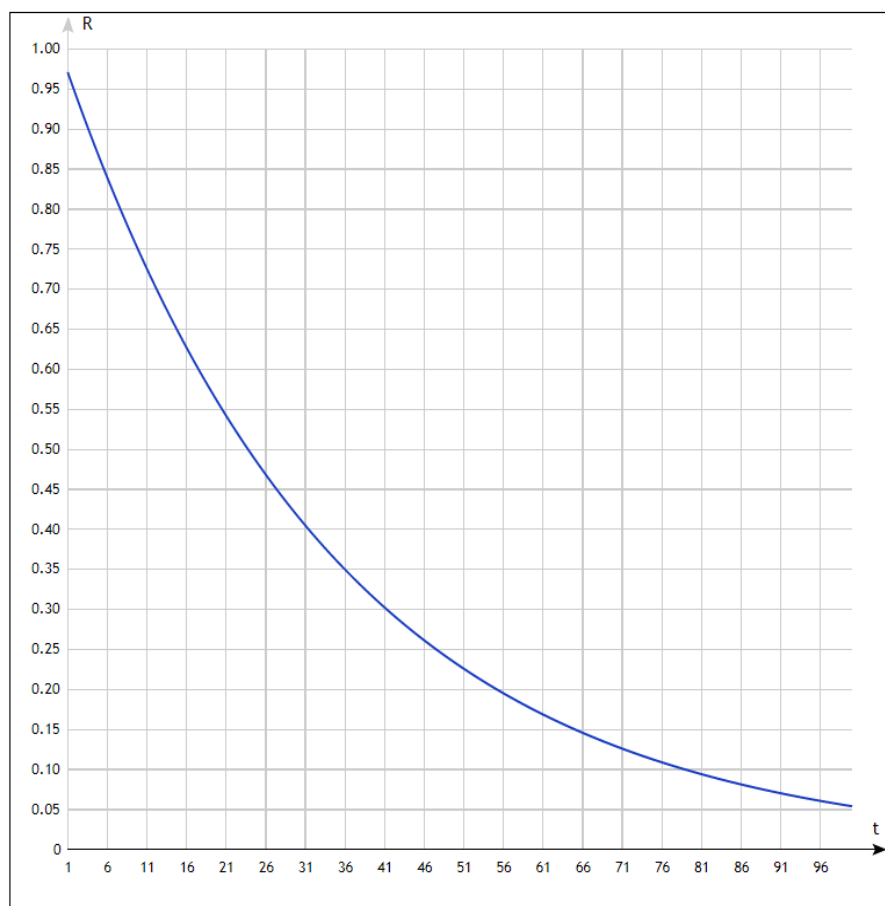


Рисунок 3.20 -График функции надёжности

Основываясь на материалах главы: выбор программных средств, описание реализованных функций интерфейса, настройка базы данных и, как итог, расчет надёжности - можно сделать вывод, что разработанное мобильное приложение «PollutionHelper», в полной мере обладает всем запланированным функционалом.

ЗАКЛЮЧЕНИЕ

Одним из шагов по решению проблемы управления мусорными потоками может выступить разработанное приложение «PollutionHelper», которое позволит разобраться с вопросом неосведомленности относительно мест раздельного сбора мусора.

За последний год ситуация с раздельным сбора мусора стала меняться в лучшую сторону, сейчас почти в каждом пункте по утилизации отходов появилось отведённое место для сбора пластика и люди используют его по назначению.

В результате выполнения работы, поставленные цели были выполнены в полном объеме:

1. проведён анализ предметной области;
2. рассмотрены аналоги проектируемой геоинформационной системы;
3. определены основные функции системы;
4. спроектирована информационная система «PollutionHelper» с учётом проведённого анализа;
5. реализовано мобильное приложение «PollutionHelper»;
6. рассчитана надёжность конечного продукта;
7. оформлена документация по проектируемой системе.

Получены практические навыки в области:

- построения диаграмм функционального проектирования;
- построения диаграмм методологии UML;
- сравнения аналогов проектируемой системы;
- работы с программой AndroidStudio;
- работы с базой данных Firebase;
- работы с языком программирования Java.

Итогом всей работы стало работоспособное мобильное приложение «PollutionHelper», предназначенное для создания и отображения на карте базы данных мест раздельного сбора мусора.

Предложения по дальнейшему улучшению приложения:

- улучшение системы поощрения – привлечение заинтересованных организаций, компаний готовых стимулировать пользователей данного рода системы;
- повышение финансовой выгоды владельцев системы – размещение направленной рекламы в приложении;
- работа без постоянного подключения к интернету – работа части или полного функционала приложения путём загрузки пакетов данных и выгрузки их при появлении соединения с сетью;
- сделать отображение маркеров на карте более информативным для пользователя приложения, раскрасить по цветам в зависимости от вида собираемого мусора;
- сделать кастомный (индивидуальный) стиль чата, для более приятного визуального эффекта, добавить отображения аватара пользователя;
- добавить подтверждение актуальности маркера, ставить пометку «точка не подтверждена» первые три дня после добавления или до снятия пометки;
- добавить элемент администрирования, интегрированный в приложение, например подтверждение работоспособности точки, изнутри системы.

Выводы по работе:

- на данный момент в России нет универсальных мест для сбора мусора, охватывающих все его виды. Разработанное приложение решает вопрос нехватки информации по пунктам сбора конкретного вида отходов. При помощи интерактивной карты, «PollutionHelper» доступно и наглядно ориентирует пользователя относительно мест сбора мусора по категориям;

- благодаря понятному интерфейсу «PollutionHelper», приложением могут пользоваться не только молодые люди, но и люди в возрасте;
- приложение разработано на операционной системе Android, активно используемой в России, поэтому оно доступно преобладающему числу граждан;
- с помощью накопленных в приложении данных, можно обобщить информацию и сделать выводы о необходимости размещения дополнительных точек сбора для определённого вида мусора;
- приложение «PollutionHelper» позволяет сформировать логистику сбора мусора для компаний, специализирующихся на раздельном сборе и переработке мусора.

СПИСОК ЛИТЕРАТУРЫ

1. Белов, В.В. Проектирование информационных систем: Учебник / В.В. Белов. - М.: Академия, 2018. - 144 с.
2. Гвоздева, Т.В. Проектирование информационных систем: технология автоматизированного проектирования. Лабораторный практикум. Учебно-справочное пособие / Т.В. Гвоздева, Б.А. Баллод. - СПб.: Лань, 2018. - 156 с.
3. Гвоздева, Т.В. Проектирование информационных систем. Стандартизация: Учебное пособие / Т.В. Гвоздева, Б.А. Баллод. - СПб.: Лань, 2019. - 252 с.
4. Гинзбург, В.М. Проектирование информационных систем в строительстве. Информационное обеспечение / В.М. Гинзбург. - М.: АСВ, 2008. - 368 с.
5. Емельянова, Н.З. Проектирование информационных систем: Учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. - М.: Форум, 2013. - 432 с.
6. Заботина, Н.Н. Проектирование информационных систем: Учебное пособие / Н.Н. Заботина. - М.: НИЦ Инфра-М, 2013. - 331 с.
7. Исаев, Г.Н. Проектирование информационных систем: Учебное пособие / Г.Н. Исаев. - М.: Омега-Л, 2013. - 424 с.
8. Исаев, Г.Н. Проектирование информационных систем: Учебное пособие / Г.Н. Исаев. - М.: Омега-Л, 2009. - 240 с.
9. Коваленко, В.В. Проектирование информационных систем: Учебное пособие / В.В. Коваленко. - М.: Форум, 2012. - 320 с.
10. Коваленко, В.В. Проектирование информационных систем: Учебное пособие / В.В. Коваленко. - М.: Форум, 2015. - 976 с.
11. Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQLWorkbench: Методы и средства проектирования информационных систем и технологий. Инструментальные

средства информационных систем: Учебное пособие / С.А. Мартишин, В.Л. Симонов,. - М.: ИД ФОРУМ, НИЦ Инфра-М, 2012. - 160 с.

12. Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQLWorkbench: Методы и средства проектирования информационных систем и технолог / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, 2017. - 62 с.

13. Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQLWorkbench: Методы и средства проектирования информационных систем и техноло / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, 2018. - 61 с.

14. Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова, О.П. Ляпина, А.В. Гусева. - М.: Academia, 2017. - 416 с.

15. Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова. - М.: Академия, 2018. - 272 с.

16. Соловьев, И.В. Проектирование информационных систем / И.В. Соловьев. - М.: Академический проспект , 2009. - 398 с.

17. Соловьев, И.В. Проектирование информационных систем. Фундаментальный курс / И.В. Соловьев, А.А. Майоров. - М.: Академический проект, 2009. - 398 с.

18. Соловьев, И.В. Проектирование информационных систем. Фундаментальный курс: Учебное пособие для высшей школы / И.В. Соловьев, А.А. Майоров; Под ред. В.П. Савиных. - М.: Академический проспект, 2009. - 398 с.

19. Федоров, Н.В. Проектирование информационных систем на основе современных CASE-технологий / Н.В. Федоров. - М.: МГИУ, 2008. - 280 с.

20. Монахов, Ю.М. Функциональная устойчивость информационных систем/ Ю.М. Монахов. – Владим. гос. ун-т, 2011. - 60 с.

21. Глобальные экологические проблемы современного мира и пути их решения. 2020. [Электронный ресурс] URI <http://vyvoz.org/blog/globalnye-jekologicheskie-problemy-sovremennogo-mira>.

22. Раздельный сбор отходов. 2020. [Электронный ресурс] URI <http://www.investmentrussia.ru/sortirovka-i-razdelnyj-sbor-otxodov>.

23. Система управления отходами. 2014. [Электронный ресурс] URI http://studopedia.ru/1_86956_sistema-upravleniya-othodami.html.

24. Раздельный сбор мусора. 2021. [Электронный ресурс] URI <http://nemusorim.com/musor/razdelniy-sbor>.

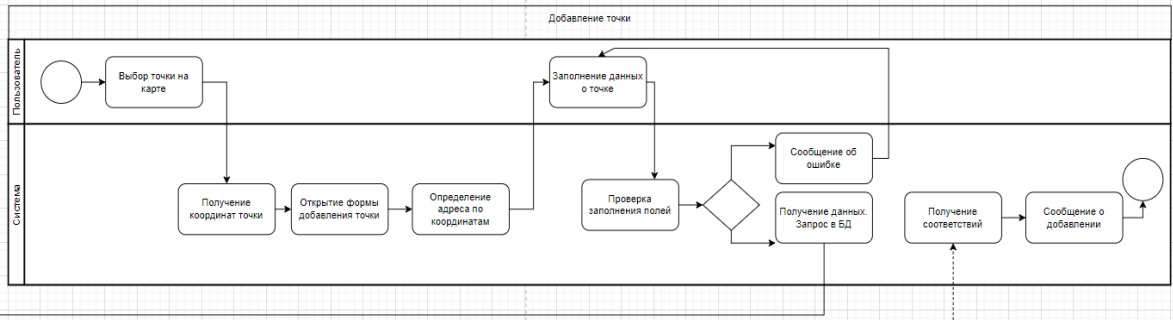


Рисунок А.2 BPMN Добавление точки

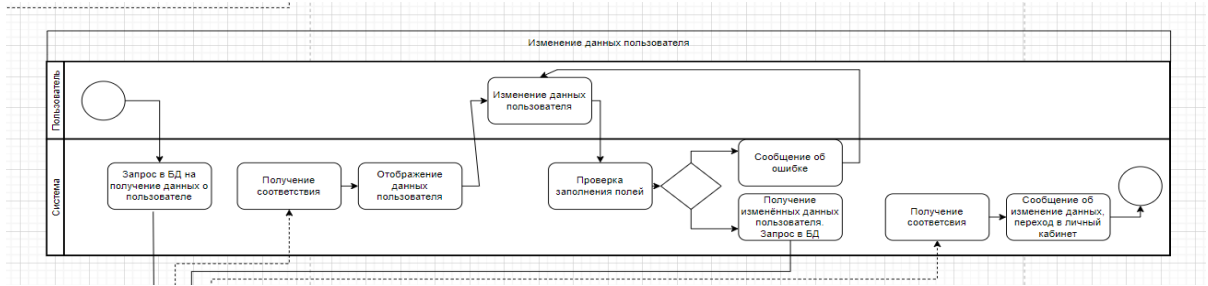


Рисунок А.3 BPMN Изменение данных пользователя

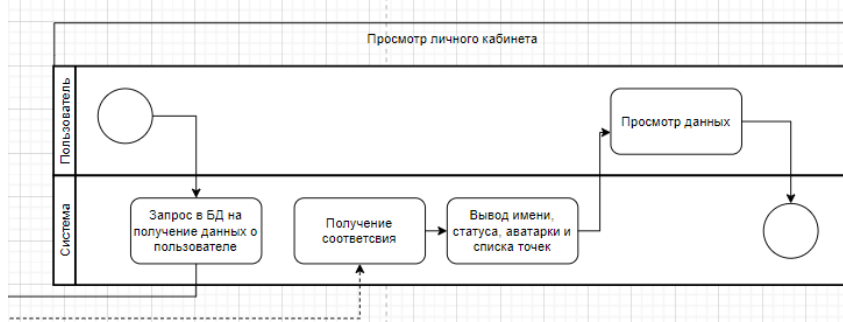


Рисунок А.4 BPMN Просмотр личного кабинета

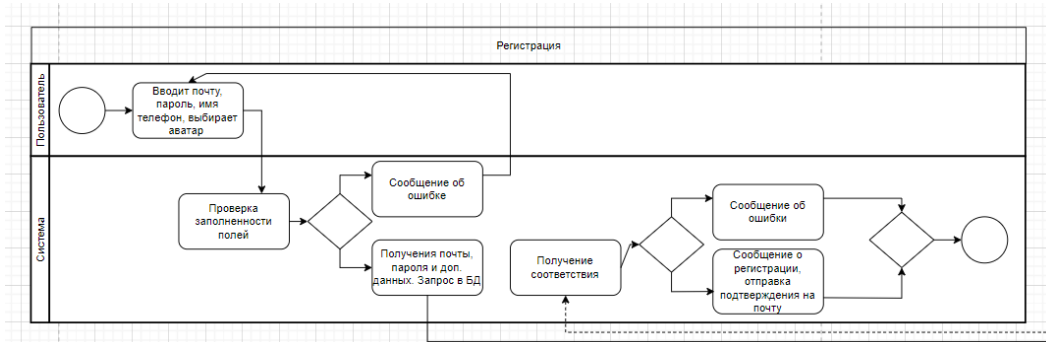


Рисунок А.5 BPMN Регистрация

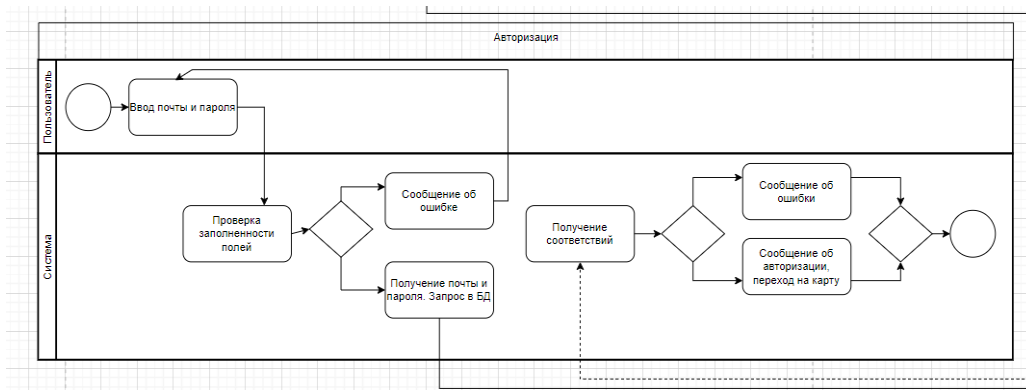


Рисунок А.6 BPMN Авторизация

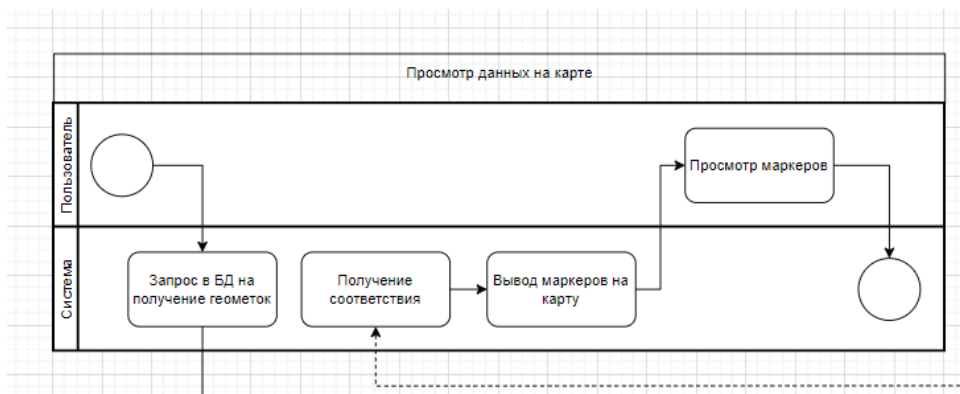


Рисунок А.7 BPMN Просмотр данных на карте

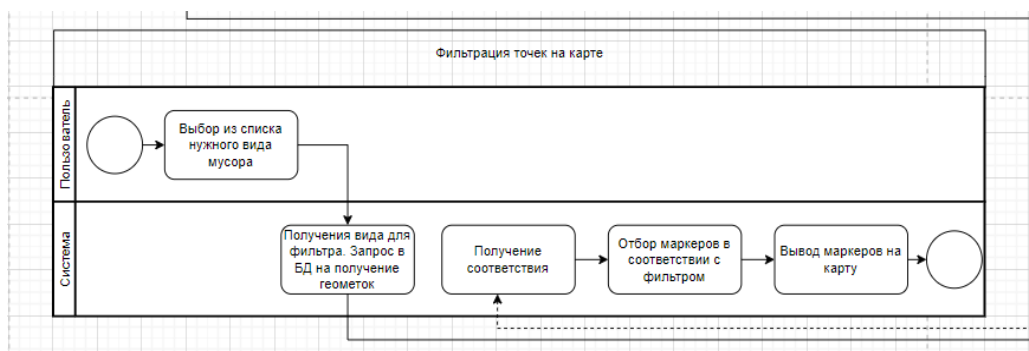


Рисунок А.8 BPMN Фильтрация точек на карте

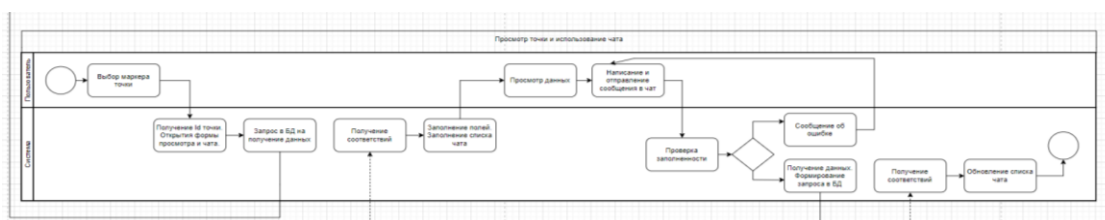


Рисунок А.9 BPMN Просмотр точки и работа чата

ПРИЛОЖЕНИЕ Б

Диаграмма последовательностей

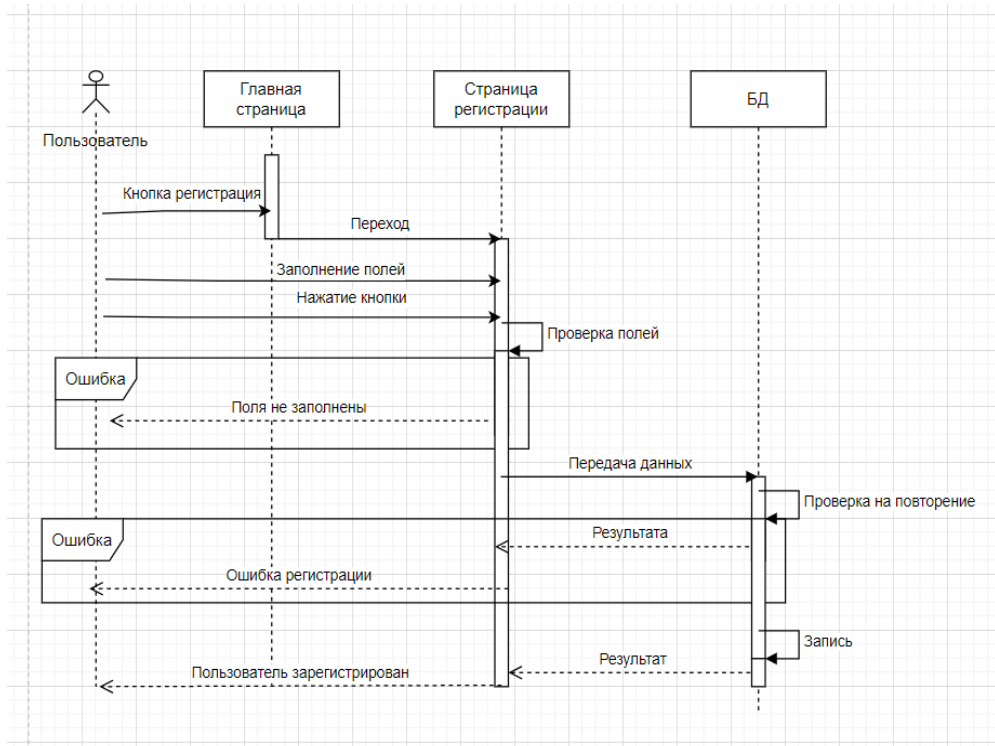


Рисунок Б.1 Диаграмма последовательностей, регистрация

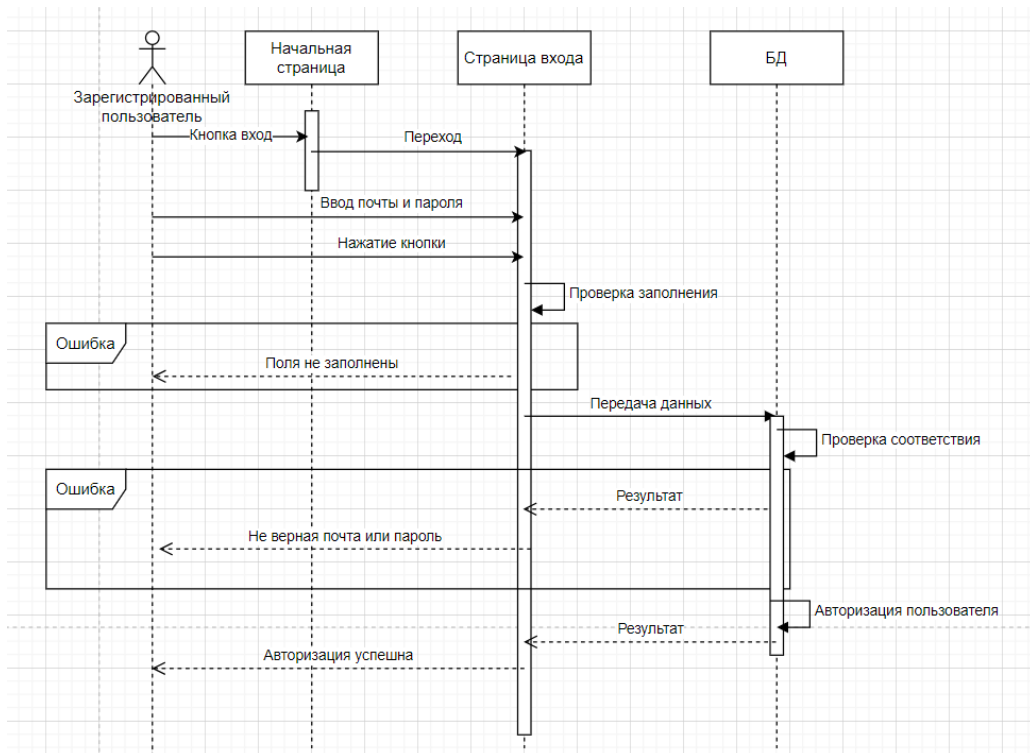


Рисунок Б.2 Диаграмма последовательностей, вход

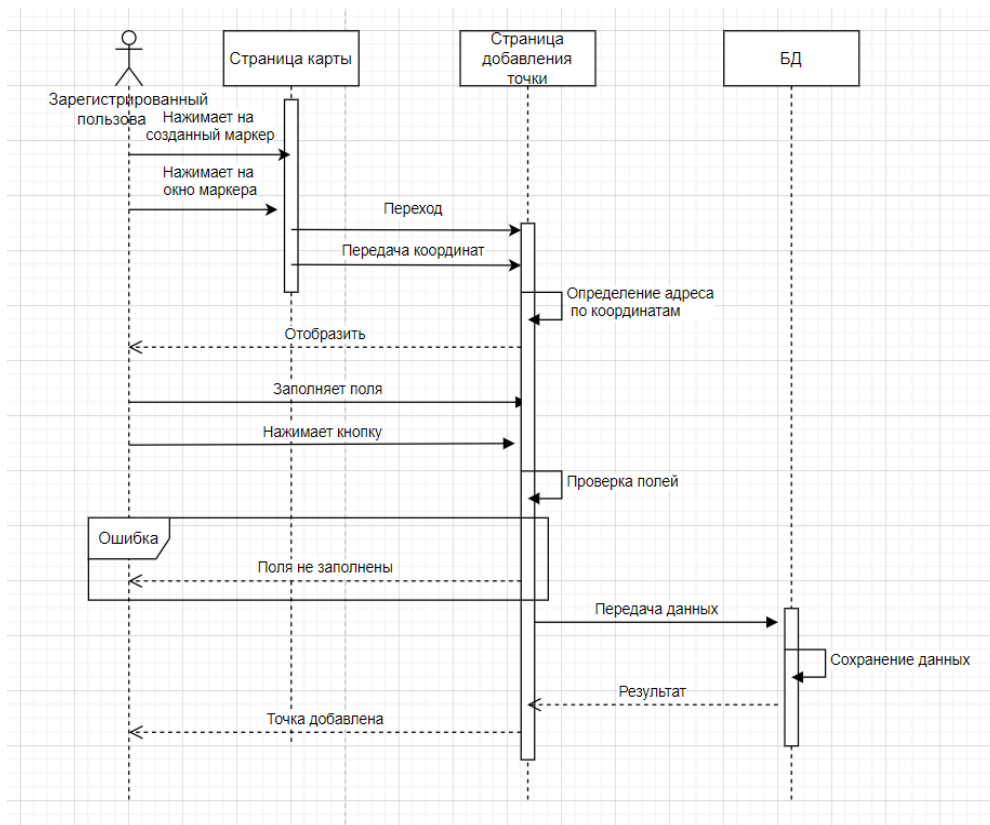


Рисунок Б.3 Диаграмма последовательностей, добавление точки

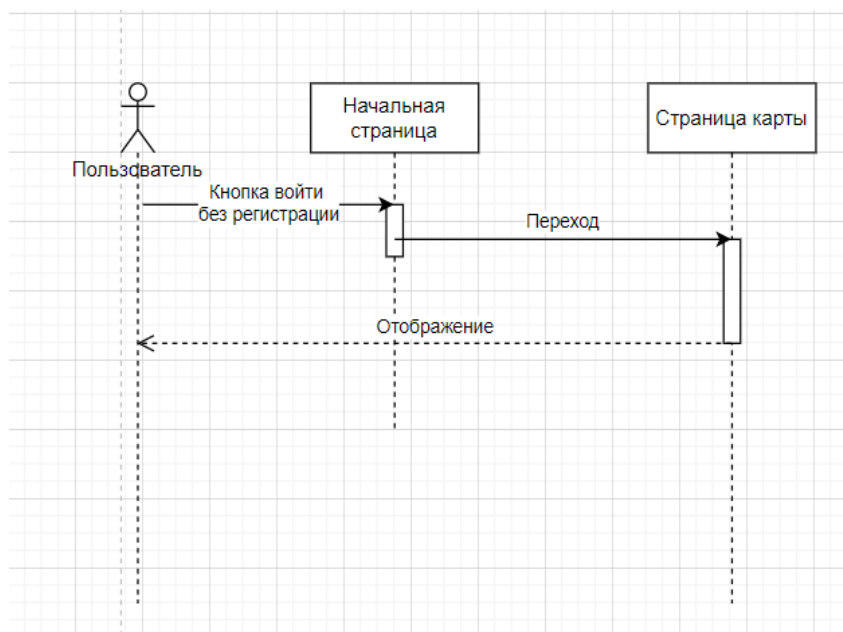


Рисунок Б.4 Диаграмма последовательностей, вход без регистрации

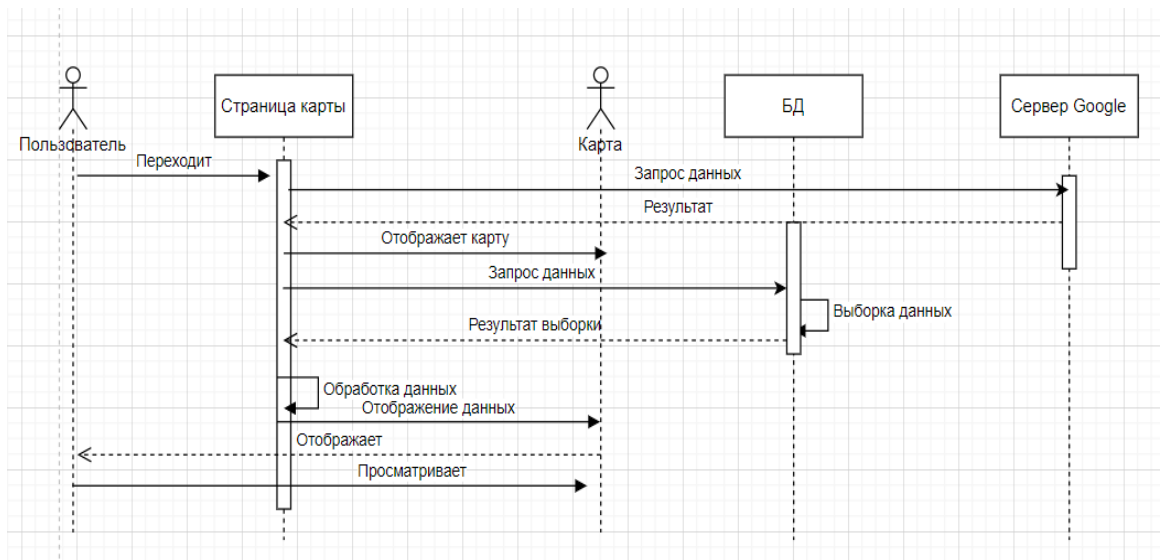


Рисунок Б.5 Диаграмма последовательностей, просмотр карты

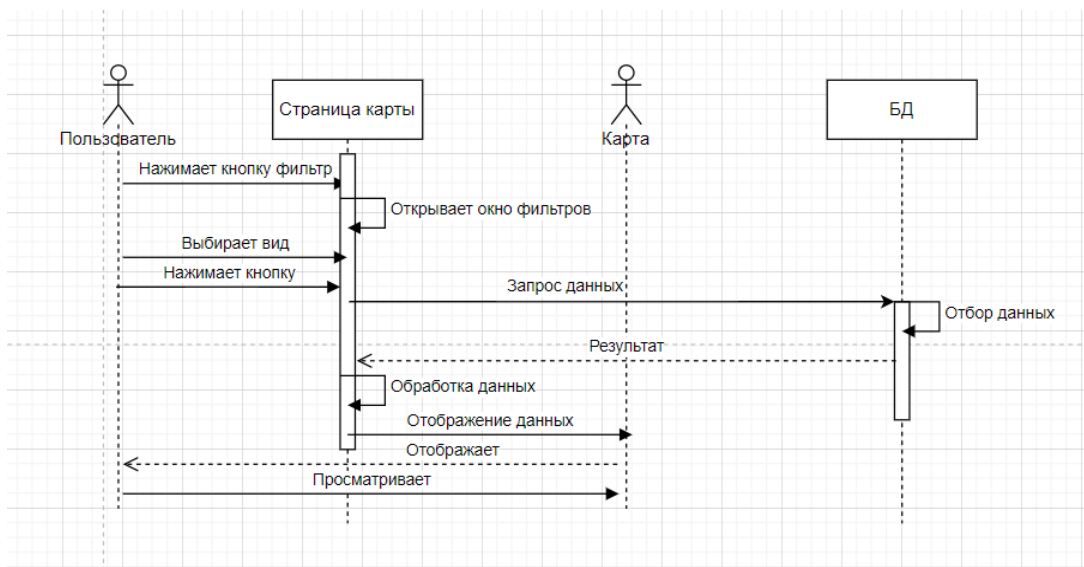


Рисунок Б.6 Диаграмма последовательностей, фильтр карты

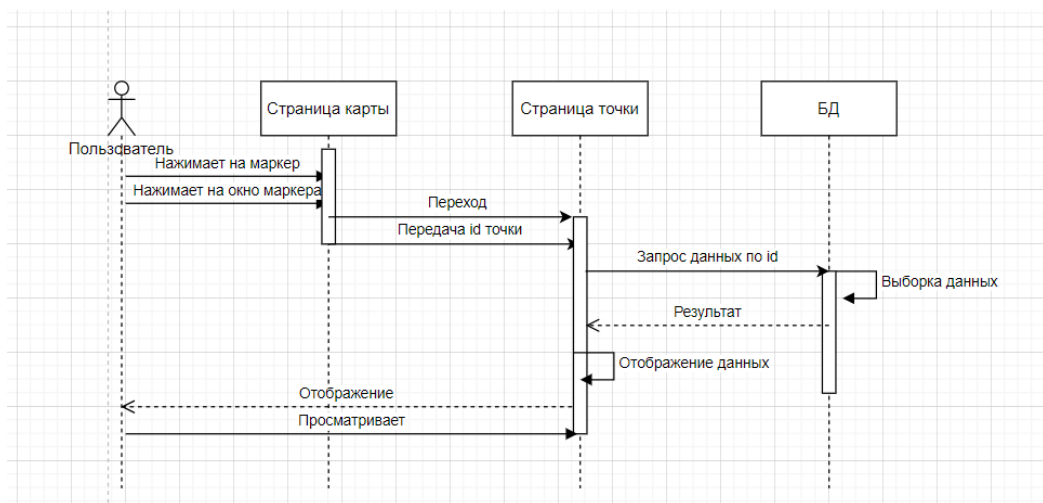


Рисунок Б.7 Диаграмма последовательностей, просмотр точки

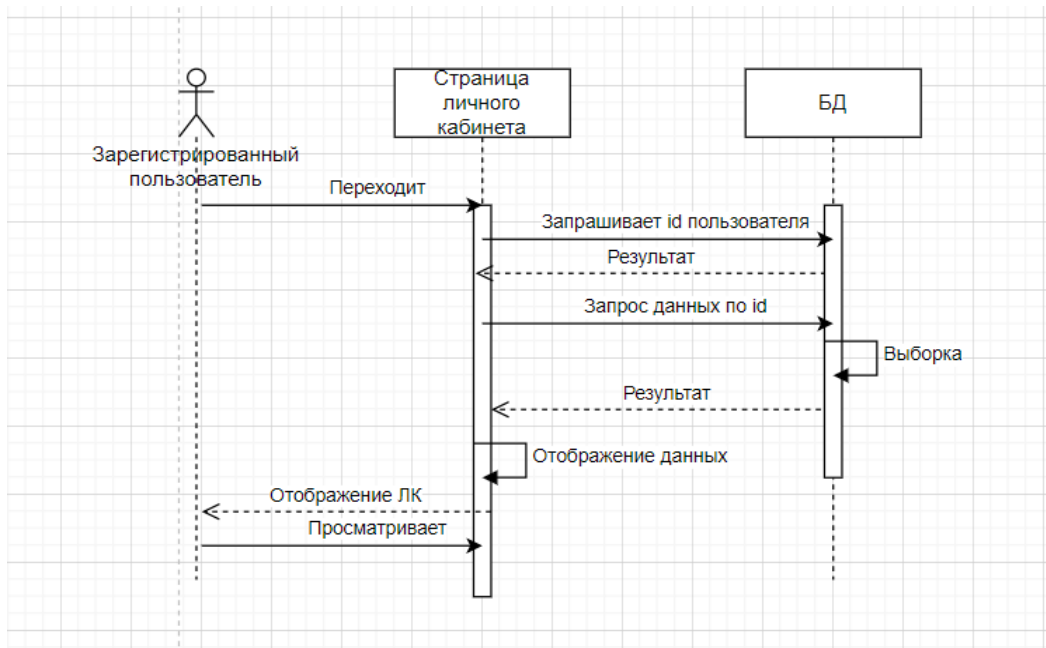


Рисунок Б.8 Диаграмма последовательностей, просмотр личного кабинета

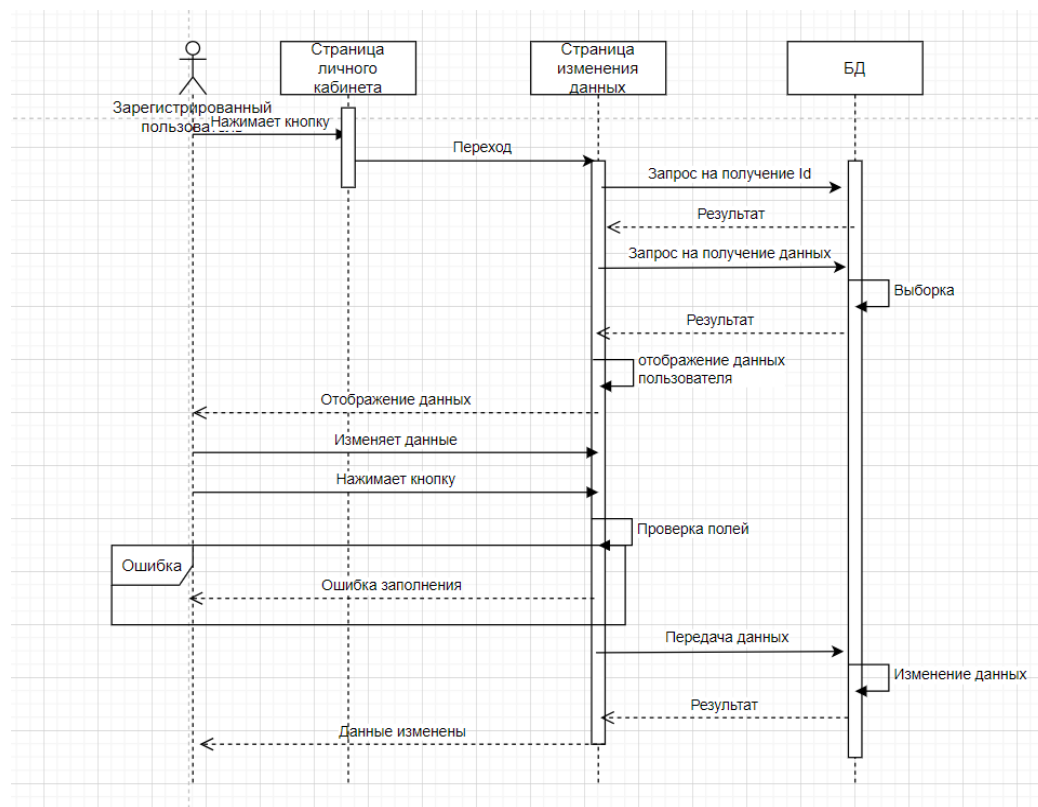


Рисунок Б.9 Диаграмма последовательностей, изменение данных

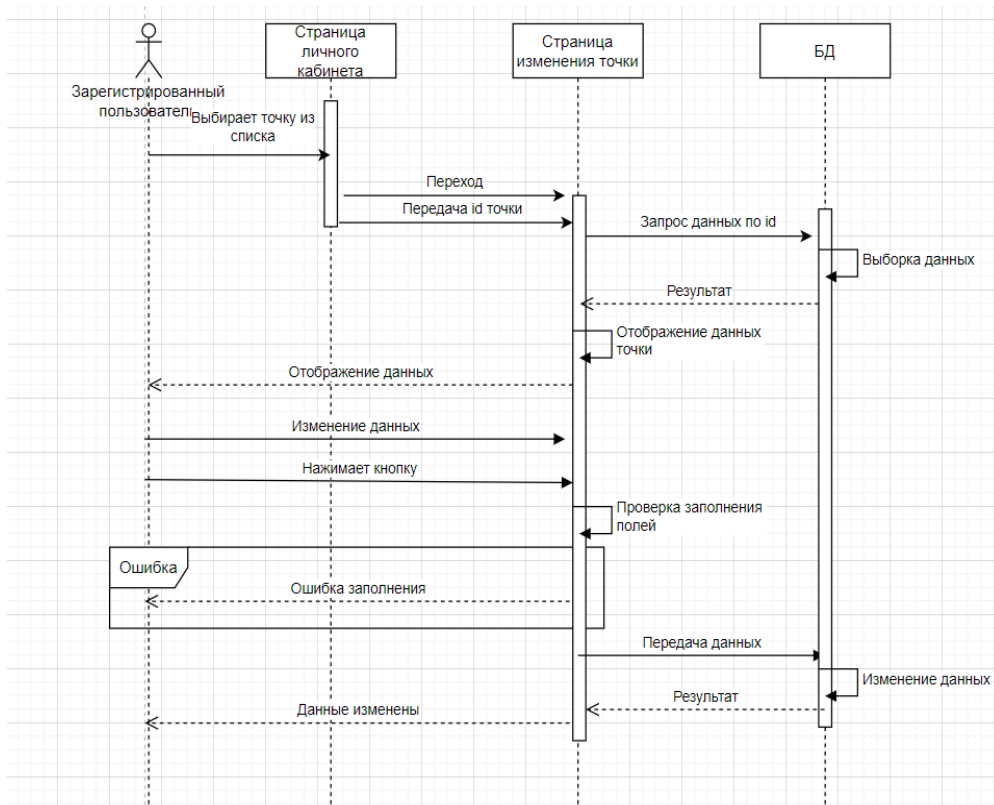


Рисунок Б.10 Диаграмма последовательностей, изменение точки

ПРИЛОЖЕНИЕ В

Диаграмма классов

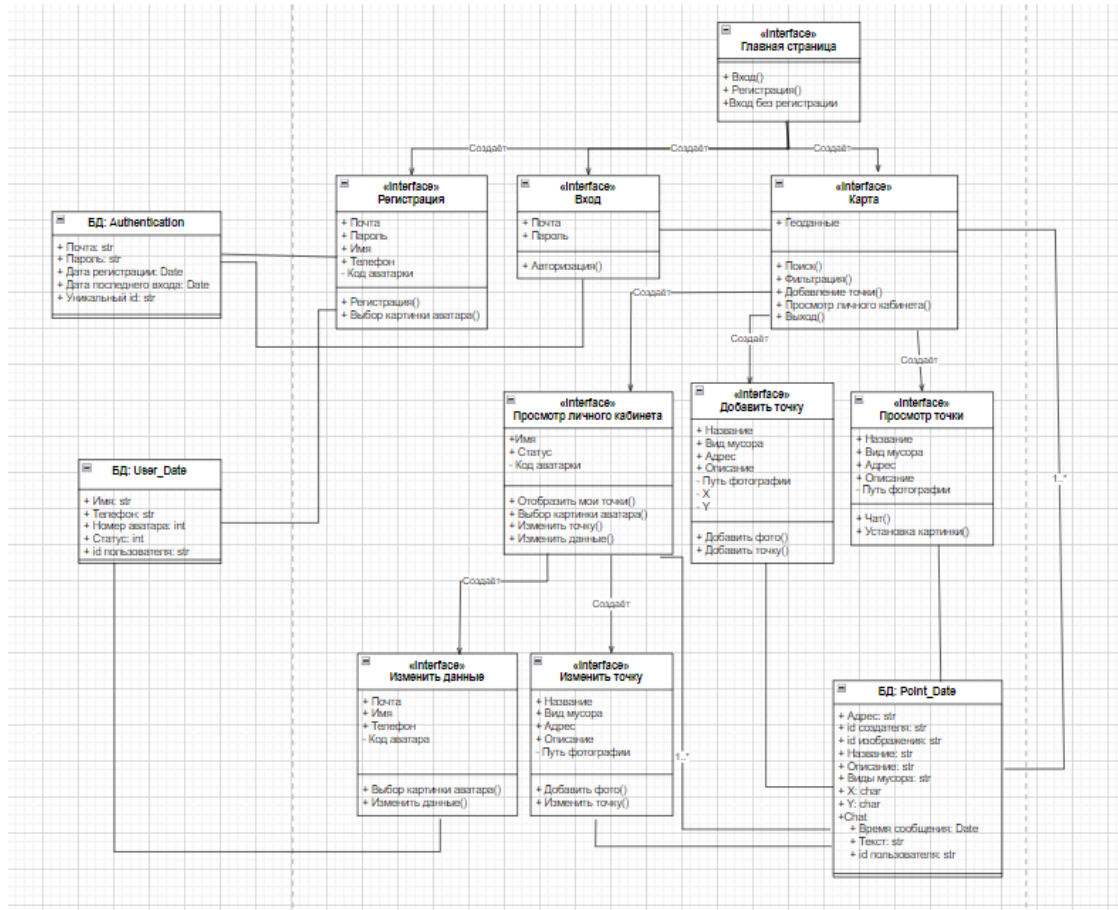


Рисунок В.1 Диаграмма классов