

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ
Базовая кафедра «Аналитика больших данных и методы видеоанализа»

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

Зав. кафедрой Гайнанов Д.Н.

«07» июня 2021 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)**

«Разработка цифровой платформы «Активный Екатеринбург»»

Научный руководитель: Медведев Максим Александрович
доцент, к.э.н.

Нормоконтролер: Медведева Марина Александровна
доцент, к.ф.-м.н., доцент

Студент группы РИМ-291217 Жульмухаметов Мурат Куанышевич

Екатеринбург
2021

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ
Базовая кафедра «Аналитика больших данных и методы видеоанализа»
Направление подготовки 09.04.03 Прикладная информатика
Образовательная программа ИТ инновации в бизнесе

УТВЕРЖДАЮ

Зав. кафедрой Гайнанов Д.Н.

«25» февраля 2021 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студента Жульмухаметова Мурата Куанышевича группы РИМ-291217
(фамилия, имя, отчество)

1. Тема ВКР Разработка цифровой платформы «Активный Екатеринбург»
Утверждена распоряжением по институту № 33.02-05/209 от 21.12.2020

2. Руководитель Медведев Максим Александрович, доцент, кандидат экономических наук
(Ф.И.О., должность, ученая степень, ученое звание)

3. Исходные данные к работе техническая документация, электронные ресурсы, нормативные документы предприятия, литературные источники

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов)

Введение

Глава 1. Анализ современных методов и систем «Умный город»

Глава 2. Архитектура и бизнес-процессы разрабатываемой системы

Глава 3. Разработка цифровой платформы вовлечения граждан в решения вопросов городского развития

Заключение

Список использованных источников

5. Перечень демонстрационных материалов Презентация, выполненная в MS PowerPoint

6. Консультанты по проекту (работе) с указанием относящихся к ним разделов*

Раздел	Консультант	Подпись, дата	
		задание выдал	задание принял

7. Календарный план

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
11	Введение, заключение	27.04.2021-16.05.2021	
22	1 глава	27.04.2021-16.05.2021	
33	2 глава	27.04.2021-16.05.2021	
44	3 глава	27.04.2021-16.05.2021	
55	Нормоконтроль	до 21.05.2021	
66	ВКР в целом	до 01.06.2021	

Руководитель _____ Медведев М.А.
(подпись) (Ф.И.О.)

Задание принял к исполнению _____
(подпись)

8. Выпускная квалификационная работа закончена 1 июня 2021 г.

Пояснительная записка и все материалы просмотрены

Оценка консультантов: * а) _____ б) _____
в) _____ г) _____

Считаю возможным допустить Жульмухаметова Мурата Куанышевича к защите его выпускной квалификационной работы в экзаменационной комиссии.

Руководитель _____
(подпись)

9. Допустить Жульмухаметова Мурата Куанышевича к защите выпускной квалификационной работы в экзаменационной комиссии (протокол заседания кафедры №5 от «07» июня 2021 г.).

Зав. кафедрой _____ Гайнанов Д.Н.
(подпись) (Ф.И.О.)

* - при наличии разделов, требующих привлечения консультантов

РЕФЕРАТ

Тема магистерской диссертации: Разработка цифровой платформы «Активный Екатеринбург».

Магистерская диссертация выполнена на 99 страниц, содержит 5 таблиц, 38 рисунков, 61 использованный источник.

Магистерская диссертация является актуальной, поскольку тема платформ вовлечения граждан в решения вопросов городского развития является не глубоко изученной на территории РФ.

Целью магистерской диссертации является разработка цифровой платформы для решения вопросов городского развития. Для этого были определены следующие задачи:

- анализ схожих по функционалу систем на территории РФ;
- разработка архитектуры и бизнес-процессов будущей системы;
- разработка системы.

Объектом разработки выпускной квалификационной работы является цифровая платформа вовлечения граждан в решения вопросов городского развития – «Активный Екатеринбург».

Предметом исследования являются основные принципы разработки информационных систем с сервис-ориентированной архитектурой.

Научная новизна исследования обусловлена его объектом и предметом и заключается в том, что впервые:

1. Проведено изучение концепции «Умный город» в разрезе западных и отечественных проектов, что дало понимание проблемы разработки подобных проектов в России.
2. Проработаны основные положения построения архитектуры информационной системы взаимодействия горожан с местной властью.
3. Разработана уникальная информационная система, покрывающая решения проблем городского хозяйства граждан Екатеринбурга.

Практическая значимость заключается в повышении эффективности решения проблем городского хозяйства.

Результатом выполненной работы является информационная система, которая принимает заявления от граждан и в дальнейшем обрабатывает их.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	8
1 Обзор и анализ современных методов и систем «Умный город».....	12
1.1 Концепция «Умный город». Определение и особенности разработки.....	12
1.2 Исследование западных проектов «Умный город».....	13
1.3 Исследование отечественных проектов «Умный город»	19
1.4 Результаты и выводы.....	26
2 Архитектура и бизнес-процессы	27
2.1 Закрытая часть Цифровой платформы «Активный Екатеринбург»	27
2.2 Открытая часть Цифровой платформы «Активный Екатеринбург»	42
2.3 Мобильное приложение Цифровое платформы «Активный Екатеринбург»	48
2.4 Система авторизации пользователей Цифровой платформы «Активный Екатеринбург».....	49
2.5 Результаты и выводы.....	54
3 Разработка цифровой платформы «Активный екатеринбург»...	55
3.1 Применяемые технологии.....	55
3.2 Описание API	65
3.3 Результаты и выводы.....	90
ЗАКЛЮЧЕНИЕ	91
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	94

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В пояснительной записке применяют следующие термины с соответствующими определениями:

Классификатор	— систематизированный перечень наименованных объектов
Мобильное приложение	— программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы
Фреймворк	— заготовки, шаблоны для программной платформы, определяющие архитектуру программной системы.
Цифровая платформа, Платформа	— информационное пространство, объединяющее жителей города и органы власти, с целью эффективного взаимодействия по вопросам городского развития
API	— (англ. application programming interface) программный интерфейс приложения, интерфейс прикладного программирования. Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой

JSON	— текстовый формат обмена данными, основанный на языке программирования JavaScript
Keycloak	— технология с открытым исходным кодом, предназначенная для идентификации и контроля доступа
SSO	— технология единого входа. Технология, при использовании которой пользователь переходит из одного раздела портала в другой, либо из одной системы в другую, не связанную с первой системой, без повторной аутентификации
React Native	— фреймворк для разработки кроссплатформенных приложений для iOS и Android
Web-интерфейс	— веб-страница или совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В пояснительной записке применяют следующие сокращения и обозначения:

API	–	Application programming interface
АИС	–	Автоматизированная информационная система
ЕСИА	–	Единая система идентификации и аутентификации
МО	–	Муниципальное образование
ОМСУ	–	Орган местного самоуправления
ПО	–	Программное обеспечение
СНИЛС	–	Страховой номер индивидуального лицевого счёта
URL	–	Uniform Resource Locator
БД	–	База данных
ИС	–	Информационная система
МУ	–	Медицинские учреждения
ОС	–	Операционная система
СУБД	–	Система управления базами данных

ВВЕДЕНИЕ

В настоящее время идет процесс урбанизации, так как именно в городах содержатся основные части инвестиций государства и создаются более комфортные условия для проживания. Возникает необходимость внедрения инноваций, повышения эффективности функционирования города, а также улучшения качества жизни жителей.

В городах РФ увеличивается население, и как следствие увеличивается количество проблем, связанных с городским хозяйством, и как следствие сроки решений данных проблем растягиваются и большой процент проблем остается нерешенным.

Основная проблема решения данных задач лежит во взаимодействии граждан с представителями власти. На сегодняшний момент для предупреждения вышестоящих органов о имеющейся проблеме городской среды необходимо знать адреса и зоны ответственности тех или иных органов местного самоуправления. Далее заявить о существующей проблеме в письменной форме и ждать решения без каких-либо официальных ответов.

Таким образом, приказом Министерства строительства и жилищно-коммунального хозяйства Российской Федерации от 31.10.2018 г. № 695 был утвержден ведомственный проект цифровизации городского хозяйства «Умный город» в рамках которого в соответствии с утверждёнными требованиям должна быть внедрена цифровая платформа вовлечения граждан в решения вопросов городского развития – «Активный Екатеринбург».

Целями создания Цифровой платформы являются:

1. Обеспечение исполнения требований нормативно правовых актов Российской Федерации, регламентирующих порядок вовлечения граждан в решения вопросов городского развития.
2. Обеспечение участия граждан в решении вопросов городского развития, а также актуальных для них проблем путём создания оперативного канала прямого взаимодействия жителей, проживающих на территории

муниципального образования «город Екатеринбург», с органами местного самоуправления муниципального образования «город Екатеринбург».

3. Создание единой системы учета заявлений, формируемых гражданами по вопросам формирования комфортной городской среды.
4. Автоматизация деятельности уполномоченного органа по приёму, контролю сроков ответов и ответов на заявления, поступающие через мобильное приложение или web-интерфейс (сайт) цифровой платформы.
5. Реализация информационного взаимодействия между уполномоченным органом и другими участниками процедуры вовлечения граждан в решения вопросов городского развития.
6. Автоматизация деятельности уполномоченного органа по формированию опросов и голосований, их проведению и подведению итогов.

Перечень задач, решаемых с использованием Цифровой платформы:

1. Выполнение функций уполномоченного органа по приёму, обработке, контролю сроков ответов и ответов на заявления, поступающие через мобильное приложение или web-интерфейс (сайт) цифровой платформы.
2. Выполнение функций уполномоченного органа по формированию опросов и голосований, их проведению и подведению итогов.
3. Взаимодействие с органами, ответственными за исполнения заявлений в соответствии с Классификатором заявлений (Департаментами, Управлениями, Комитетами, Отделами Администрации города Екатеринбург и Администрациями районов города Екатеринбурга).
4. Взаимодействие с гражданами, проживающими на территории муниципального образования «город Екатеринбург» по вопросам городского развития.
5. Автоматизация процесса подсчета и предоставления результатов опросов и голосований в порядке, установленном муниципальными нормативно правовыми актами.
6. Аналитическая и статистическая работа по каждому голосованию и опросу.

7. Аналитическая и статистическая работа по поступившим от граждан и обработанным заявлениям.

Объектом автоматизации является процесс взаимодействия жителей муниципального образования «город Екатеринбург» с органами местного самоуправления в части размещения заявлений в адрес органов местного самоуправления, публикации заявлений граждан, организации деятельности органов местного самоуправления (далее – ОМСУ) по обработке, исполнению и ответов на заявления, контроля сроков и эффективности исполнения заявлений, а также организации проведения публичных опросов и голосований.

Объект автоматизации включает в себя:

1. Деятельность по формированию и отправке заявлений гражданами в адрес ОМСУ.
2. Деятельность по приёму, контролю сроков ответов и ответов на заявления граждан.
3. Деятельность по взаимодействию исполнителя и гражданина в ходе исполнения заявления с целью уточнения информации по заявлению.
4. Деятельность по формированию Классификатора заявлений.
5. Деятельность по формированию опросов и голосований, их проведению и подведению итогов.
6. Деятельность по подготовке аналитических и статистических данных по каждому голосованию и опросу, а также по заявлениям.

Уникальность исследования заключается в новом способе взаимодействия администрации с гражданами Екатеринбурга, а также в пути решения пробел городского хозяйства. В итоге была разработана уникальная система, решающая данные задачи.

Практическая значимость заключается в повышении эффективности решения проблем городского хозяйства.

Теоретической базой являются научные труды ученых и научных центров в области концепции «умный город», разработки частных компаний и Минстроя РФ.

Информационной базой исследования послужили данные Федеральной службы государственной статистики, данные территориального органа Федеральной службы, статьи, электронные публикации НИИТС, «Банк решений умного города», разработанный анализ Минстроя РФ

При выполнении магистерской диссертации использовался офисный пакет приложений Microsoft Office.

Выпускная квалификационная работа содержит 99 страниц текстового документа, 38 иллюстраций, 5 таблиц, 61 использованных источника.

1 ОБЗОР И АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ И СИСТЕМ «УМНЫЙ ГОРОД»

1.1 Концепция «Умный город». Определение и особенности разработки

Существуют различные толкования определения «Умный город»:

«Умный город» — концепция интеграции нескольких информационных и коммуникационных технологий (ИКТ) и Интернета вещей (IoT решения) для управления городским имуществом; активы города включают, в частности, местные отделы информационных систем, школы, библиотеки, транспорт, больницы, электростанции, системы водоснабжения и управления отходами, правоохранительные органы и другие общественные службы. Целью создания «умного города» является улучшение качества жизни с помощью технологии городской информатики для повышения эффективности обслуживания и удовлетворения нужд резидентов [1].

Умный город (с английского Smart City) – это взаимосвязанная система коммуникативных и информационных технологий с интернетом вещей (IT), благодаря которой упрощается управление внутренними процессами города и улучшается уровень жизни населения [2].

«Умный» город — эффективная интеграция физических, цифровых и человеческих систем в искусственной среде ради устойчивого и благополучного будущего жителей, симбиоз гражданской сознательности и технологических инноваций [3].

Из совокупности определений можно прийти к выводу о том, что «Умный» город — это город с развитыми информационными системами управления городским имуществом для комфортного и безопасного проживания граждан с минимальными затратами природных и человеческих ресурсов.

В зависимости от поставленных целей «Умный» город можно обозначить как «экогород» или «цифровой город». Такие системы должны постоянно контролировать важнейшие инфраструктурные объекты городов, таких как: автомобильные и железные дороги, мосты, аэропорты, системы связи и т. д. для эффективного распределения ресурсов и обеспечение безопасности населения.

Основное отличие «умного» города от обычного заключается в качестве взаимодействий с гражданами. В обычном городе информационные услуги и сервисы не могут так же эффективно и гибко реагировать на изменения экономических, социальных и культурных условий города. В таком случае «умный» город прежде всего направлен на самого человека, основывается на системе инфраструктуры ИКТ и непрерывном развитии города при непрерывном учете требований экологической и экономической устойчивости.

В западных государствах уже давно внедряется концепция «Умного» города, которая становится основой стратегического развития, объединив в себе современные коммуникации и технологии природного и социального капитала.

1.2 Исследование западных проектов «Умный город»

Рассмотрим некоторые зарубежные реализации проектов концепции «Умный город».

Первый пилотный проект «умного» города появился на юго-западе Токио в районе Kansai, в котором проходило испытание новых коммуникационных технологий и программ. Для реализации проекта скооперировались правительство и частные компании. Результатом стала успешная интеграция высокотехнологичных услуг в повседневную жизнь участников проекта, а также расширение возможностей для развития бизнеса.

В 1993 году группа компаний в Силиконовой долине решила создать полноценную местную инфраструктуру. В эту группу входили коммуникационные компании, исследовательские институты, инвестиционные банки и местные органы власти. Система получила название «умный Вале́й» (Valley). «Умный Вале́й» создавался в коммерческих целях и финансировался исключительно из частных источников. Компании-участники отмечали обширные экономические и социальные выгоды от развития коммуникационных технологий в городской среде [4].

В идее обеспечения конкурентоспособности и развития городов с помощью информационно-коммуникационных технологий были заинтересован ряд крупных компаний, которые стали развивать концепцию «умного» города в рамках своей специализации. Ярким примером стала компания Cisco, которая в конце 90-х годов разработала и внедрила концепт «умного» города в Милане, используя механизмы государственно-частного партнерства. А компания IBM реализовывала комплексные проекты «умный город» во многих крупных городах мира, таких как Нью-Йорк, Чикаго, Мадрид, для которых применяла различные схемы партнерств с бизнесом и с властями разных уровней.

«Умные города» в мире широко распространились, и к 2020 году их количество достигло 143 проекта разной степени завершенности: в Северной Америке – 35 проектов, в Южной Америке – 11, в Европе – 47, в Азии – 40, 10 на Среднем Востоке и в Африке [5]. Проекты условно подразделяются на строительные проекты новых городов и проекты трансформации существующих.

Бюджеты проектов «умных городов» за рубежом представлены в таблице 1.

Таблица 1 – Бюджеты проектов строительства «умных городов» за рубежом
[составлено автором]

Город	Страна	Бюджет проекта, млрд. долл.	Год завершения проекта	Основные отличия концепции «Умного города»
Songdo International Business District	Южная Корея	40	2015	Построен с «нуля». Компьютеры встроены в здания и улицы
Джазан (JEC)	Саудовская Аравия	30	2016	Обычные населенные пункты переходят в «подключенные сообщества»
Искандар	Малайзия	30	2025	Главной частью города станет «промышленная долина», рассчитанная на 3000 предприятий и 330 000 рабочих.
PlanIT Valley	Португалия	29	2015	Первый город на Земле, который был разработан с собственным программным обеспечением.
КАЕС	Саудовская Аравия	26	-	Главной частью города станет «промышленная долина»
Лаваса	Индия	20	2020	Полностью частный город
Масдар Masdar City	ОАЭ	16	2018	Полностью независимый, автономный город с возобновляющими источниками энергии

1.2.1 Songdo International Business District

Международный деловой район Сондо (Songdo IBD) – это «умный» город, реализованный с нуля на 600 га на рекультивированных землях вдоль набережной Инчхон, расположенной в 30 километрах к юго-западу от Сеула.

Сондо был построен, чтобы быть «вездесущим городом». Под «вездесущим» подразумеваются датчики и компьютеры, встроенные в здания и улицы города. Они могут контролировать освещение, отопление, кондиционирование воздуха и много другое, все с нажатием кнопки на панели управления. Жители Сондо могут проводить видеоконференции с соседями или даже посещать занятия удаленно.

Датчики собирают информацию о транспортном потоке и потреблении энергии. Данную информацию преобразуют и могут оповещать граждан, например, когда прибудет автобус или, информируют властей, когда происходит преступление.

1.2.2 Джазан (JEC)

Экономический город Джазан (Jazan Economic City) будет расположен у побережья Красного моря в 60 километрах от города Джизан. Город может стать одним из важнейших центров побережья Красного моря. Предполагается, что Джазан усилит торговые отношения между Азией и Африкой. Две трети территории JEC буду выделены под промышленные предприятия. Остальное пространство займет жилая застройка. В них будут поддерживаться различные типы образа жизни для сотрудников предприятий. Предположительно в «экогороде» Джазан будут проживать 300 человек.

1.2.3 Искандар

Главной частью города Искандар станет «промышленная долина» с 3000 предприятий с 330 000 рабочих. Основными направлениями города станет производство потребительских товаров, машин, а также обслуживание

транспорта, строительство зданий и фармацевтика. На площади 13.8 км² будет реализован проект по самому крупному порту в регионе. На данный момент идет строительство инфраструктуры.

1.2.4 PlanIT Valley

PlanIT Valley – первый город в мире, который был спроектирован с собственным программным обеспечением. Первым зданием, которое будет готово к сдаче в эксплуатацию, станет центр обработки данных. Центр является детищем компании Living PlanIT. Первыми жителями города станут сотрудники и партнеры компании, которые и испытают на себе новейшую разработку.

Планируется, что первые 150 тысяч жителей будут сотрудниками и партнерами компании, и в то же самое время они станут теми, кому придется испытать на себе новые технологии.

Здания, датчики и сервисы будут соединены через облачное пространство, а устаревшие, вышедшие из строя здания будут отключаться сервером. Португальский PlanIT Valley является прототипом «умных» городов, которые компания надеется соорудить также и в Китае и Индии.

1.2.5 King Adbullah Economic City (KAEC)

КАЕС – один из четырех новых городов Саудовской Аравии. Город предназначен для создания дополнительного 1 миллиона рабочих мест, а также может стать приютом для 10 миллионов жителей, половина из которых необразованные молодые люди в возрасте до 17 лет.

КАЕС был задуман как Силиконовая долина Саудовской Аравии. В «умном» городе будет отдел регистрации и обслуживания правительственных услуг в экономических городах. Разработчики обещают, что сфера услуг будет доступна 24 часа в сутки, семь дней в неделю, каждую минуту.

1.2.6 Лаваса (Lavasa)

Лаваса – частный, спланированный город, построенный недалеко от Пуны. Если все пойдет по задуманному Строительной компанией Индостана (Hindustan Construction Company) плану, то в будущем году у индийского города Лаваса показатели IPO (первоначального публичного предложения) достигнут суммы в 20 миллиардов рупий или 437 миллионов долларов. Расположившись в трех часах езды от Мумбаи, самый известный «умный» город Индии разработан таким образом, что дома расположены на склонах гор – подальше от изнуряющей летней жары.

Из первой из пяти деревень планируется построить 50 тысяч домов на продажу и еще 25 тысяч для сдачи в аренду. «Умные» технологии в Лавасе предоставлены компаниями Cisco и Wirpo.

1.2.7 Масдар (Masdar City)

Масдар – проект будущего экогорода, расположенного в эмирате Абу-Даби Объединённых Арабских Эмиратов, строится в 17 километрах к юго-востоку от столицы страны, вблизи Международного аэропорта Абу-Даби.

Масдар будет сочетать традиционную арабскую архитектуру и информационные технологии. В планах осуществить запуск электрических машин без водителя.

Новый город Масдар по мнению экспертов слишком закрытый и эксклюзивный [6].

Варианты развития концепции «Умный город» в европейской и азиатской части отличаются: в азиатских странах преобладает технологическая направленность, а в европейских – социальная направленность.

Опыт внедрения проектов «Умный город» показывает, что при больших финансовых затратах на реализацию концепции «Умный город» в реализованных городах проектах наблюдается окупаемость в виде экономической прибыли или за счет большого социального эффекта.

1.3 Исследование отечественных проектов «Умный город»

Фактический этап развития «умных» город в РФ характеризуется точечным развитием. Умные города возникают как экспериментальные площадки при сильной поддержке правительства. В использовании проектов концепции «Умный город» не заинтересованы и не замотивированы ни власти, ни застройщики, ни население. На данный момент застройщики строят доступное жилье, а не «современное», что соответствует существующему спросу.

Как правило, российское понимание «умного» города в основном совпадает с использованием информационных технологий для управления коммунальным хозяйством. Хотя понятие намного шире, к «умному» городу также относятся транспортная инфраструктура, медицина, образование, экологическая обстановка, общественная безопасность и даже общественная и культурная сфера.

В настоящее время к различным изменениям в России относятся довольно негативно. Однако это не мешает правительству выделять бюджеты на новые проект концепции «Умный» город. В развитие города нарушается привычная жизнь и приходится привыкать к изменениям.

Из ярких примеров российских проектов, которые подходят под российское понимание «умный» город выделяются «Активный гражданин» в Москве и «Наш Санкт-Петербург» в Санкт-Петербурге. Под западное понимание термина подходит проект «Умный и безопасный город Казань» и Иннополис в городе Казань.

1.3.1 «Активный гражданин»

С целью реализации потребностей развивающегося гражданского общества и адаптации прогресса технологий правительство Москвы запустило проект «Активный гражданин», ставший очередным шагом в процессе сближения правительства Москвы и гражданского общества. Проект

представляет собой систему электронных опросов о ключевых городских темах: об общественном транспорте, о нестационарных торговых объектах, рекламных и информационных конструкциях, учреждениях здравоохранения, объектах хозяйственной деятельности, учреждениях социальной защиты и др. Пользователи при регистрации указывают свой адрес и участвуют в опросах окружного и районного значения. Принять участие в референдумах в рамках проекта «Активный гражданин» можно на сайте проекта или с помощью мобильного приложения посредством установки программы на смартфон на базе iOS или Android, идентификации по номеру мобильного телефона и заполнения профиля участника. Данная информация способствует сетевому продвижению проекта.

Пилотный проект «Активный гражданин» стартовал с запуском в 2011 г. Портала «Наш город» и приложения «Мобильная приемная», с помощью которых стал возможен общественный контроль качества выполняемых городскими службами работ и оказания услуг. Еще одним поводом для запуска проекта является инициатива проведения проекта «Московское парковочное пространство» в части публичного обсуждения введения платной парковки в центре Москвы. Инициатива проведения референдума принадлежит партии «Справедливая Россия». При организации референдума потребовалось бы 30 дней, чтобы собрать 2% избирателей, что составляет около 145 тыс., а что бы итоги референдума были признаны, в голосовании должны участвовать не менее 50% москвичей. Традиционная процедура организации референдума в соответствии с законом города Москвы о референдуме связана со значительными рисками неявки большого числа избирателей и отсутствие минимально необходимо числа подписей. Поэтому проект электронных референдумов является ответом на обозначившийся в общественности запрос на проведение публичных консультаций. По мнению мирового идеолога соучаствующего в проектировании Генри Саноффа [7], основной целью участия в городской политике является стимулирование гражданского образования что является результатом привлечения горожан к решению

городских вопросов. Процесс обучения возникает в том случае, когда процесс понятен, налажены коммуникации, существует открытость доступа к участию, вследствие чего возникает диалог, дебаты и сотрудничество.

Название проекта является символичным, поскольку о желании более активно обсуждать с властями городские проблемы заявили 77% респондентов, являющихся пользователями Московского портала госуслуг и портала «Наш город», что говорит о том, что феномен гражданского активизма подкрепляется с помощью создания открытых площадок публичного диалога.

В таблице 2 отражен возрастной состав участников голосований.

Таблица 2 – Возрастной состав участников голосования [8]

Возраст	Доля участников, %
младше 18 лет	2
18 – 24 года	26
25 – 34 года	49
35 – 44 года	17
свыше 45 лет	6

За первые три месяца существования в проекте участвовали 260 тыс. пользователей. Еженедельно поступает около 11 тыс. сообщений, что в 2.5 раза больше, чем жалоб, которые приходят в мэрию Москвы на бумаге. В проекте принимают участие профильные департаменты города Москвы в целях изучения мнения москвичей. К примеру, Департамент торговли и услуг провел референдум «Торговые объекты на улицах Москвы».

В таблице 3 приведено распределение ответов на один из вопросов референдума.

Таблица 3 – Распределение ответов на вопрос: «Поддерживаете ли вы идею развития в городе магазинов шаговой доступности?» [8]

Вариант ответа	Доля участников, %
Да	84,20
Нет	10,94
Затрудняюсь ответить	4,87

Опрос проводился по ключевым темам в фиксированный период времени (21 мая – 30 мая 2014 г.), в течение которого проголосовал 51 581 москвич. На основании голосований были приняты конкретные практические решения: Департамент стал более активно развивать мобильную межсезонную торговлю; развивать малоформатные сетевые магазины шаговой доступности; освобождать городское пространство от нестационарных торговых объектов; ограничивать продажу пива в нестационарных торговых объектах; ограничивать продажу продуктов питания из автомагазинов. Результаты опросов во многом ожидаемы, но вместе с тем гражданская поддержка стала инструментом противодействия лобби со стороны предпринимателей в целях организации цивилизованных форматов торговли.

На данный момент в проекте «Активный гражданин» насчитывается более 5 000 000 активных пользователей, 5000 прошедших голосований и более 150 000 000 принятых мнений по тем или иным вопросам.

В таблице 4 приведено распределение ответов на одно из последних голосований в проекте «Активный гражданин»

Таблица 4 – Распределение ответов на голосование: «Выбор онлайн-форматов мероприятий на ВДНХ» [9]

Вариант ответа	Доля участников, %
Видеоэкскурсии	31,43
Видеотрансляции концертов и спектаклей	20
Мастер-классы онлайн	19,98
Видеолекции	13,47
Хочу предложить свой вариант	0,19
Мне не интересен онлайн-формат мероприятий	14,93

Всего в голосовании приняли участие 181 333 человека. Голосование проведено с 24.11.2020 по 10.01.2021 г. По итогам голосования активные граждане выбрали онлайн-форматы мероприятий, к которым относятся: видеоэкскурсии, мастер-классы и видеотрансляции концертов и спектаклей. В январе, феврале и марте 2021 года было проведено 17 мероприятий. Программа была разработана с учетом мнения активных граждан.

1.3.2 «Наш Санкт-Петербург»

Портал «Наш Санкт-Петербург» известен как самый крупный городской интернет-ресурс, где принимаются сообщения по вопросам ЖКХ и городского благоустройства. Портал работает с 2014 года, и за это время стал самой массовой «жалобной книгой» города, у него более 200 тысяч

зарегистрированных пользователей и более 3 миллионов принятых сообщений.

Портал «Наш Санкт-Петербург» создан по инициативе Губернатора Санкт-Петербурга для оперативного взаимодействия жителей города с представителями органов власти Санкт-Петербурга.

Используя портал «Наш Санкт-Петербург», пользователь может [10]:

- направить сообщения о проблемах, связанных с жилищно-коммунальным хозяйством и благоустройством города, состоянием дорог и тротуаров, незаконными объектами строительства и торговли, нарушением земельного или миграционного законодательства;

- проинформировать городские службы об отсутствии справочной информации на информационных стендах, а также неудовлетворительном санитарном состоянии помещений в бюджетных учреждениях сферы образования, здравоохранения, культуры, социальной защиты населения, сферы занятости;

- получить дополнительную информацию, касающуюся адресных городских программ и управляющих организаций, а также справочную информацию по интересующему объекту городского хозяйства;

- ознакомиться с технико-экономическими паспортами многоквартирных домов Санкт-Петербурга и получить сведения об организациях, их обслуживающие.

- получать информацию о ходе рассмотрения и отработки направленных Вами сообщений;

- оценить полученный ответ.

Сообщения, направленные через портал «Наш Санкт-Петербург», в обязательном порядке рассматриваются городскими службами в строго установленные сроки.

Процедура регистрации на портале предельно проста и понятна. Необходимо указать только ФИО, электронную почту и мобильный телефон,

на которые придут письмо и код с подтверждением. Регистрация через ЕСИА нужна только для сообщений о нарушениях земельного законодательства. [11]

В профиле пользователя отображаются лишь имя и первая буква фамилии (например, Мария Л.), никакая другая информация о пользователе не представляется и не передается.

Пользователь может зарегистрироваться под своим реальным именем, но это не является обязательным требованием, ведь на необходимость решения проблемы — если она действительно есть — это никак не влияет. К тому же многие люди по различным причинам не хотят подписываться своими реальными именами. Достоверность же самой проблемы подтверждается фото, которое пользователь прикладывает к своему сообщению.

Отправка сообщения занимает пару минут. Пользователю, увидевшему проблему, необходимо сфотографировать, приложить фото к тексту с описанием проблемы и отправить на портал. Выбрать нужную категорию для сообщения поможет нейросеть: в поисковой строке достаточно ввести три ключевых слова по проблеме, и нейросеть дальше уже сама предложит подходящие категории.

Портал работает по принципу «одного окна»: пользователи оставляют свои сообщения по одной из 200 категорий, а дальше эти сообщения автоматически маршрутизируются и разлетаются по исполнителям. То есть за каждой конкретной категорией закреплён конкретный исполнитель (районная администрация, профильный комитет и т. д.), который отрабатывает эту жалобу. Это сделано для максимальной оперативности решения проблем и для того, чтобы сами пользователи могли видеть, кто и как работает с их жалобой. Если в одном сообщении пользователь пожалуется и на лампочку в парадной, и на песок в песочнице, и на заброшенный автомобиль во дворе, то автоматизировать и быстро отработать такое сообщение уже не получится.

1.4 Результаты и выводы

В этой главе были рассмотрены общие понятия «умного» города, разделение определения на западное и отечественное толкования. Также были выявлены отличия в подходах к концепции «умный» город в разных государствах мира. Далее был проведен анализ отечественных проектов, подходящих под концепцию «умный» город. Для анализа были выбраны столичные проекты – Московский «Активный гражданин» и Санкт-Петербургский портал «Наш Санкт-Петербург». В результате проведения анализа и сбора статистики был сделан вывод, что выгоднее всего разрабатывать систему по российскому толкованию определения «умный» город. В итоге на данный момент в городе Екатеринбург разрабатывается цифровая платформа «Активный Екатеринбург» аналогично Московскому проекту «Активный гражданин».

2 АРХИТЕКТУРА И БИЗНЕС-ПРОЦЕССЫ

Цифровая платформа «Активный Екатеринбург» включает в себя следующие подсистемы:

- закрытая часть платформы;
- открытая часть платформы;
- мобильное приложение;
- система авторизации пользователей.

2.1 Закрытая часть Цифровой платформы «Активный Екатеринбург»

Подсистема «Закрытая часть» Цифровой платформы «Активный Екатеринбург» предназначена для администрирования, автоматизированного сбора, анализа и хранения данных, аккумулируемых всеми подсистемами Цифровой платформы.

Подсистема «Закрытая часть» Цифровой платформы «Активный Екатеринбург» включает в себя следующие модули:

- модуль авторизации;
- модуль распределения доступа;
- модуль «Модерирование заявлений»;
- модуль «Исполнение заявлений»;
- модуль «Администрирование»;
- модуль «Аналитика»;
- модуль «Голосования».

2.1.1 Модуль авторизации

Модуль авторизации должен обеспечивать:

- идентификацию, аутентификацию и авторизацию пользователей;
- проверку прав пользователей на доступ в Систему;
- регистрацию изменений данных пользователей.

Учетные записи и информация, необходимая для проверки паролей, должны храниться в базе данных. Исключено хранение паролей и другой безопасно-значимой информации в открытом виде.

Бизнес-процесс авторизации закрытой части представлен на рисунке 1.

2.1.1 Модуль распределения прав доступа

Модуль должен определять правила доступа пользователей к функциям Системы, и осуществлять контроль доступа.

Модуль разграничения прав доступа должен обеспечивать защиту информации, хранимой и обрабатываемой Системой от несанкционированного доступа на основе установленной ролевой модели.

Модуль должен иметь возможность добавления объектов доступа, субъектов доступа и их прав, как на объект в целом, так и на отдельные его характеристики.

В системе должна быть предусмотрена гибкая ролевая модель, которая позволит создавать любой произвольный набор ролей и гибко регулировать для него набор доступных операций из заранее заложенного в систему списка этих операций.

Пользователям в системе должна быть присвоена роль, которая определяет список доступных подсистем и модулей. Подсистемы и модули назначаются на роль и добавить дополнительную подсистему или модуль одному пользователю в рамках его роли невозможно. Один пользователь может иметь несколько ролей в Системе, права по которым «суммируются».

Модуль разграничения прав доступа должен реализовывать атрибутивную систему доступа к объектам Системы – доступ на выполнение той или иной функции Системы обеспечивается за счет предоставления доступа к параметрам объекта и методам, оперирующими данным объектом в Системе.

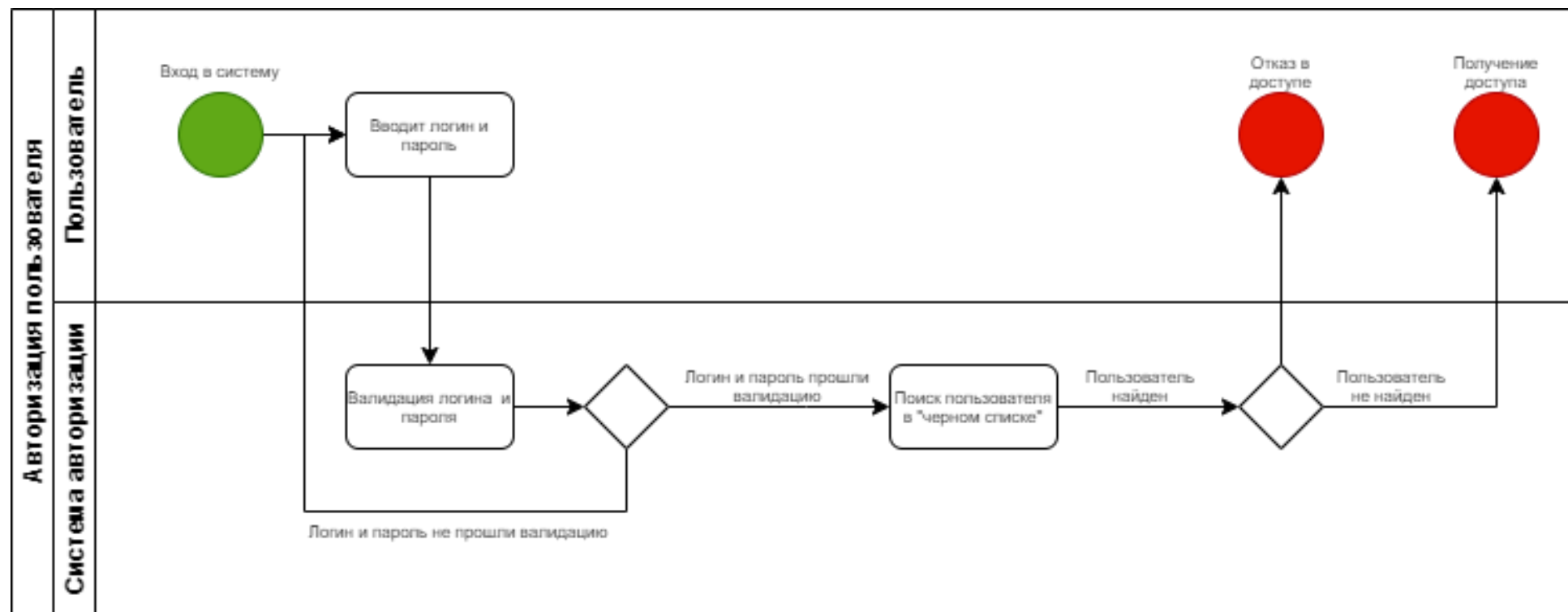


Рисунок 1 – Бизнес-процесс авторизации закрытой части цифровой платформы «Активный Екатеринбург»¹

¹ Составлено автором по: [12].

Назначение прав доступа осуществляется администратором информационной Системы.

В системе должны быть предусмотрены следующие роли, представленные в Таблице 5.

Таблица 5 – Права доступа и роли пользователей Цифровой платформы
[составлено автором]

Роль	Функции
Администратор	Создание ролей
	Создание пользователей Закрытой части платформы и назначение прав доступа
	Управление учётными данными пользователей
	Управление нормативно-справочной информацией Цифровой платформы
	Формирование структур заявлений с помощью конструктора заявлений
	Просмотр заявлений
	Создание уточняющих вопросов по заявлению
	Изменение статуса заявления
	Создание окончательных и промежуточных ответов по заявлению
	Контроль сроков исполнения и ответов на заявления
	Создание опросов и голосований с помощью конструктора опросов
	Открытие/закрытие/приостановление проведения голосования
	Просмотр результатов голосований и опросов
	Формирование статистических и аналитических данных

Продолжение таблицы 5 – Права доступа и роли пользователей
Цифровой платформы

Роль	Функции
Исполнитель	Просмотр заявлений
	Создание уточняющих вопросов по заявлению
	Изменение статуса заявления
	Создание окончательных и промежуточных ответов по заявлению
	Контроль сроков исполнения и ответов на заявления
Контролер	Контроль сроков исполнения и ответов на заявления
	Контроль ответов на заявления
Модератор контента	Управление контентом открытой части Цифровой платформы

2.1.2 Модуль «Модерирование заявлений»

Модуль «Модерирование заявлений» предназначен для предварительной проверки и обработки заявлений от заявителей (граждан) с целью уточнения значений полей заявления, отклонения заявления в случае нарушения заявителем правил пользования Цифровой платформы, а также контроля сроков модерирования и исполнения заявлений.

Модуль «Модерирование заявлений» представлен в виде набора графических веб-интерфейсов, предназначенных для модерирования заявлений, управления и контроля за сроками исполнения и ответами на заявления граждан в рамках установленного классификатора.

Модуль «Модерирование заявлений» должен предоставлять пользователю следующие функции:

- просмотр заявлений списком или на карте в соответствии с геопозиционированием заявления. Интерактивная карта, отображаемая через графический веб-интерфейс, должна

позволять пользователю Цифровой платформы просматривать территориальное расположение объектов улично-дорожной сети МО «города Екатеринбурга» в разрезе территориально-административного деления города Екатеринбурга и визуализировать информацию о поступивших заявлениях с применением деловой графики. В качестве картографической основы используются карты из открытых источников (OpenStreetMap, Яндекс.Карты, Google Карты);

- проведение предварительной проверки заявлений граждан с целью уточнения данных по заявлению. Реализация чата с гражданином с возможностью отправки медиа-файлов и ограничением взаимодействия со стороны гражданина – путём запрета отправки сообщения без разрешения администратора Цифровой платформы о начале информационного взаимодействия;
- отклонение заявления в случае нарушения гражданином правил пользования Web-сайта или мобильного приложения Цифровой платформы «Активный Екатеринбург» с возможностью отправки текстового комментария и изменения статуса рассмотрения заявления;
- контроль сроков исполнения заявления и предоставления ответа на заявление с помощью цветовой схемы. В случае, если по заявке не были проведены должные работы и не дан ответ в установленные сроки, функция контроля сроков исполнения должна сигнализировать администратору и ответственным специалистам о невыполнение работ;
- установление наблюдения за ходом работы с заявлениями путем подписки на события, связанные с заявлением;

- сортировка по категориям заявлений в соответствии с классификатором и ответственными исполнителями, поиск заявлений по параметрам.

Модуль «Модерирование заявлений» обеспечивает:

- ввод информации;
- хранение и модификацию информации;
- физическую и логическую целостность и непротиворечивость информации.

Доступ к функционалу Модуля «Модерирование заявлений» должен определяться ролевой моделью Цифровой платформы.

Бизнес-процесс модуля «Модерирование заявлений» представлен на рисунке 2.

2.1.3 Модуль «Исполнений заявлений»

Модуль «Исполнение заявлений» предназначен для принятия в работу заявлений Исполнителем (ответственным ведомством), осуществления связи между исполнителем и заявителем путем направления уточняющих вопросов и размещения исполнителем ответа на заявления.

Модуль «Исполнение заявлений» должен предоставлять пользователю следующие функции:

- просмотр заявлений списком или на карте в соответствии с геопозиционированием заявления. Интерактивная карта, отображаемая через графический веб-интерфейс, должна позволять пользователю Цифровой платформы просматривать территориальное расположение объектов улично-дорожной сети МО «города Екатеринбурга» в разрезе территориально-административного деления города Екатеринбурга и визуализировать информацию о поступивших заявлениях с применением деловой графики. В качестве картографической

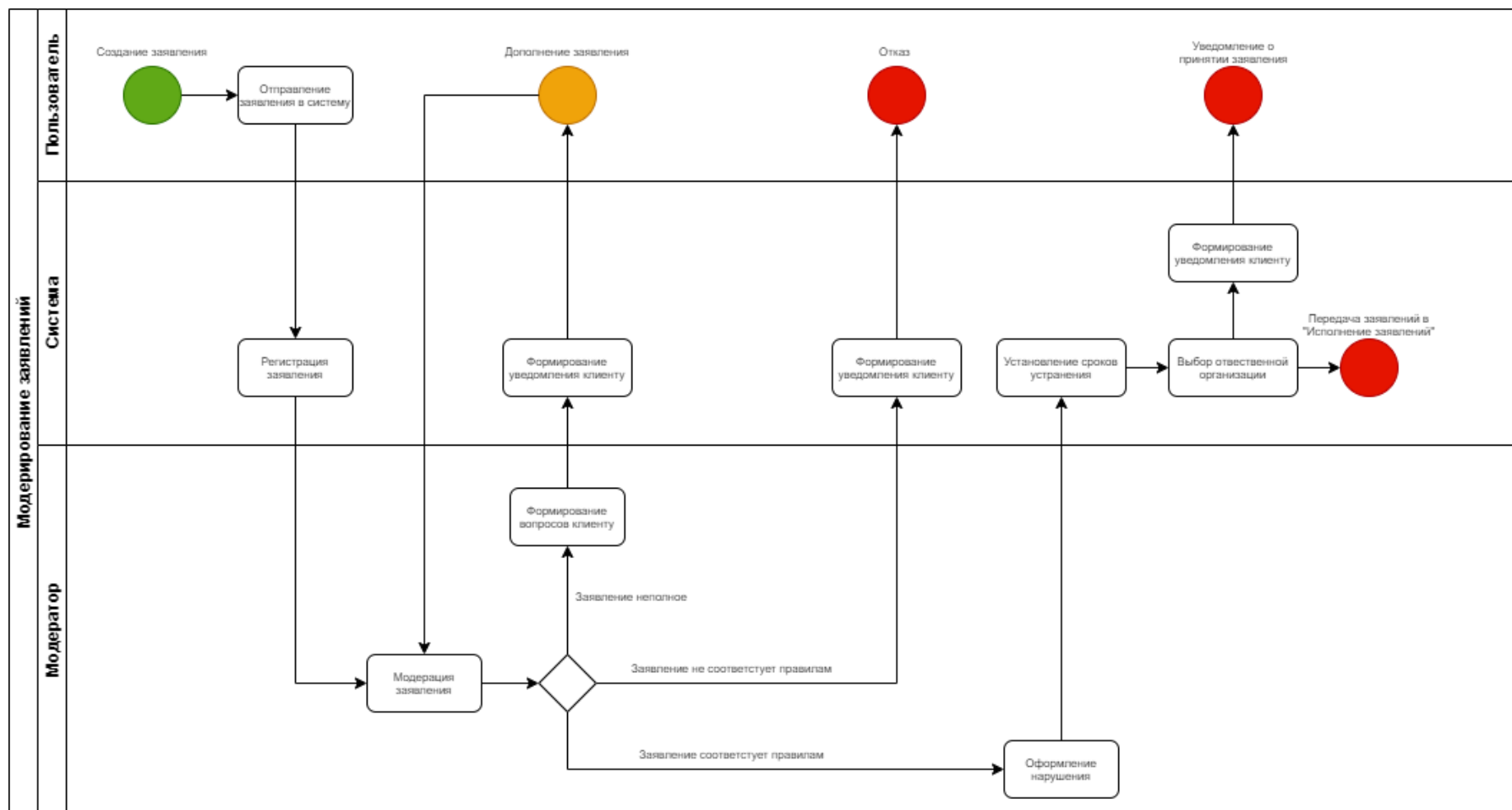


Рисунок 2 – Бизнес-процесс модуля «Модерирование заявлений»²

² Составлено автором по: [12].

основы используются карты из открытых источников (OpenStreetMap, Яндекс.Карты, Google Карты);

- формирование уточняющих вопросов по заявлениям с использованием чата;
- формирование промежуточного или окончательного ответа гражданину с указанием категории ответа (проблема устранена, обещание устранения проблемы), срока устранения проблемы, исполнения заявлений, текста ответа, прикрепленных фотоматериалов;
- изменение статуса обработки заявлений в соответствии со статусной моделью;
- сортировка по категориям заявлений в соответствии с классификатором и ответственными исполнителями, поиск заявлений по параметрам.

Исполнение заявлений граждан подтверждается при помощи элементов графического интерфейса в заявлении, доступ к которому определяется ролевой моделью Цифровой платформы.

Заявления, исполнение по которым завершено или прекращено, должны храниться в архивной части базы данных Цифровой платформы.

Бизнес-процесс модуля «Исполнение заявлений» представлен на рисунке 3.

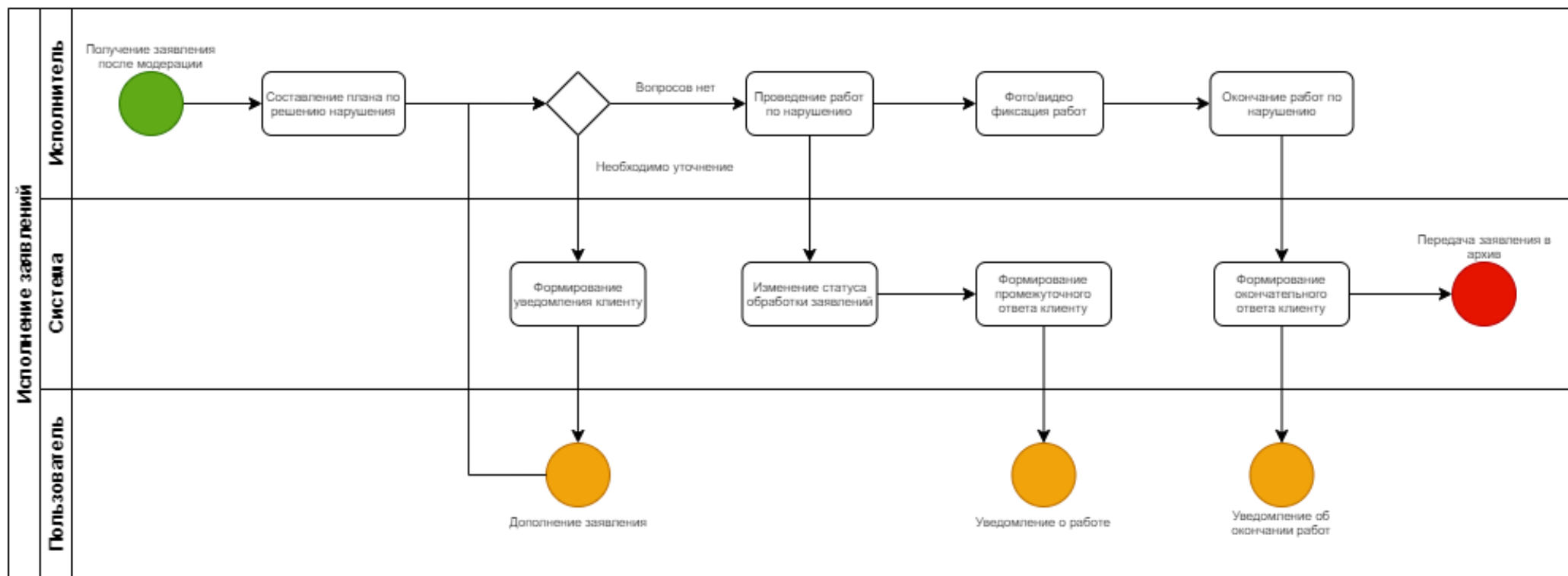


Рисунок 3 – Бизнес-процесс модуля «Исполнение заявлений»³

³ Составлено автором по: [12].

2.1.4 Модуль «Администрирование»

Модуль «Администрирование» предназначен для управления настраиваемыми параметрами конфигурации Цифровой платформой, учетными записями пользователей Закрытой подсистемы платформы, ведения нормативно-справочной информации и классификатора заявлений.

Модуль «Администрирование» должен предоставлять пользователям экранные формы для ввода и редактирования нормативно-справочной информации Цифровой платформы через стандартизированный программный интерфейс.

Модуль «Администрирование» должен предоставлять пользователю следующие функции:

- создание пользователей закрытой части Цифровой платформы «Активный Екатеринбург» и управление их учётными записями;
- ведение нормативно-справочной информации: классификатора категорий заявлений, с указанием исполнителя - ответственных ведомств ОМСУ, нормативных сроков ответов и исполнения заявлений в соответствии с категориями;
- формирование структур заявлений с помощью конструктора заявлений. Конструктор заявлений должен представлять пользователям возможность формирования общей структуры заявления и исключительных структур для специфичных категорий заявлений.

Доступ к функционалу Модуля «Администрирование» должен определяться ролевой моделью Цифровой платформы.

2.1.5 Модуль «Аналитика»

Модуль «Аналитика» предназначен для автоматизированного сбора, анализа, хранения и визуализации статистических данных, формирования

аналитических отчетов по принятым и исполненным заявлениям в различных разрезах.

Модуль «Аналитика» должен производить массовую выборку необходимых данных Цифровой платформы «Активный Екатеринбург» и их групповую подготовку для визуализации данных.

Модуль «Аналитика» обеспечивает формирование статистических данных в соответствии с условиями задачи.

Модуль представлен в виде программного интерфейса для формирования и просмотра:

- информации по заявлениям в соответствии с классификатором;
- информации о исполнении/неисполнении заявлений;
- статистических данных о количестве заявлений в различных разрезах: категории, исполнители, факт исполнения заявления в установленный срок и т. д.;
- статистических данных о количестве заявлений в различных разрезах в заданные промежутки времени;
- показателей эффективности.

Модуль «Аналитика» должен осуществлять выборку и расчет данных по запросам пользователей, группировку, сортировку, и приведение данных к виду, необходимому для построения таблиц, в целях визуализации сформированных с помощью средств деловой графики аналитических отчетов в пользовательском интерфейсе модуля.

Также модуль должен содержать графический веб-интерфейс для загрузки электронных документов пользователем. Документы, сформированные в Системе, должны обладать возможностью быть переданными в печать на бумажных носителях в формате Word и Excel.

2.1.6 Модуль «Голосования»

Модуль «Голосования» предназначен для создания опросов и голосований с использованием конструктора создания опросов, проводимых

среди граждан, открытия и прекращения опросов и голосований, а также автоматизированного сбора данных голосования и подведения итогов, предоставления статистической и аналитической информации.

Модуль «Голосования» должен предоставлять пользователю следующие функции:

- создание опросов и голосований с помощью конструктора опросов. Конструктор опросов представляет собой экранные формы программного интерфейса, позволяющие создавать произвольный набор вопросов и ответов с установленными правилами. Настраиваемые правила ответов к вопросам:
 - а) Выбор одного или нескольких ответов;
 - б) Возможность ввода своего варианта ответа в текстовом формате;
- создание таргетированных опросов и голосований. В конструкторе опросов должны быть реализованы настраиваемые параметры конфигурации опроса и голосования – таргетирование по признакам: возрастному, территориальному (в соответствии с местом нахождения по геометке или адресом регистрации), половому и т. д.;
- возможность указания периодов проведения опросов или голосований, путем назначения даты начала и даты завершения прохождения опроса, а также функционал открытия, приостановления или завершения проведения опроса или голосования;
- возможность ограничения по минимальному значению индекса доверия пользователя;
- предоставление пользователю информации о результатах проведения опросов или голосований в статистическом или графическом виде;

- формирование статистических данных по опросам и голосованиям в различных разрезах в соответствии с ключевыми полями профилей граждан (пол, возраст, территориальное присутствие, индекс доверия и т. д.);
- представление результатов опросов и голосований граждан в Открытой части и мобильном приложении Цифровой платформы «Активный Екатеринбург».

Карта закрытой части платформы «Активный Екатеринбург» представлена на рисунке 4.



Рисунок 4 – Карта закрытой части платформы «Активный Екатеринбург»⁴

⁴ Составлено автором по: [13]

2.2 Открытая часть Цифровой платформы «Активный Екатеринбург»

Подсистема «Открытая часть» Цифровой платформы «Активный Екатеринбург» предназначена для организации публичного доступа к Цифровой платформе авторизованных и неавторизованных Пользователей посредством сети Интернет с использованием web-интерфейса (сайта).

Подсистема «Открытая часть» Цифровой платформы «Активный Екатеринбург» включает в себя следующие модули:

- модуль «Личный кабинет»;
- модуль «Заявления»;
- модуль «Карта»
- модуль «Голосования».

2.2.1 Модуль «Личный кабинет»

Модуль «Личный кабинет» представлен в виде набора графических веб-интерфейсов, предназначенных для управления профилем пользователя (гражданина), взаимодействия с модератором Цифровой платформы или/и исполнителем заявления, а также настройки рассылок и подписки на события.

Модуль «Личный кабинет» должен предоставлять пользователю следующие возможности:

- возможность редактирования личных данных и дополнения своего профиля в рамках повышения индекса доверия к профилю;
- возможность гражданина подтверждать данные, введенные в профиль с помощью авторизации через ЕСИА;
- взаимодействия гражданина и модератора и/или исполнителя через чат по инициативе ответственного ведомства;
- получения обратной связи от ответственного ведомства в виде изменения статусов рассмотрения заявлений;

- получения обратной связи от ответственного ведомства в виде промежуточных и окончательного ответов на заявления;
- управления подписками на рассылку уведомлений по интересующим событиям;
- настройка автоинформатора по e-mail;
- формирования запроса в техническую поддержку по работе с Цифровой платформой «Активный Екатеринбург».

Доступ к модулю «Личный кабинет» и его функциям предоставляется в соответствии с ролевой моделью Цифровой платформы «Активный Екатеринбург» и индексом доверия профиля.

2.2.2 Модуль «Заявления»

Модуль «Заявления» предназначен для формирования заявлений, просмотра сформированных и отправленных заявлений, просмотра статусов и ответов исполнителей в зависимости от индекса доверия пользователю.

Подсистема «Заявления» предоставляет пользователю возможность:

- создания и отправки заявления с тематикой, соответствующей Цифровой платформе «Активный Екатеринбург» и не нарушающей прав использования Цифрой платформы;
- просмотра опубликованных заявлений (своих и других пользователей Цифровой платформы);
- просмотра статусов рассмотрения заявлений;
- просмотра промежуточных и окончательного ответов на заявления;
- поддержки заявлений (повышения значимости заявлений других пользователей Цифровой платформы).

Интерфейс карточки заявления должен предоставлять возможность модератору и исполнителю получения подробной информации по заявлению гражданина и его контактных данных.

Примерные поля заявления в соответствии со структурой, заданной в конструкторе заявлений:

1. Фамилия, имя, отчество заявителя.
2. Контактные данные.
3. Категория заявления в соответствии с классификатором заявлений.
4. Геолокация заявления (в текстовом формате адреса и координатах).
5. Изображение, фото-подтверждение заявления;
6. Текстовое описание.

К обратной связи для гражданина от модератора и/или исполнителя относится уведомление при изменении статуса заявления.

- отклонено;
- принято в работу;
- передано исполнителю;
- закрыто.

Также к обратной информации относятся развернутые промежуточные комментарии ответственного специалиста, окончательный ответ по заявлению и уточняющие вопросы, касающиеся указанного заявления.

Доступ к модулю «Заявления» и его функциям предоставляется в соответствии с ролевой моделью Цифровой платформы «Активный Екатеринбург» и индексом доверия профиля.

2.2.3 Модуль «Карта»

Модуль «Карта» должен предоставлять пользователям экранные формы для работы с заявлениями в виде пространственных объектов с привязкой к электронной карте местности в соответствии с заданными геоданными заявления. Модуль позволяет осуществлять фильтрацию заявлений по тематике, категориям заявлений и периодам, а также по статусу обработки заявления.

Модуль «Карта» представляет собой интерактивную карту, отображаемую через графический веб-интерфейс Цифровой платформы, позволяющий пользователю просматривать территориальное расположение объектов улично-дорожной сети МО «города Екатеринбурга» в разрезе территориально-административного деления города Екатеринбурга и визуализировать информацию о заявлениях с применением деловой графики.

Функция визуализации картографических данных объектов ИК должна предоставлять возможность:

- отражать местоположение заявления, содержащееся в заявлении гражданина;
- отображать географические параметры объектов улично-дорожной сети МО «город Екатеринбург».

В качестве картографической основы используются карты из открытых источников - OpenStreetMap, Яндекс.Карты, Google Карты.

Доступ к модулю «Карта» и его функциям предоставляется в соответствии с ролевой моделью Цифровой платформы «Активный Екатеринбург» и индексом доверия профиля.

2.2.4 Модуль «Голосования»

Модуль «Голосования» предназначен для участия пользователей Цифровой платформы в опросах и голосованиях, просмотра пользователями результатов проведения опросов и голосований, а также статистических и аналитических данных.

Модуль «Голосования» предоставляет пользователю возможность:

- просмотра доступных голосований и опросов и информации по ним;
- просмотра анонсов голосований и опросов, проводимых в рамках Цифровой платформы «Активный Екатеринбург»;
- участия в голосованиях и опросах в соответствии с правилами пользования Цифровой платформы «Активный Екатеринбург» и индексом доверия профиля;
- участия в таргетированных голосованиях и опросах с заданными параметрами;
- однократного голосования;
- просмотра результатов проведенных голосований и опросов в табличном и графическом виде представления информации и статистических данных, опубликованных модераторами Цифровой платформы «Активный Екатеринбург»;
- наблюдение за текущими голосованиями и опросами в режиме реального времени.

Доступ к модулю «Голосования» и его функциям предоставляется в соответствии с ролевой моделью Цифровой платформы «Активный Екатеринбург» и индексом доверия профиля.

Диаграмма прецедентов открытой части цифровой платформы «Активный Екатеринбург» представлена на рисунке 5.

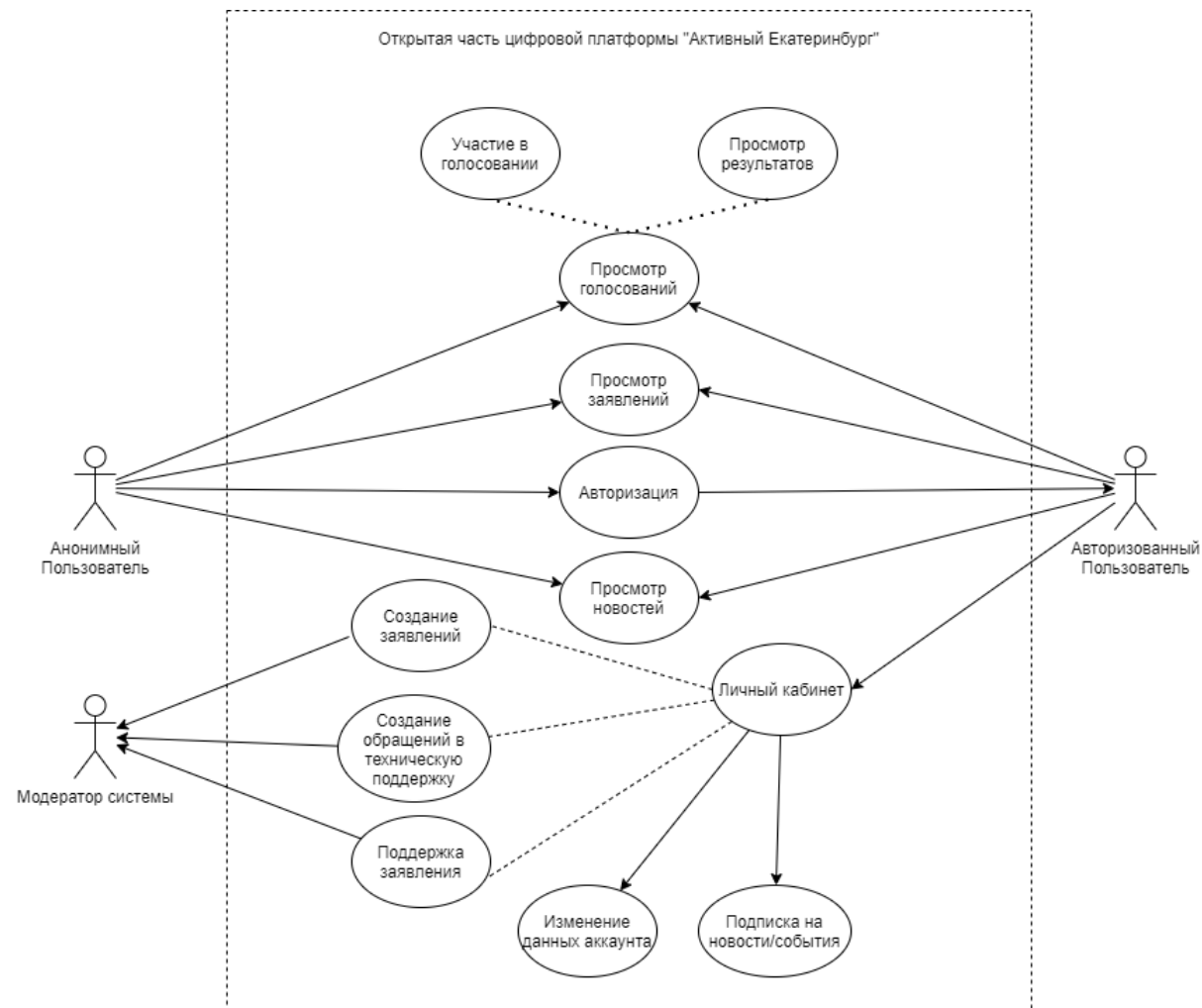


Рисунок 5 – Диаграмма прецедентов открытой части цифровой платформы «Активный Екатеринбург»⁵

⁵ Составлено автором по: [14]

2.3 Мобильное приложение Цифровой платформы «Активный Екатеринбург»

Подсистема «Мобильное приложение» Цифровой платформы «Активный Екатеринбург» предназначена для организации публичного доступа к Цифровой платформе авторизованных и неавторизованных Пользователей посредством мобильного устройства (смартфона, планшетов) с применением установленного мобильного приложения.

Мобильное приложение должно обеспечивать возможность создания и отправки заявления в соответствии с правилами пользования Цифровой платформы «Активный Екатеринбург» и классификатором заявлений, получения обратной связи от ответственных специалистов, участия в голосованиях и просмотра всех заявлений, поданных на территории МО «город Екатеринбург».

Мобильное приложение должно поддерживать работу на мобильных устройствах, работающих на системах Android и iOS.

Мобильное приложение должно быть разработано на платформе React Native и являться неотъемлемой частью Цифровой платформы «Активный Екатеринбург».

Мобильное приложение должно быть опубликовано силами Заказчика в онлайн-магазинах, таких как Apple Appstore и Google Play. Заказчик выполняет все действия необходимые для регистрации и публикации приложения в онлайн-магазинах.

Функционал мобильного приложения, доступный гражданам, должен обеспечивать выполнение следующих базовых функциональных требований к системам автоматизации процессов обратной связи и взаимодействия с гражданами и включать в себя следующие модули:

- модуль «Личный кабинет»;
- модуль «Заявления»;
- модуль «Карта»;

- модуль «Голосования».

Предъявляемые требования к модулям соответствуют требованиям, предъявляемым к подсистеме «Открытая часть» (Web-сайт) Цифровой платформы «Активный Екатеринбург».

2.4 Система авторизации пользователей Цифровой платформы «Активный Екатеринбург»

Система авторизации пользователей Цифровой платформы «Активный Екатеринбург» предназначена для регистрации и авторизации Пользователей на Цифровой платформе посредством Web-Сайта и мобильного устройства (смартфона, планшетов) с применением установленного мобильного приложения.

Система авторизации пользователей Цифровой платформы «Активный Екатеринбург» включает в себя следующие модули:

- модуль авторизации;
- модуль управления профилем пользователя;
- модуль расчета индекса доверия;
- модуль интеграции (API).

2.4.1 Модуль авторизации

Система авторизации предназначена для регистрации и авторизации пользователей (граждан) Цифровой платформы «Активный Екатеринбург», управления идентификацией и доступом.

Система авторизации должна предоставлять пользователям такие функции, как единый вход (SSO), брокерская идентификация и социальный вход в систему и поддерживать следующих провайдеров:

- Facebook;
- ВКонтакте;
- Одноклассники;
- Google;

- ЕСИА.

Пользователю доступна функция регистрация через мобильное приложение и Web-Сайт Цифровой платформы «Активный Екатеринбург» или прямая авторизация через реквизиты доступа - логин/пароль. После регистрации или авторизации в Системе авторизации пользователю доступен Личный кабинет пользователя Цифровой платформы «Активный Екатеринбург» и модуль управления профилем пользователя.

2.4.2 Модуль управления профилем пользователя

Модуль управления профилем пользователя Цифровой платформы «Активный Екатеринбург» предназначена для добавления и изменения данных пользователей, используемых для определения индекса доверия профилям пользователей.

Стандартизированный набор полей данных профиля:

- фамилия;
- имя;
- отчество;
- пол;
- дата рождения;
- СНИЛС;
- телефон;
- электронная почта;
- адрес регистрации;
- место проживания.

Управлять данными профиля должно быть возможно как с помощью прямой авторизации через провайдеров Системы авторизации с использованием разрешения о предоставлении данных, так и путём ручного заполнения данных профиля.

2.4.3 Модуль расчета индекса доверия

Модуль расчета индекса доверия профиля пользователя предназначен для расчёта процентного индекса, позволяющего использовать данную информацию как параметр доступности функционала Цифровой платформы «Активный Екатеринбург».

Индекс доверия профилю пользователя должен рассчитываться исходя из заполненных и незаполненных полей профиля личными данными.

При расчете индекса доверия профилю должна учитываться достоверность данных, введенных пользователем, и соответствовать следующим требованиям:

- подтверждение личных данных через ЕСИА;
- подтверждение личных данных через социальные сети (провайдеров системы авторизации);
- подтверждение личных данных электронным межведомственным взаимодействием с сервисами подтверждения адреса регистрации (УФМС или АИС МРН);
- подтверждение личных данных электронным межведомственным взаимодействием через АИС, аккумулирующие в себе данные заявителей.

Алгоритм расчета индекса доверия профилю должен соответствовать следующей схеме расчёта:

- ФИО + СНИЛС + Дата рождения + Пол = 50%
- + Телефон = 60%
- + Электронная почта = 70%
- + Адреса регистрации = 80%
- + Адрес проживания = 90%
- + Разрешенная геолокация = 100%

2.4.4 Модуль интеграции (API)

Модуль интеграции (API) предназначен для предоставления сторонним сервисам информации о индексе доверия профилю пользователя для определения перечня доступного функционала.

К интеграционному API предъявляются следующие требования:

- API должно ограничивать доступ клиентов;
- API должно идентифицировать клиента;
- API должно обрабатывать запросы только в синхронном режиме;
- API принимает данные только в формате JSON;
- API должно валидировать данные;
- если вызов метода производится с неверными параметрами, API должно вернуть ошибку с кодом и пояснением;
- логирование запросов и ответов.

На стороне Системы авторизации пользователей Цифровой платформы «Активный Екатеринбург» должно осуществлять логирование запросов и ответов, информация по которым хранится в базе данных в формате JSON. Ошибки при записи логов запросов и ответов в базу должны храниться в текстовом файле «Журнал событий» в формате JSON.

Бизнес-процесс системы авторизации пользователей Цифровой платформы «Активный Екатеринбург» представлен на рисунке 6.

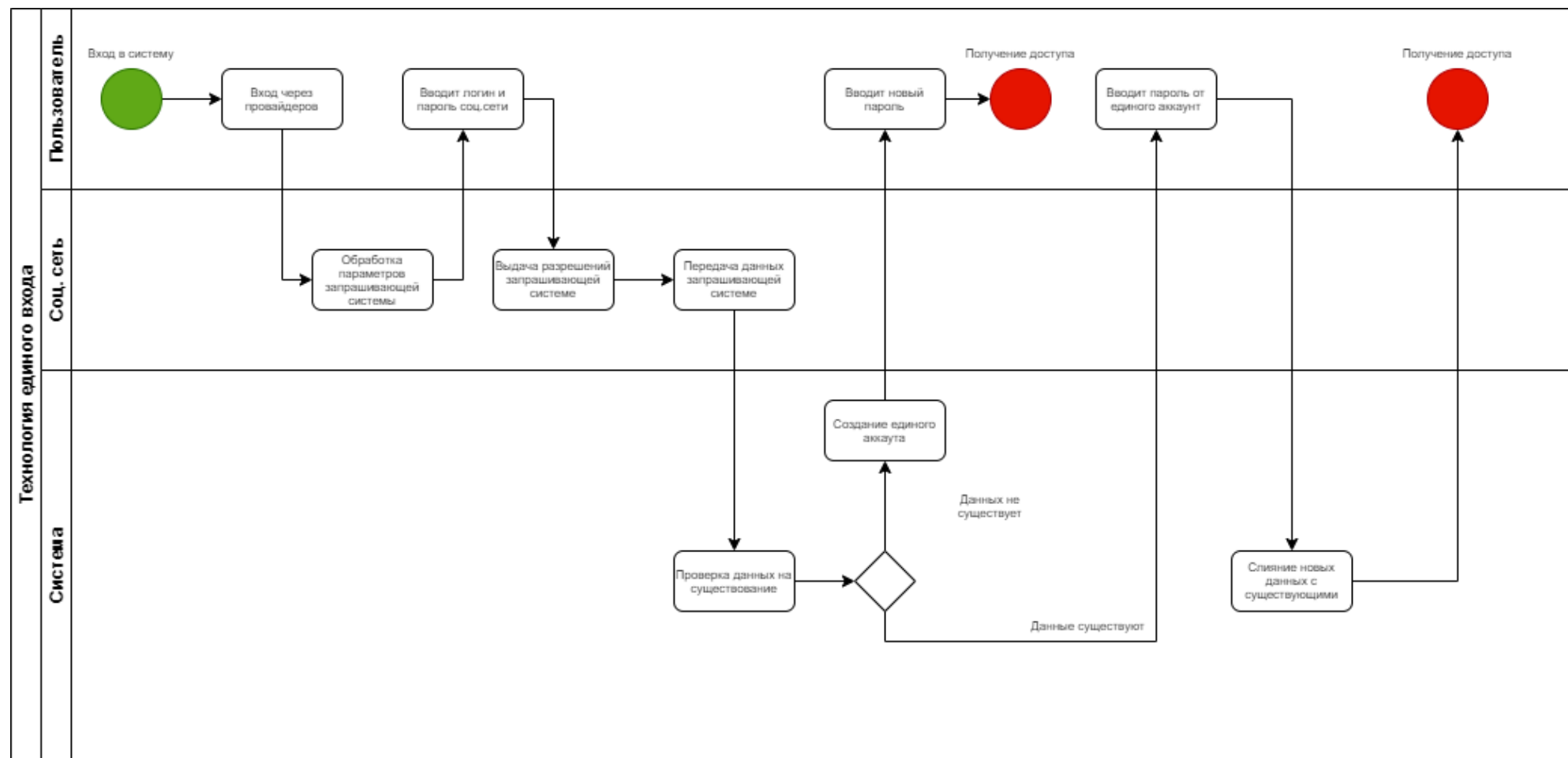


Рисунок 6 – Бизнес-процесс системы авторизации пользователей Цифровой платформы «Активный Екатеринбург»⁶

⁶ Составлено автором по: [12].

2.5 Результаты и выводы

В этой главе были рассмотрены основные представления архитектуры цифровой платформы «Активный Екатеринбург». По данным представлениям были разработаны бизнес-процессы и диаграмма прецедентов для лучшего понимания разработки системы.

Приняв во внимание все аспекты разработки цифровой платформы, была разработана архитектура, представленная на рисунке 7.

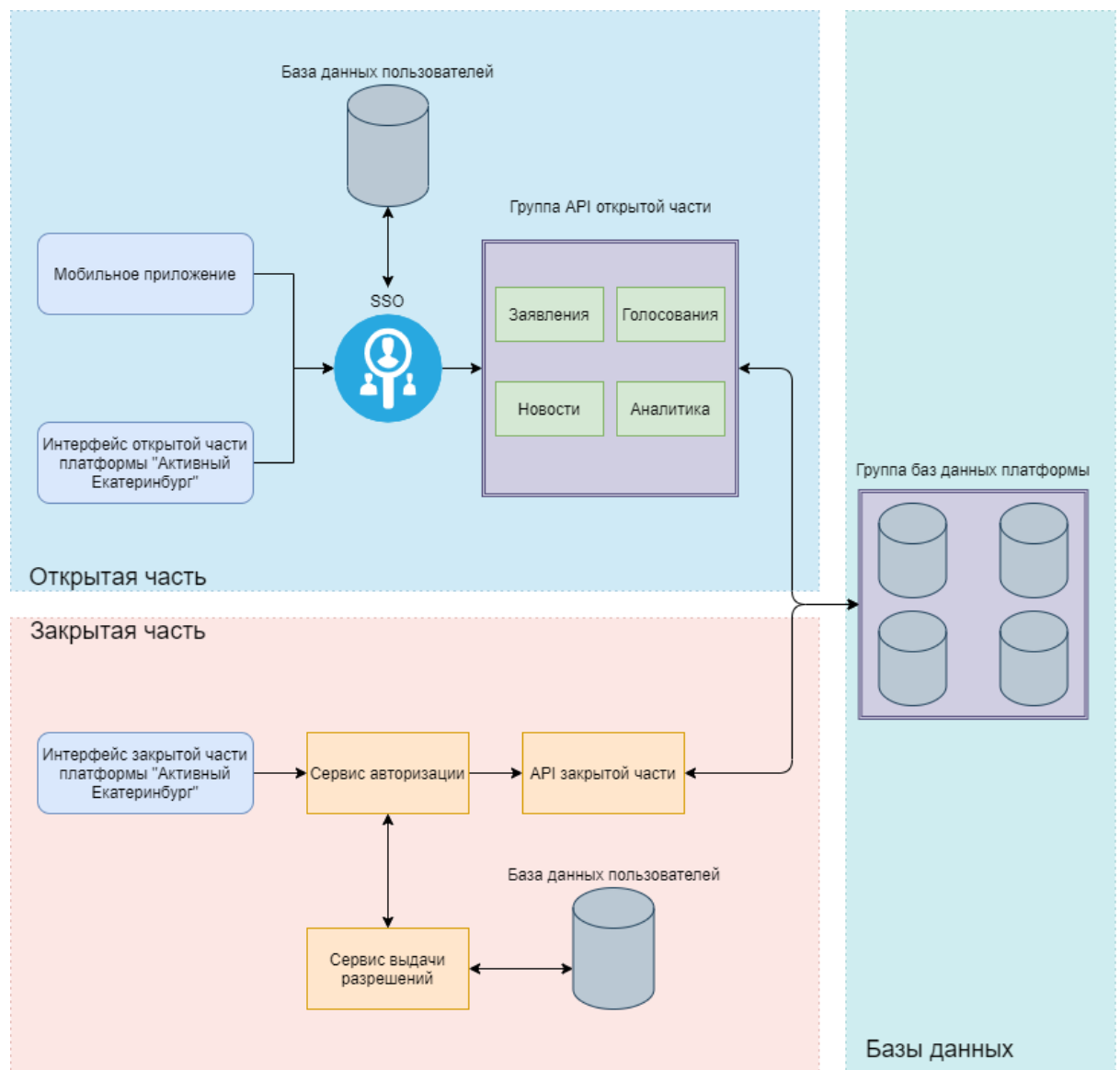


Рисунок 7 – Архитектура цифровой платформы «Активный Екатеринбург»⁷

⁷ Составлено автором по: [15].

3 РАЗРАБОТКА ЦИФРОВОЙ ПЛАТФОРМЫ «АКТИВНЫЙ ЕКАТЕРИНБУРГ»

3.1 Применяемые технологии

3.1.1 Клиентская часть

В качестве основного языка для разработки клиентской части и мобильного приложения использовался JavaScript, а именно библиотека React.

React — JavaScript-библиотека для работы с пользовательскими интерфейсами (UI), которую создали разработчики Facebook. Библиотеку начали использовать на сайте этой социальной сети в 2011 году. А в 2013 году Facebook открыл исходный код React [16].

С помощью React разработчики создают веб-приложения, которые изменяют отображение без перезагрузки страницы. Благодаря этому приложения быстро реагируют на действия пользователя, например, заполнение форм, применение фильтров, добавление товаров в корзину и так далее.

React применяют для отрисовки компонентов пользовательского интерфейса. Также библиотека может полностью управлять фронтом. В этом случае React используют с библиотеками для управления состоянием и роутинга, например, Redux [17] и React Router.

Одна из ключевых особенностей React — универсальность. Эту библиотеку можно использовать на сервере и на мобильных платформах с помощью React Native. Это принцип Learn Once, Write Anywhere или «Научитесь один раз – пишите где угодно».

Ещё одна важная особенность библиотеки — декларативность [18]. С помощью React разработчик описывает, как компоненты интерфейса выглядят в разных состояниях. Декларативный подход сокращает код и делает его понятным.

React основан на компонентах, это ещё одна ключевая особенность библиотеки. Каждый компонент возвращает часть пользовательского интерфейса со своим состоянием. Объединяя компоненты, программист создаёт завершённый интерфейс веб-приложения.

Важная особенность React — использование JSX [19]. Это расширение синтаксиса JavaScript, которое удобно использовать для описания интерфейса. JSX похож на HTML, тем не менее это всё-таки JavaScript. Пример JSX представлен на рисунке 8.

```
const header = text ? <h1>{text}</h1> : null;

const vdom = (
  <div>
    {header}
    <Hello />
  </div>
);
```

Рисунок 8 – Пример JSX

3.1.2 Серверная часть

В качестве основного языка программирования рассматривались C# и Java.

C# — язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции [20].

Преимущества [21]:

- для малых предприятий и некоторых отдельных разработчиков бесплатные инструменты включают Visual Studio, Azure Cloud, Windows Server, Parallels Desktop для Mac Pro и многие другие;
- большое количество синтаксических конструкций, разработанных для лучшего понимания написания кода;
- очень прост в изучении;

- после покупки Xamarin на C # вы можете писать программы и приложения для операционных систем, таких как iOS, Android, MacOS и Linux;
- имеется целое сообщество из опытных программистов.

Недостатки:

- приоритетная ориентированность на Windows платформу;
- язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. Крупной компании покупка лицензионной версии этого языка обойдется в круглую сумму;
- в языке осталась возможность использования оператора безусловного перехода.

Java — строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle) [22].

Преимущества [23]:

- в корпоративных приложениях Java способен поддерживать строительные блоки системы или различные библиотеки, с их помощью создают необходимые функции;
- запуск приложений в песочнице с устранением распространённых, уязвимых объектов в соответствии с политикой безопасности;
- независимое положение от платформ;
- повышенная производительность труда благодаря встроенному механизму, чтобы совместно использовать данные программы на нескольких компьютерах;
- возможность автоматического управления памятью с одновременным независимым запуском потоков;

Недостатки:

- в 2019 года компанией Oracle было объявлено о взимании платы за коммерческое использование языка;
- низкая скорость и безопасность;
- отсутствует нативный дизайн;
- многословность и сложность кода;
- большой объем потребления памяти

Взвесив все преимущества и недостатки обзореваемых языков, было вынесено решение, что в качестве основного языка программирования будет взят C#. Главным критерием послужило то, что сервера размещения проекта в основном будут базироваться на ОС Windows. Таким образом, недостаток в приоритетной ориентированности языка C# на Windows платформу обернется в преимущество за счет крепкой связи язык-сервер.

В качестве системы управления базами данных был выбран PostgreSQL.

PostgreSQL — это популярная свободная объектно-реляционная система управления базами данных. PostgreSQL базируется на языке SQL и поддерживает многочисленные возможности [24].

Преимущества [25]:

- поддержка БД неограниченного размера;
- мощные и надёжные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования и поддержка загрузки C-совместимых модулей;
- наследование;
- легкая расширяемость.

Текущие ограничения:

- нет ограничений на максимальный размер базы данных;
- нет ограничений на количество записей в таблице;
- нет ограничений на количество индексов в таблице;
- максимальный размер таблицы — 32 Тбайт;
- максимальный размер записи — 1,6 Тбайт;

- максимальный размер поля — 1 Гбайт;
- максимум полей в записи 250—1600 (в зависимости от типов полей);

Особенности:

Функции в PostgreSQL являются блоками кода, исполняемыми на сервере, а не на клиенте БД. Хотя они могут писаться на чистом SQL, реализация дополнительной логики, например, условных переходов и циклов, выходит за рамки, собственно, SQL и требует использования некоторых языковых расширений. Функции могут писаться с использованием различных языков программирования. PostgreSQL допускает использование функций, возвращающих набор записей, который далее можно использовать так же, как и результат выполнения обычного запроса. Функции могут выполняться как с правами их создателя, так и с правами текущего пользователя. Иногда функции отождествляются с хранимыми процедурами, однако между этими понятиями есть различие [26].

Триггеры в PostgreSQL определяются как функции, инициируемые DML-операциями. Например, операция INSERT может запускать триггер, проверяющий добавленную запись на соответствия определённым условиям. При написании функций для триггеров могут использоваться различные языки программирования. Триггеры ассоциируются с таблицами. Множественные триггеры выполняются в алфавитном порядке [27].

Механизм правил в PostgreSQL представляет собой механизм создания пользовательских обработчиков не только DML-операций, но и операции выборки. Основное отличие от механизма триггеров заключается в том, что правила срабатывают на этапе разбора запроса, до выбора оптимального плана выполнения и самого процесса выполнения. Правила позволяют переопределять поведение системы при выполнении SQL-операции к таблице [28].

Индексы в PostgreSQL следующих типов: B-дерево, хэш, R-дерево, GiST, GIN. При необходимости можно создавать новые типы индексов, хотя это далеко не тривиальный процесс [29].

Многоверсионность поддерживается в PostgreSQL [30] — возможна одновременная модификация БД несколькими пользователями с помощью механизма Multiversion Concurrency Control (MVCC). Благодаря этому соблюдаются требования ACID, и практически отпадает нужда в блокировках чтения.

Расширение PostgreSQL для собственных нужд возможно практически в любом аспекте [31]. Есть возможность добавлять собственные преобразования типов, типы данных, домены (пользовательские типы с изначально наложенными ограничениями), функции (включая агрегатные), индексы, операторы (включая переопределение существующих) и процедурные языки.

Наследование в PostgreSQL реализовано на уровне таблиц. Таблицы могут наследовать характеристики и наборы полей от других таблиц (родительских). При этом данные, добавленные в порождённую таблицу, автоматически будут участвовать (если это не указано отдельно) в запросах к родительской таблице [32].

В разработке простых сайтов PostgreSQL используется несколько реже, чем MySQL / MariaDB, но всё же эта пара с заметным отрывом опережает по частоте использования остальные системы управления базами данных. При этом в разработке сложных сайтов и веб-приложений PostgreSQL опережает по использованию MySQL и MariaDB. Большинство фреймворков (например, Ruby on Rails, Yii, Symfony, Django) поддерживают использование PostgreSQL в разработке.

3.1.3 Авторизация

В качестве решения авторизации для открытой части платформы был выбран Keycloak.

Keycloak — это решение для управления идентификацией и доступом с открытым исходным кодом, предназначенное для использования в ИС, где могут использоваться паттерны микросервисной архитектуры [33].

Keycloak предлагает такие функции, как единый вход (SSO), брокерская идентификация и социальный вход в систему, федерация пользователей, клиентские адаптеры, консоль администратора и консоль управления учетными записями [34].

Базовый функционал, поддерживаемый в Keycloak [35]:

1. Single-Sign On and Single-Sign Out для браузерных приложений.
2. Поддержка OpenID/OAuth 2.0/SAML.
3. Identity Brokering – аутентификация с помощью внешних OpenID Connect или SAML идентификационных провайдеров.
4. Social Login – поддержка Google, GitHub, Facebook, Twitter для идентификации пользователей.
5. User Federation – синхронизация пользователей из LDAP и Active Directory серверов и других идентификационных провайдеров.
6. Kerberos bridge – использование Kerberos сервера для автоматической аутентификации пользователей.
7. Admin Console — для единого управления настройками и параметрами решения через Web.
8. Account Management Console – для самостоятельного управления профилем пользователей.
9. Кастомизация решения на основе фирменного стиля компании.
10. 2FA Authentication – поддержка TOTP/HOTP с помощью Google Authenticator или FreeOTP.
11. Login Flows – возможна саморегистрация пользователей, восстановление и сброс пароля и прочие.
12. Session Management – администраторы могут управлять из единой точки сессиями пользователей.

- 13.Token Mappers – привязка атрибутов пользователей, ролей и иных требуемых атрибутов в токены.
- 14.Гибкое управление политиками через realm, application и пользователей.
- 15.CORS Support – клиентские адаптеры имеют встроенную поддержку CORS.
- 16.Service Provider Interfaces (SPI) – большое количество SPI, позволяющих настраивать различные аспекты работы сервера: потоки аутентификации, идентификационных провайдеров, сопоставление протоколов и многое другое.
- 17.Клиентские адаптеры для JavaScript applications, WildFly, JBoss EAP, Fuse, Tomcat, Jetty, Spring.
- 18.Поддержка работы с различными приложениями, поддерживающими OpenID Connect Relying Party library или SAML 2.0 Service Provider Library.
- 19.Возможность расширения с использованием plugins.

Возможно использование логина из социальных сетей. Для активации возможности аутентифицировать пользователей используется консоль администратора Keycloak. Изменений в коде приложений не требуется и данный функционал доступен «из коробки» и может быть активирован в любой стадии реализации проекта.

JWT (JSON Web Token) — открытый стандарт [36] который определяет компактный и автономный способ для защищенной передачи информации между сторонами в виде JSON-объекта.

Согласно стандарту, токен состоит из трех частей в base-64 формате, разделенных точками. Первая часть называется заголовком (header), в которой содержится тип токена и название хэш-алгоритма для получения цифровой подписи. Вторая часть хранит основную информацию (пользователь, атрибуты и т. д.). Третья часть – цифровая подпись.

Access-токен — это токен, который предоставляет доступ его владельцу к защищенным ресурсам сервера [37]. Обычно он имеет короткий срок жизни и может нести в себе дополнительную информацию, такую как IP-адрес стороны, запрашивающей данный токен.

Refresh-токен — это токен, позволяющий клиентам запрашивать новые access-токены по истечении их времени жизни [39]. Данные токены обычно выдаются на длительный срок.

Основные преимущества применения в микросервисной архитектуре [40]:

- возможность доступа к различным приложениям и сервисам посредством единовременной аутентификации;
- при отсутствии ряда требуемых атрибутов в профиле пользователей возможно дообогащение данными, которые могут быть добавлены в полезную нагрузку, в том числе и автоматизировано и «на лету»;
- отсутствует необходимость хранения информации об активных сессиях, серверное приложение должно только проверить подпись;
- более гибкое управление доступом за счет дополнительных атрибутов в полезной нагрузке;
- применение подписи токена для заголовка и полезной нагрузки повышает безопасность решения в целом;

Исходя из плана проекта было принято решение применить для разработки сервисную архитектуру. Для каждого сервиса будет выделена сущность, разработана база данных и API для общения с клиентской частью. Каждый сервис будет привязан к SSO для проверки данных пользователя и выдачи ему прав.

Для разработки сервисов были выделены следующие сущности:

1. Заявления.
2. Голосования.
3. Новости.

Так же были выделены дополнительные сервисы, которые являются пересечениями этих сущностей:

1. Аналитика.
2. Справочная информация.
3. Распределение прав доступа.

Схема разработанной архитектуры представлена на рисунке 9.

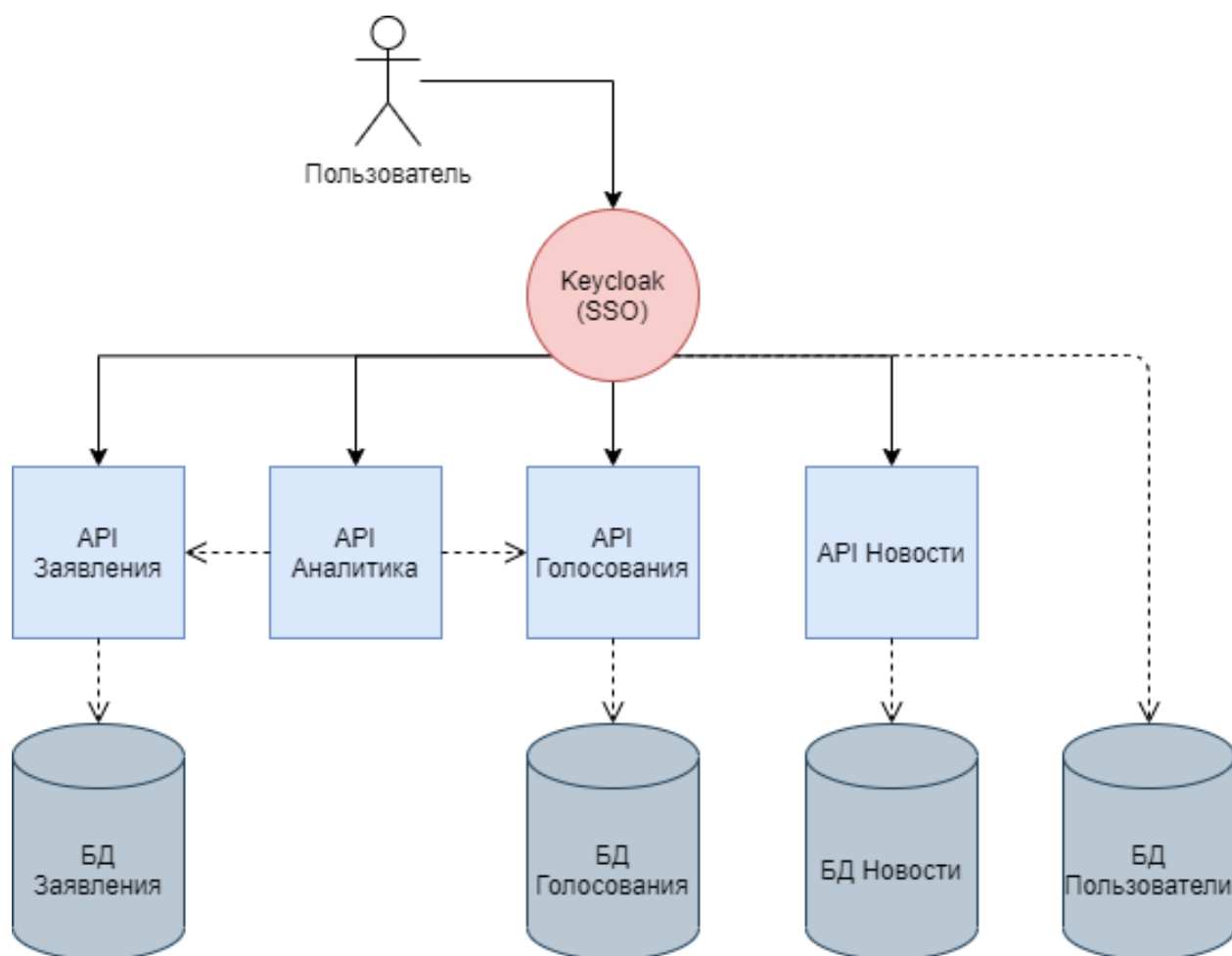


Рисунок 9 – Разработанная архитектура сервисов⁸

⁸ Составлено автором по: [40]

3.2 Описание API

3.2.1 API Заявления

API Заявления включает в себя методы и функции для работы с заявлениями полученными от пользователей цифровой платформы «Активный Екатеринбург».

API включает в себя такие методы как:

1. POST /api/violations/create.

Метод используется для создания заявления. В теле запроса передается форма заполнения заявления.

На рисунках 10, 11 представлены примеры использования кода.

```
/// <summary>|
/// Создание заявления
/// </summary>
/// <param name="registerViolationForm">Форма создания заявления</param>
/// <returns></returns>
[HttpPost(template: "create")]
public ViolationDto Create([FromForm] RegisterViolationForm registerViolationForm)
{
    var violation = violationModelHandler.CreateViolation(registerViolationForm);
    return violation;
}
```

Рисунок 10 – Пример кода в контроллере

```

public ViolationDto CreateViolation(RegisterViolationForm registerViolationForm)
{
    var coordinates = new Coordinates()
    {
        Latitude = double.Parse(registerViolationForm.Latitude, CultureInfo.InvariantCulture),
        Longitude = double.Parse(registerViolationForm.Longitude, CultureInfo.InvariantCulture)
    };

    var district:string = openStreetMapService.GetDistrictByCoordinates(coordinates);
    var districtEnum = EnumHelper.GetEnumByAttribute<District>(district);

    var violation = violationFactory.Create(
        personId: registerViolationForm.UserId,
        categoryId: registerViolationForm.SubCategoryIds.Any() ?
            registerViolationForm.SubCategoryIds.First() :
            registerViolationForm.CategoryId,
        registerViolationForm.Address,
        districtEnum,
        coordinates.Latitude,
        coordinates.Longitude);

    violationRepository.Add(violation);
    violationRepository.SaveChanges();

    var violationHistory = violationHistoryFactory.OnCreate(violation.Id);
    writeViolations.Add(violationHistory);

    writeViolations.SaveChanges();
}

```

Рисунок 11 – Пример кода работы с базой

2. GET /api/Violations/published

Метод используется для получения списка опубликованных модератором заявлений. В параметрах передаются параметры для отображения на странице.

Примеры кода метода представлены на рисунках 12, 13.

```
/// <summary>
/// Получение списка опубликованных заявлений
/// </summary>
/// <param name="page">Номер страницы</param>
/// <param name="number">Количество элементов на странице</param>
/// <returns></returns>
[HttpGet(template: "published")]
public List<ViolationListDto> GetPublished(int page = 1, int number = 8)
{
    var violations :List<ViolationListDto> = violationModelBuilder.GetPublishedViolations(page, number);
    return violations;
}
```

Рисунок 12 – Пример кода в контроллере

```
0+2 usages
public List<ViolationListDto> GetPublishedViolations(int page, int number)
{
    var violations :IPagedList<Violation> = violationRepository.Query(
        x :Violation => x.IsPublished
        && (x.ViolationStatus == ViolationStatuses.finished
            || x.ViolationStatus == ViolationStatuses.moderated
            || x.ViolationStatus == ViolationStatuses.inwork)) // List<Violation>
        .OrderByDescending(x :Violation => x.DateOfCreation) // IOrderedEnumerable<Violation>
        .AsQueryable() // IQueryable<Violation>
        .ToPagedList( pageNumber: page, pageSize: number);

    var violationDtos :List<ViolationListDto> = violations // IPagedList<Violation>
        .Select(x :Violation => violationDtoBuilder.BuildList(x)) // IEnumerable<ViolationListDto>
        .ToList();

    return violationDtos;
}
```

Рисунок 13 – Пример кода работы с базой данных

3. GET /api/violations/{id}

Метод для получения данных по одному заявлению. В параметрах метода передается идентификатор заявления.

На рисунках 12, 13 приведены примеры использования кода.

```
/// <summary>
/// Получение заявления по id
/// </summary>
/// <param name="id">Уникальный идентификатор заявления</param>
/// <returns></returns>
[HttpGet(template: "{id}")]
public ViolationDto Get(Guid id)
{
    var violation = violationModelBuilder.GetViolation(id);
    return violation;
}
```

Рисунок 12 – Пример кода в контроллере

```
public ViolationDto GetViolation(Guid id)
{
    var violation = violationRepository.Find(id);

    if (violation == null) throw new NotFoundException();

    var violationDto = violationDtoBuilder.Build(violation);

    return violationDto;
}
```

Рисунок 13 – Пример кода работы с базой данных

4. POST /api/Violations/{id}/support

Метод для увеличения приоритета заявки. Пользователи имеют возможность единоразово поддержать заявку. В параметрах передается идентификатор заявления и флаг на подписку на события данного заявления.

На рисунках 14, 15 представлены примеры использования кода.

```

/// <summary>
/// Поддержать заявление
/// </summary>
/// <param name="id">Уникальный идентификатор заявления</param>
/// <param name="isSubscribe"></param>
/// <returns></returns>
[HttpPost( template: "{id}/support")]
public IActionResult Support(Guid id, bool isSubscribe = false)
{
    violationModelHandler.SupportViolation(id, isSubscribe);
    return Ok();
}

```

Рисунок 14 – Пример кода в контроллере

```

public void SupportViolation(Guid violationId, bool isSubscribe)
{
    var userId :Guid? = currentKeycloakUserProvider.Current.Id;
    if (userId == null) throw new ForbiddenException();

    var violation = violationRepository.Find(violationId);
    if (violation == null) throw new NotFoundException();

    var support = supportFactory.Create(userId.Value, violationId, isSubscribe);

    try
    {
        writeViolations.Add(support);
        writeViolations.SaveChanges();
    }
    catch
    {
        throw new AisaeException( message: "Уже поддержано вами");
    }
}

```

Рисунок 15 – Пример кода в работе с базой данных

3.2.2 API Голосований

API Голосований включает в себя методы и функции для работы с голосованиями. Выбор варианта голосования, получения результатов голосования, просмотр аналитики по ответам, полученным от пользователей цифровой платформы «Активный Екатеринбург».

API включает в себя следующие методы:

1. GET /api/votes/available

Метод для получения списка доступных голосований для пользователей цифровой платформы «Активный Екатеринбург». Содержит в себе параметры для отображения на странице.

На рисунках 16, 17 приведены примеры использования кода.

```
/// <summary>
/// Получение списка доступных голосований
/// </summary>
/// <param name="page">Номер страницы</param>
/// <param name="number">Количество отображаемых элементов</param>
/// <returns></returns>
[HttpGet(template: "available")]
public VoteCardModelHolder GetAvailable(int page = 1, int number = 8)
{
    var result :VoteCardModelHolder = voteModelBuilder.BuildAvailable(page, number);
    return result;
}
```

Рисунок 16 – Пример кода в контроллере

```
0+1 usages
public VoteCardModelHolder BuildAvailable(int page, int number)
{
    var votes :List<VoteEntity> = voteRepository.GetActiveAvailable();
    return BuildVoteCardModelHolder(votes.GetPaged(page, number).ToList(), votes.Count);
}
```

Рисунок 17 – Пример кода работы с базой данных

2. GET api/results/create

Метод для отправки ответа по голосованию. Содержит в себе форму ответа на голосование.

На рисунках 18, 19 приведены примеры использования кода.

```
/// <summary>
/// Создание ответа на голосование
/// </summary>
/// <param name="resultForm">Модель ответа пользователя</param>
/// <param name="latitude">Широта текущей позиции пользователя</param>
/// <param name="longitude">Долгота текущей позиции пользователя</param>
/// <returns></returns>
[HttpPost(template: "create")]
public IActionResult Create(ResultForm resultForm, double? latitude, double? longitude)
{
    resultModelHandler.Create(resultForm, latitude, longitude);
    if (resultForm.IsCompleted) analyticsProducer.Send(resultForm);
    return Ok();
}
```

Рисунок 18 – Пример кода в контроллере

```
public void Create(ResultForm resultForm, double? latitude, double? longitude)
{
    var currentUser = currentKeycloakUserProvider.Current;

    var vote = voteRepository.FindById(resultForm.VoteId);
    if (vote == null) throw new NotFoundException();

    if (vote.Position == PositionType.Geolocation)
    {
        var isNotInRegion = voteCheckHelper.IsNotInRegion(vote.Coordinates, latitude, longitude);
        if (isNotInRegion) throw new AisaeException(message: "Для прохождения голосования Вы должны находиться в указанной территории");
    }

    UserVoteMap userVoteMap;
    if (currentUser.Id.HasValue)
    {
        if (!userEntityRepository.IsExist(currentUser.Id.Value)) throw new AisaeException(message: "Пользователя не существует");

        userVoteMap = userVoteMapRepository.Find(userId: currentUser.Id.Value, resultForm.VoteId);
        if (userVoteMap == null)
        {
            userVoteMap = userVoteMapFactory.Create(currentUser.Id, resultForm.VoteId, resultForm.IsCompleted);
            writeDbContext.Add(userVoteMap);
        }
    }
    else
    {
        userVoteMap = userVoteMapFactory.Create(currentUser.Id, resultForm.VoteId, resultForm.IsCompleted);
        writeDbContext.Add(userVoteMap);
    }
}
```

Рисунок 19 – Пример кода в работе с базой данных

3. GET /api/results

Метод нужен для получения завершеного голосования по его уникальному идентификатору. В параметре принимает значение уникального идентификатора заявления

Пример использования кода приведен на рисунках 20, 21.

```
/// <summary>
/// Получение завершеного голосования по его уникальному идентификатору и ответов пользователя на него
/// </summary>
/// <param name="voteId">Уникальный идентификатор голосования</param>
/// <returns></returns>
[HttpGet(template: "")]
public ResultModel Get([Required] Guid voteId)
{
    var result = resultModelBuilder.Build(voteId);
    return result;
}
```

Рисунок 20 – Пример кода в контроллере

```
public ResultModel Build(Guid voteId)
{
    var currentUser = currentKeycloakUserProvider.Current;

    var vote = voteRepository.FindById(voteId);
    if (vote == null) throw new NotFoundException();

    var voteAnalytics = voteAnalyticsRepository.FindByVoteId(vote.Id);

    var voteResultModel = new VoteResultModel
    {
        Id = vote.Id,
        Title = vote.Title,
        Description = vote.Description,
        StartDateTime = vote.StartDateTime,
        EndDateTime = vote.EndDateTime,
        Type = vote.Type.ToString(),
        VotersNumber = voteAnalytics.VotersNumber,
        IsDigitalResult = vote.IsDigitalResult,
        Status = vote.GetStatus().ToString(),
        FileId = vote.FileDataId,
        Coordinates = vote.Coordinates != null
            ? JsonConvert.DeserializeObject<List<CoordinateModel>>(vote.Coordinates)
            : null,
        Questions = BuildQuestionModelList(vote.Id, vote.IsDigitalResult)
    };

    return new ResultModel
    {
        Vote = voteResultModel,
        UserAnswers = currentUser.Id.HasValue ? BuildUserAnswerModels(currentUser.Id.Value, vote.Id) : null
    };
}
```

Рисунок 21 – Пример кода работы с базой данных

4. GET api/votes/{id}

Метод для получения голосования по уникальному идентификатору. В параметре передается уникальный идентификатор голосования, а также координаты текущего местоположения пользователя для сортировки голосований по геолокации.

Пример использования кода приведен в рисунках 22, 23.

```
/// <summary>
/// Получение голосования по уникальному идентификатору
/// </summary>
/// <param name="id">Уникальный идентификатор</param>
/// <param name="latitude">Широта текущей позиции пользователя</param>
/// <param name="longitude">Долгота текущей позиции пользователя</param>
/// <returns></returns>
[HttpGet(template: "{id}")]
public VoteModelHolder Get(Guid id, double? latitude, double? longitude)
{
    var result :VoteModelHolder = voteModelBuilder.Build(id, latitude, longitude);
    return result;
}
```

Рисунок 22 – Пример кода в контроллере

```

public VoteModelHolder Build(Guid id, double? latitude, double? longitude)
{
    var vote = voteRepository.FindById(id);
    if (vote == null) throw new NotFoundException();

    var voteModel = BuildVoteModel(vote);

    if (vote.Position == PositionType.Geolocation)
        voteModel.IsNotInRegion = voteCheckHelper.IsNotInRegion(vote.Coordinates, latitude, longitude);

    var profile = keycloakProfileBuilder.Build();
    if (profile.Id != null)
    {
        voteModel.IsSuitableForIdp = voteCheckHelper.IsSuitableForIdp(vote.Idp, profile.Idp);
        voteModel.IsSuitableForAge = voteCheckHelper.IsSuitableForAge(vote.FromAge, vote.ToAge, profile.BirthDate);
        voteModel.IsSuitableForGender = voteCheckHelper.IsSuitableForGender(vote.Gender, profile.Gender);

        if (vote.Position == PositionType.Place)
        {
            switch (vote.Place)
            {
                case PlaceType.Registration:
                {
                    voteModel.IsSuitableForRegistrationAddress =
                        voteCheckHelper.IsSuitableForPlace(vote.Coordinates, profile.RegistrationAddress);
                    break;
                }
                case PlaceType.Living:
                {
                    voteModel.IsSuitableForLivingAddress =
                        voteCheckHelper.IsSuitableForPlace(vote.Coordinates, profile.LivingAddress);
                    break;
                }
            }
        }
    }
}

```

Рисунок 23 – Пример кода работы с базой данных.

3.2.3 API Новостей

API Новостей включает в себя методы и функции для работы с новостями.

API включает в себя следующие методы:

1. GET api/news

Метод служит для предоставления списка новостей упорядоченных от новых к поздним.

Пример кода представлен в Рисунке 24.

```

[HttpGet, ProducesJson]
public NewsGetListResponse GetList(NewsTypes? type, [Range(0, int.MaxValue)]int skip = 0, [Range(1, 100)]int take = 20)
{
    var query :IQueryable<News> = context.Get<DataLayer.Entities.News>()
        .ByType(type.HasValue, type)
        .ByState(NewsState.Published);

    var count :int = query.Count();

    var list = query //IQueryable<News>
        .OrderByDescending(t :News => t.Type.HasFlag(NewsTypes.Fixed))
        .ThenByDescending(t :News => t.Date) //IOrderedQueryable<News>
        .Select(t :News => new NewsGetListItemResult(
            t.Id,
            t.Title,
            t.Date, t.NewsFiles.Select(f :NewsFile => new NewsGetNewsFile(
                f.Id,
                f.Name
            )) //IEnumerable<NewsGetNewsFile>
        ))
        .Skip(skip > 0, skip)
        .Take(take) //IQueryable<NewsGetListItemResult>
        .ToList();

    return new NewsGetListResponse(
        count,
        list
    );
}

```

Рисунок 24 – Пример кода работы с базой данных

2. GET api/news/{id}

Метод служит для получение данных новости по уникальному идентификатору. В параметрах к запросу указывается уникальный идентификатор новости, а также параметры для отображения на клиентской части

Пример кода представлен на Рисунке 25.

```

/// <summary>
/// получение новости
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
[HttpGet(template: "{id}"), Produces(contentType: "application/json")]
public Task<NewsGetNewsItem> GetNews(Guid id)
{
    return context.Get<DataLayer.Entities.News>()
        .ById(id)
        .ByState(NewsState.Published) // IQueryable<News>
        .Select(t:News => new NewsGetNewsItem(
            t.Id,
            t.Title,
            t.Date,
            t.Content,
            t.NewsFiles.Select(f:NewsFile => new NewsGetNewsFile(
                f.Id,
                f.Name
            )) // IEnumerable<NewsGetNewsFile>
        )).FirstOrDefaultAsync(); // Task<NewsGetNewsItem>
}

```

Рисунок 25 – Пример кода работы с базой данных

3. POST api/news/create

Метод служит для создание новости. В теле запроса необходимо указать параметры данных новостей.

На Рисунке 26 приведен пример использования кода.

```

/// <summary>
/// создание новости
/// </summary>
/// <param name="form"></param>
/// <returns></returns>
[HttpPost, ProducesJson]
[ProducesResponseType(typeof(NewsSetNewsResponse), statusCode: 200)]
public IActionResult CreateNews(NewsCreateForm form)
{
    var news = context.SetNews(id: null, form);
    context.SaveChanges();

    return new JsonResult(new NewsSetNewsResponse(
        news.Id
    ));
}

```

Рисунок 26 – Пример кода работы с базой данных

4. POST api/news/{newsId}/file

Метод служит для прикрепления файлов по новости, указанной по уникальному идентификатору. В теле запроса необходимо указать уникальный идентификатор новости, а так же отправить файл, который реализует интерфейс `IFormFile`.

Пример использования кода представлен на Рисунке 27.

```

[HttpPost(template: "{newsId}/file"), ProducesJson]
[ProducesResponseType(typeof(NewsSetFileResponse), statusCode: 200)]
public async Task<IActionResult> SetFile(Guid newsId, [Required] IFormFile file)
{
    var news = await context.Get<News.DataLayer.Entities.News>()
        .ById(newsId) // IQueryable<News>
        .FirstOrDefaultAsync(); // Task<News>

    if (news is null)
    {
        return new NotFoundResult();
    }

    var newsFile = new NewsFile
    {
        Id = Guid.NewGuid(),
        NewsId = newsId,
        Name = file.FileName
    };
    context.Add(newsFile);

    await context.NewsFiles.SetAsync(newsFile.Id, file.OpenReadStream());

    await context.SaveChangesAsync();

    return new JsonResult(new NewsSetFileResponse(
        newsFile.Id
    ));
}

```

Рисунок 27 – Пример кода работы с базой данных

3.2.1 Пример работы с заявлениями

На рисунках 28 - 30 представлены примеры клиентской части работы с заявлениями.

 **АКТИВНЫЙ
ЕКАТЕРИНБУРГ**

[О ПРОЕКТЕ](#)

[ГОЛОСОВАНИЯ](#)

[НОВОСТИ](#)

[ЗАЯВЛЕНИЯ](#)

МЖ

Мурат Ж. 

ЛИЧНЫЙ КАБИНЕТ

[МОИ ДАННЫЕ](#)

[МОИ ЗАЯВЛЕНИЯ](#)

[НАСТРОЙКИ](#)

[УВЕДОМЛЕНИЯ](#)

НОВОЕ ЗАЯВЛЕНИЕ

КАТЕГОРИЯ

Содержание территории образовательной организации 

ПОДКАТЕГОРИИ

Уборка снега и наледи на территории образовательных организаций 

АДРЕС

улица Мира, 32 



Заполненность профиля

100%

Для доступа ко всем разделам сайта необходимо полностью заполнить данные профиля и подтвердить их.

ФИО, СНИЛС, д/р, пол	50%
Телефон	60%
E-mail	70%
Адреса регистрации	80%
Адрес проживания	90%
Разрешить геолокацию	100%

Редактировать данные

Рисунок 28 – Пример создания заявки на клиентской части.



АКТИВНЫЙ
ЕКАТЕРИНБУРГ

О ПРОЕКТЕ

ГОЛОСОВАНИЯ

НОВОСТИ

ЗАЯВЛЕНИЯ

МЖ

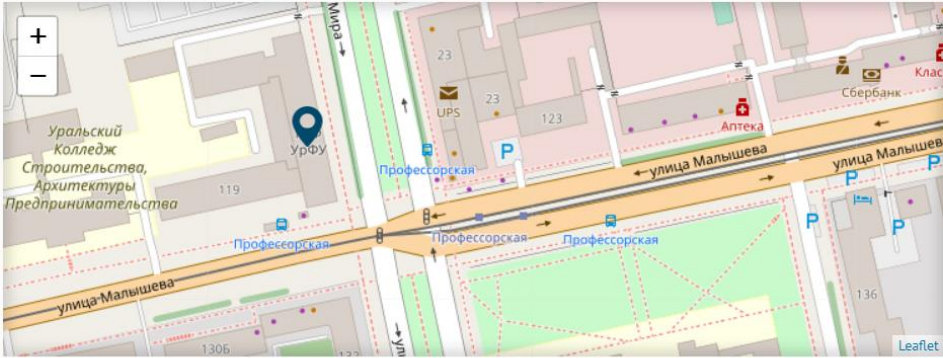
Мурат Ж.

✓

РЕДАКТИРОВАТЬ ДАННЫЕ

АДРЕС

улица Мира, 32



ПРИКРЕПИТЕ ФАЙЛЫ



Создать заявку

Отмена

Рисунок 29 – Пример создания заявки на клиентской части

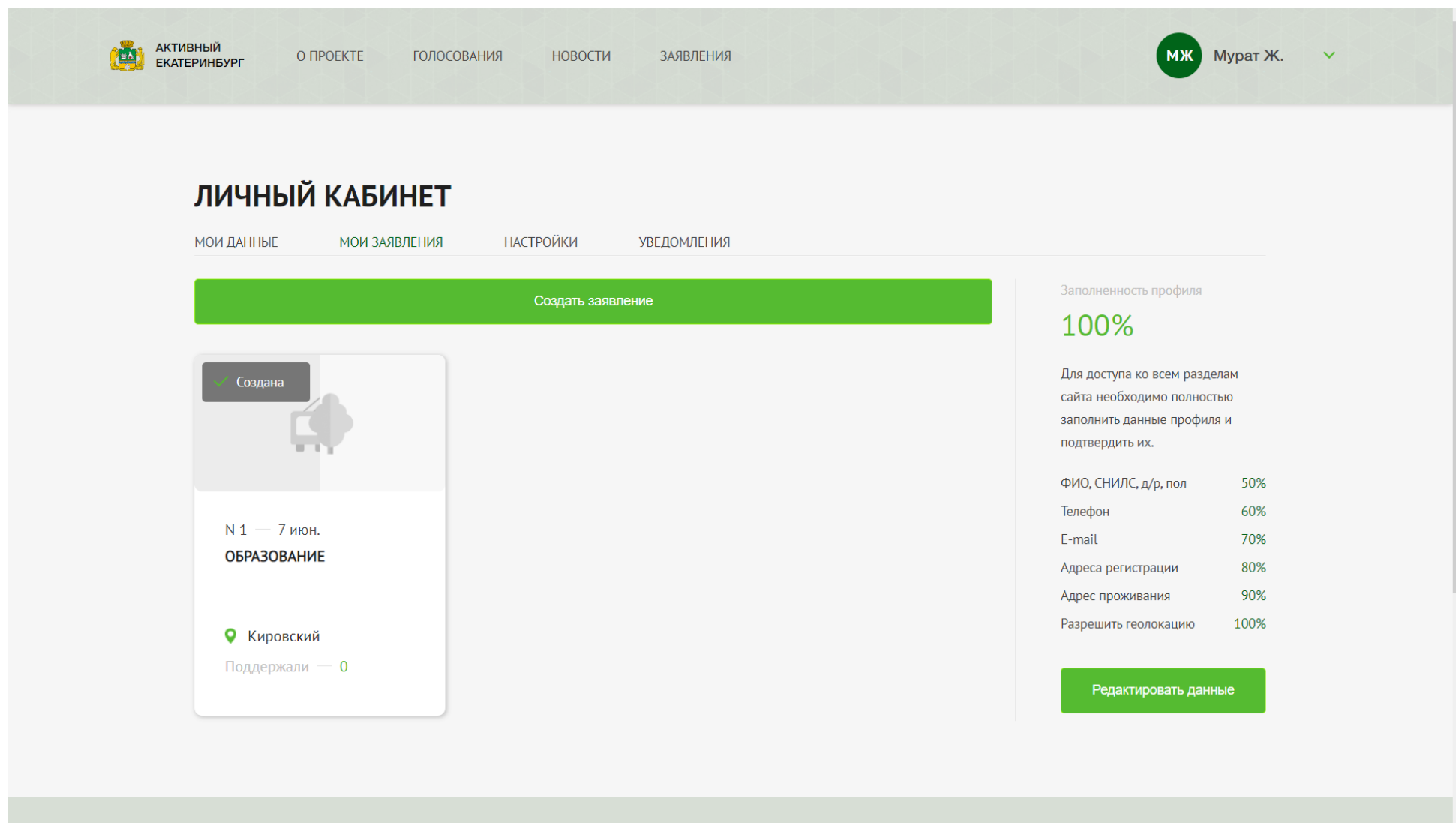


Рисунок 30 – Пример списка заявлений на клиентской части

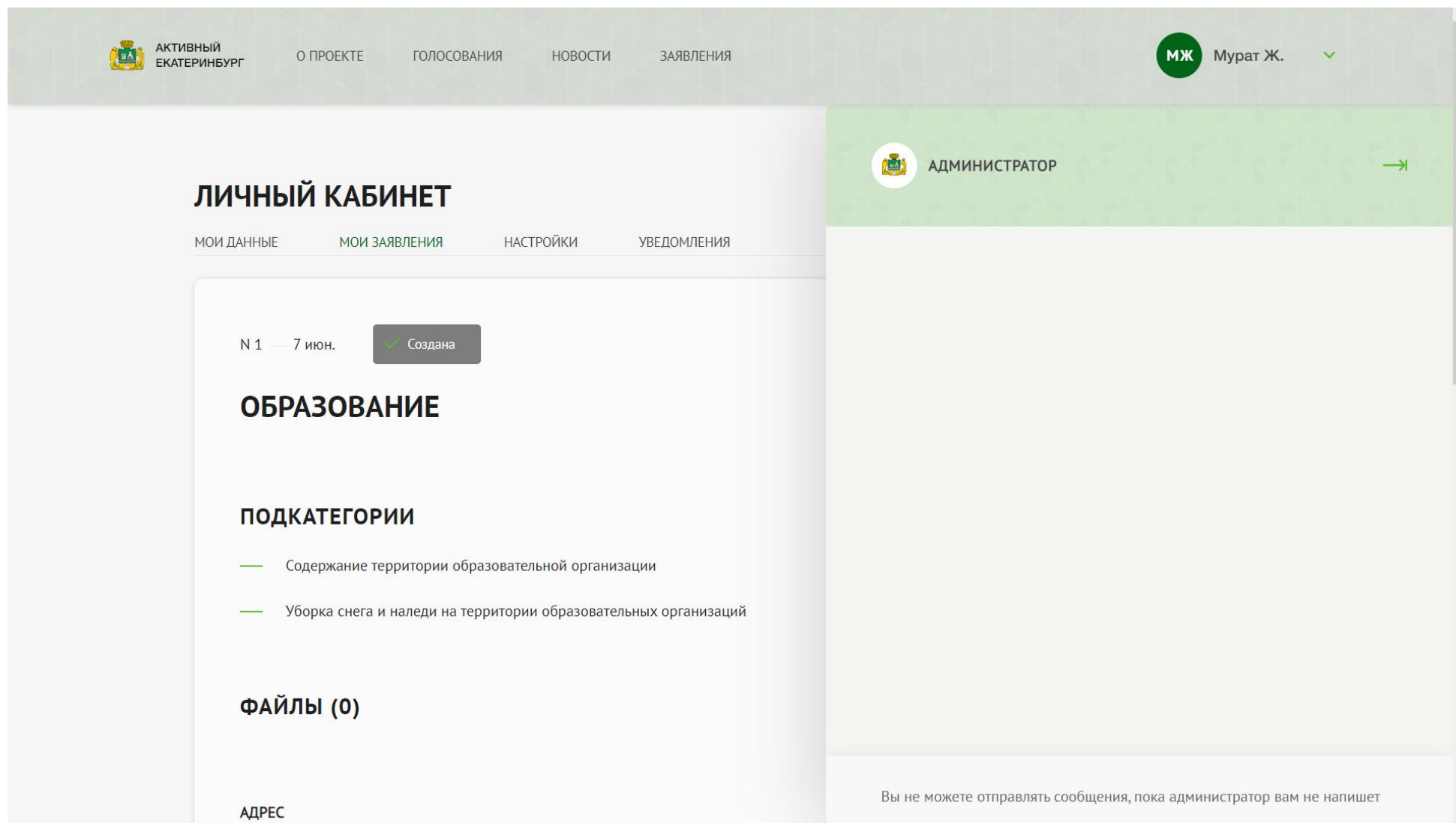


Рисунок 31 – Пример данных по заявке с открытым чатом с Администратором системы

3.3.1 Пример работы с голосованиями

На рисунках 32 - 36 представлены примеры клиентской части работы с голосованиями.

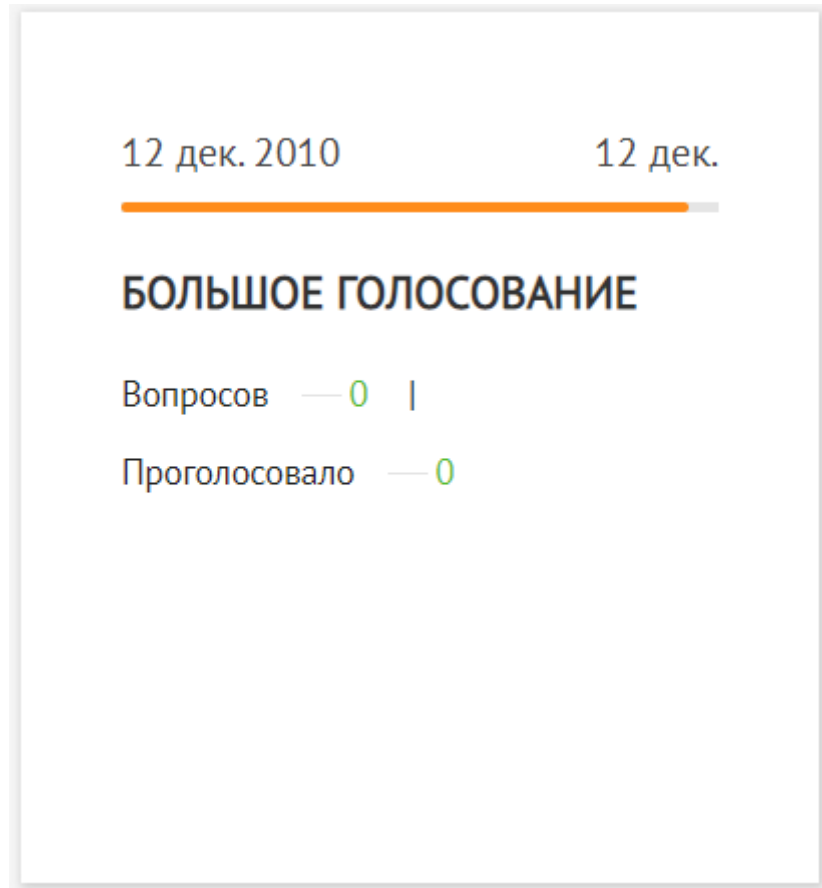


Рисунок 32 – Пример оформления карточки голосований

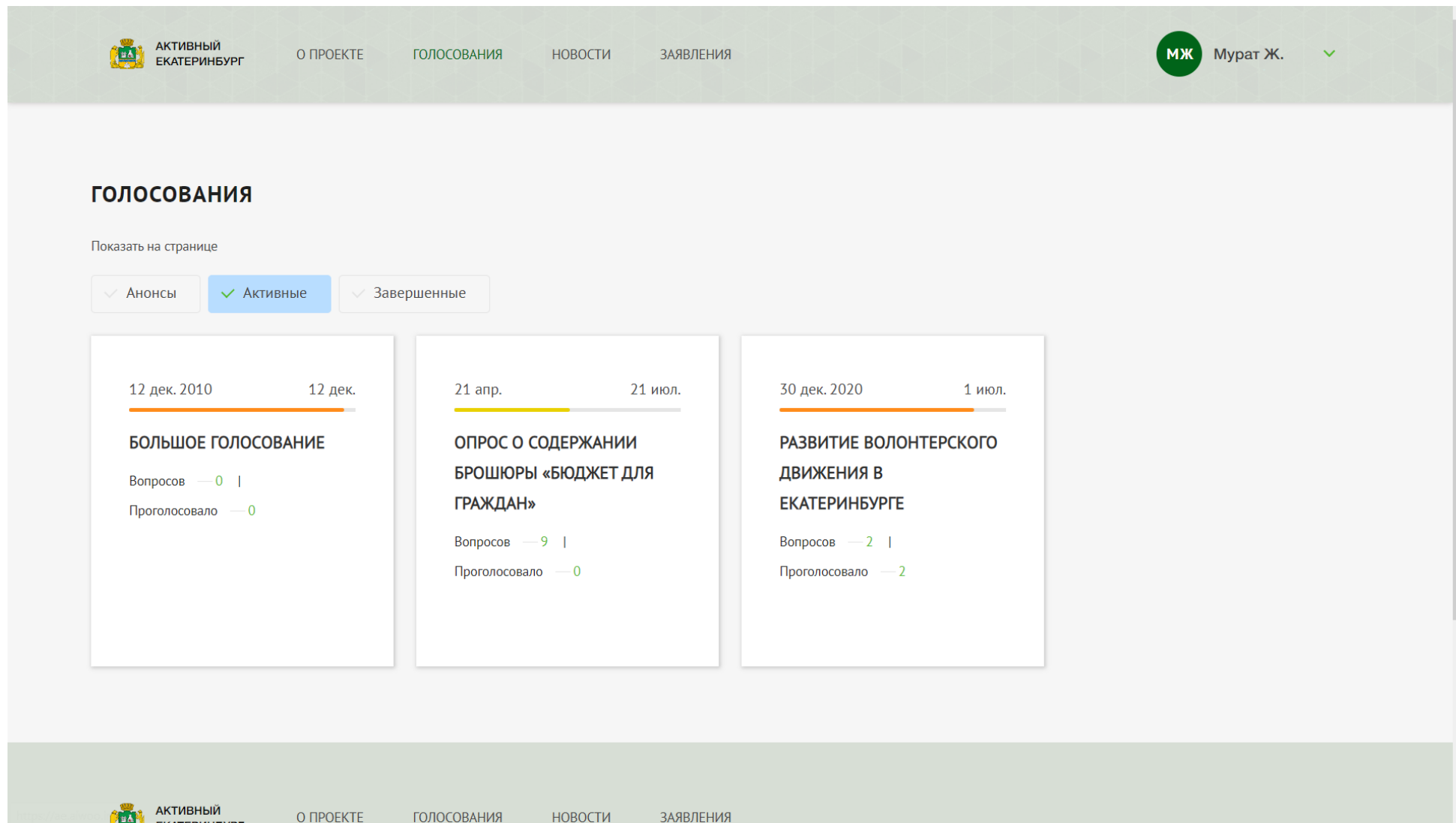


Рисунок 33 – Пример списка голосований

РАЗВИТИЕ ВОЛОНТЕРСКОГО ДВИЖЕНИЯ В ЕКАТЕРИНБУРГЕ

30 дек. 2020

1 июл.



Волонтерское движение в Екатеринбурге активно развивается. С каждым днем к нему присоединяется все больше неравнодушных жителей. При поддержке городских властей был создан ресурсный центр "Волонтер". На его сайте можно найти информацию о крупных городских событиях, волонтерских программах и акциях. Каждый желающий сможет выбрать направление волонтерской деятельности, которое ему наиболее интересно.

Рисунок 34 – Пример открытого голосования

 **АКТИВНЫЙ
ЕКАТЕРИНБУРГ**

[О ПРОЕКТЕ](#)

[ГОЛОСОВАНИЯ](#)

[НОВОСТИ](#)

[ЗАЯВЛЕНИЯ](#)

МЖ

Мурат Ж. 

2 КАКОЙ ИЗ ПЕРЕЧИСЛЕННЫХ КАНАЛОВ ТРАНСЛЯЦИИ ИНФОРМАЦИИ ПРЕДПОЧТИТЕЛЬНЕЕ ИМЕННО ДЛЯ ВАС ПО ТЕМАТИКЕ ВОЛОНТЕРСТВА?

ВАРИАНТЫ ОТВЕТОВ

Возможен выбор нескольких вариантов

Каналы в Telegram

Социальные сети (VK, FB, Instagram)

Печатные СМИ

Радио

Телевидение

Рассылка сообщений в мессенджерах WhatsApp или Viber

Посты в социальных сетях

Рисунок 35 – Пример вариантов ответа в открытом голосовании

ГОЛОСОВАНИЯ

[Показать на странице](#)

✓ Анонсы

Активные

✓ Завершенные



9 апр.

9 мая.

СТОИТ ЛИ ВЕРНУТЬ ПРЯМЫЕ ВЫБОРЫ МЕРА В Г. ЕКАТЕРИНБУРГ?

Голосование завершено

За последние недели в информационную повестку Екатеринбурга прочно вошла тема с народной инициативой по возврату...

1 апр.

1 ИЮН.

ЗНАЕТЕ ЛИ ВЫ, ЧТО 24
АПРЕЛЯ ОБЪЯВЛЕН
ОБЩЕГОРОДСКИМ ДНЁМ

Голосование завершено

Всемирный день чистоты – это идея, которая, несмотря на своё скромное начало, меняет мир...

5 мая.

21 мая.

THIS.EDITEDSTARTDATETIME

Голосование завершено

```
this.editedStartDateTime >
this.editedEndDateTime
```

9 апр.

9 мая.

СТОИТ ЛИ ВОЗВРАЩАТЬ ПРЯМЫЕ ВЫБОРЫ МЕДА В Г

12 дек. 2012

5 мая.

НАЗВАНИЕ

1 мая.

ЛУЧШИЙ РАЙОН
СВЯТОДМИТРОВСКА

30 ноя. 2020

1 мая.

КАК ВЫ СТАЛИ "АКТИВНЫМ ЕКАТЕРИНБУРЖИКОМ"?

Рисунок 36 – Пример списка завершённых голосований

3.3.1 Пример работы с новостями

На рисунках 37 - 38 представлены примеры клиентской части работы с новостями.

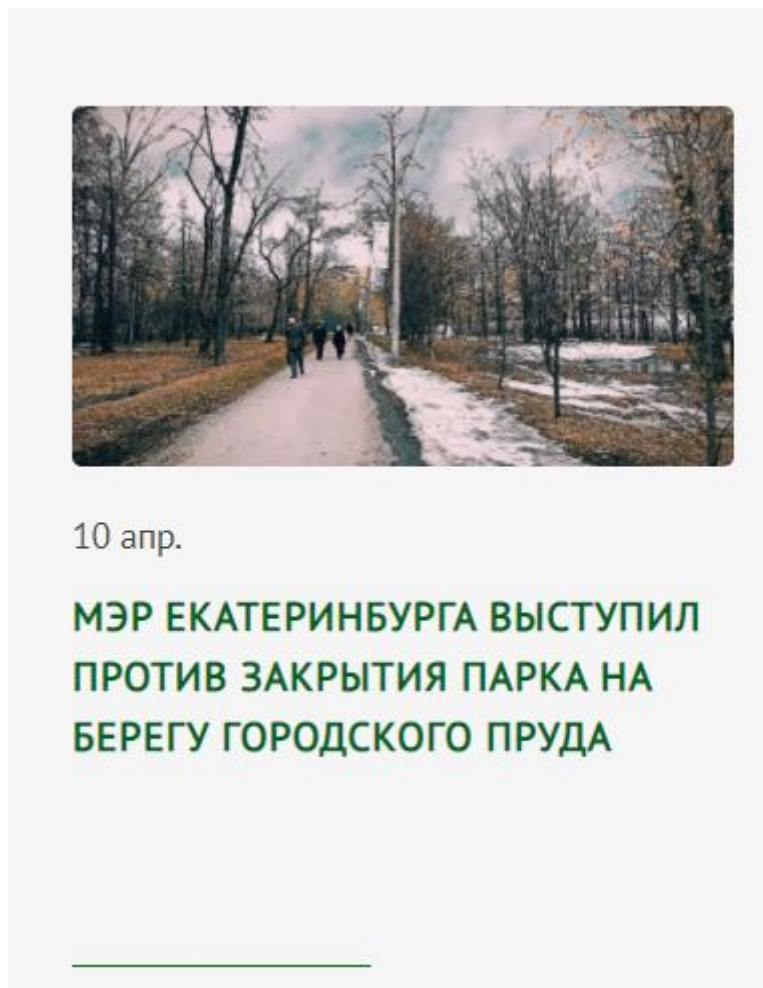


Рисунок 37 – Пример карточки новостей

89

3.3 Результаты и выводы

В этом разделе были рассмотрены инструменты разработки цифровой платформы «Активный Екатеринбург». Рассмотрев разные языки программирования и библиотеки для клиентской части, были выбраны оптимальные инструменты для разработки: C# и React. В качестве системы управления базами данных был выбран PostgreSQL. Приняв во внимание построенную ранее архитектуру, были выделены основные сущности баз данных и были разработаны интерфейсы для работы с ними. Для единой системы авторизации был внедрен инструмент Keycloak.

Далее были приведены примеры использования кода в основных API. Код писался с учетом принципов SOLID, не нарушал основные парадигмы объектно-ориентированного программирования.

Таким образом, были разработаны 3 системы пользования цифровой платформой «Активный Екатеринбург»: мобильное приложение, открытая и закрытая части.

ЗАКЛЮЧЕНИЕ

«Умный город» — концепция интеграции нескольких информационных и коммуникационных технологий (ИКТ) и Интернета вещей (IoT решения) для управления городским имуществом; активы города включают, в частности, местные отделы информационных систем, школы, библиотеки, транспорт, больницы, электростанции, системы водоснабжения и управления отходами, правоохранительные органы и другие общественные службы.

Чем концепция «Умный город» полезна для общества:

- Оптимизация работы транспортной системы. Нет опозданий, пробок, минимизация аварий на дороге, экономия расхода топлива или полная экологичность транспорта (электрокары).
- Энергоэффективность и экономия используемых ресурсов.
- Упрощение многих процессов. Оплата коммунальных счетов онлайн, сбор информации со счетчиков без вмешательства человека, быстрый поиск парковочного места и т.д.
- Повышение безопасности.
- Участие граждан в городской жизни и в принятии решений совместно с местными властями.
- Повышение комфорта и уровня жизни. «Умные» дома, беспилотные автомобили, управление многими процессами в доме, офисе одним нажатием кнопки на смартфоне.

Рынок умных городов растет не по дням, а по часам. Не является исключением и Россия, которая реализует национальную программу «Цифровая экономика» и проект «Умный город». В будущем цифровизация коснется всех городов РФ, но на данный момент утверждено 37 городов-пилотов, участвующих в проекте: Воронеж, Липецк, Самара, Великий Новгород, Челябинск, Томск и т.д. Для того чтобы стать пилотным умным городом, нужно было предоставить «дорожную карту» (или проект), где

прописана комплексная цифровизация городского хозяйства – от стандартных мероприятий до индивидуальных с учетом проблем того или иного региона.

Развитие концепции «Умный город» так же и не обошел город Екатеринбург. Приказом Министерства строительства и жилищно-коммунального хозяйства Российской Федерации от 31.10.2018 г. № 695 был утвержден ведомственный проект цифровизации городского хозяйства «Умный город» в рамках которого в соответствии с утверждёнными требованиям должна быть внедрена цифровая платформа вовлечения граждан в решения вопросов городского развития – «Активный Екатеринбург», разработка которого легла в основные цели данной диссертации.

Актуальность темы обусловлена потребностью администрации города Екатеринбург. Цифровая платформа «Активный Екатеринбург» решает задачи взаимодействия местной власти с гражданами, повышения эффективности решения проблем по городскому хозяйству, а также сокращения ресурсных затрат.

Предметами исследования выпускной квалификационной работы были создание сайтов, построению общей базы данных, разработка бизнес-процессов и архитектуры систем.

В диссертации был представлен обзор и сравнительный анализ концепции «Умный город», технических средств для разработки цифровой платформы, а именно рассмотрены общие понятия «Умных городов» и их преимуществ, а также разобраны примеры зарубежных и отечественных проектов, была представлена статистика и анализ языков программирования и выбран язык для разработки, а также произведен выбор других средств для реализации проекта.

В ходе выполнения дипломной работы был разработана цифровая платформа «Активный Екатеринбург», повышающая эффективность взаимодействия граждан и местных властей, разработанного на языке программирования C#.

Личный вклад автора заключается в выполнении основного объема теоретических и экспериментальных исследований, изложенных в диссертационной работе, включая разработку теоретических моделей, проведение исследований, анализ и оформление результатов. Также автор принимал прямое участие в разработке цифровой платформы и выполнял следующие задачи:

- организация работы команды проекта;
- разработка архитектур проекта и базы данных;
- разработка серверной части;
- разработка основных API и их документирование;
- поддержка мобильного приложения;
- развертывание цифровой платформы.

Подводя итоги выпускной квалификационной работы, можно сделать вывод, что поставленная цель исследования была достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Умный город [Электронный ресурс]// Wikipedia. - Режим доступа: <https://ru.wikipedia.org>
2. Умный город (Smart City) [Электронный ресурс]// Академический (Научно-технологический) класс. Режим доступа: <http://profil.mos.ru>
3. Что такое "умные" города и как они повышают качество жизни граждан? [Электронный ресурс]// informБЮРО. - Режим доступа: <https://informburo.kz>
4. Bright Julian The smart city: Communications utopia or future reality? / Telecommunications, - 9 ABI/INFORM Global, 1995, p.175
5. Интеллектуальные города [Электронный ресурс]// Tadviser Государство. Бизнес. ИТ. - Режим доступа: <https://www.tadviser.ru>
6. Международный форум "Современные городские технологии - "Smart Cities" [Электронный ресурс]// Информационный бизнес-портал Нижнего Новгорода "По'Нижнему" - Режим доступа: - <http://po-nn.ru>
7. Sanoff H. 2000 Community Participation Methods in Design and Planning. John Wiley & Sons. 306 p.
8. Итоги электронных референдумов, инициированных Департаментом торговли и услуг в проекте "Активный гражданин" в мае-августе 2014 года. - Режим доступа: <http://www.slideshare.net/DITMoscow/>
9. Итоги голосования "Городские мероприятия - Онлайн-программа мероприятий на ВДНХ"[Электронный ресурс]// "Активный гражданин" - Режим доступа: <https://ag.mos.ru/results/3545>
10. Портал "Наш Санкт-Петербург"[Электронный ресурс]//. "О проекте" - Режим доступа: <https://gorod.gov.spb.ru/about/>
11. "Вместо жалобной книги. Объясняем, что такое портал «Наш Санкт-Петербург» и как им пользоваться"[Электронный ресурс]// Новости Санкт-Петербурга "Фонтанка.ру" - Режим доступа: <https://www.fontanka.ru/2021/04/21/69876572/>

12. Swimlanes [Электронный ресурс]// Grapholite - Режим доступа: <https://grapholite.ru/swimlanes/>
13. "Создание архитектуры программы или как проектировать табуретку" [Электронный ресурс] // Хабр - Режим доступа: <https://habr.com/ru/post/276593/>
14. "Учебное пособие по диаграмме прецедентов" [Электронный ресурс] // creately - Режим доступа: <https://creately.com/blog/ru/диаграмма/учебное-пособие-по-диаграмме-прецедентов>
15. "Архитектура веба: основы для начинающих разработчиков" [Электронный ресурс] // tproger - Режим доступа: <https://tproger.ru/translations/web-architecture-101/>
16. "Библиотека React: особенности, перспективы, ситуация на рынке труда" [Электронный ресурс] // Хекслет - Режим доступа: <https://ru.hexlet.io/blog/posts/biblioteka-react-review-article>
17. "React и Redux. Функциональная веб-разработка".[Текст]/ Бэнкс Алекс, Порселло Ева, 2018. - 336 с.
18. "React - JavaScript-библиотека для создания пользовательских интерфейсов" [Электронный ресурс] // React - Режим доступа: <https://ru.reactjs.org/>
19. "What is JSX?" [Электронный ресурс]// W3Schools - Режим доступа: https://www.w3schools.com/react/react_jsx.asp
20. "Документация по C#" [Электронный ресурс]// Microsoft Docs - Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
21. "C# — Преимущества и недостатки" [Электронный ресурс]// Code blog - Режим доступа: <https://shwanoff.ru/plus-minus-c-sharp/>
22. "История создания языка программирования Java" [Электронный ресурс]// Интерфейс - Режим доступа: <http://www.interface.ru/home.asp?artId=1549>

23. "Плюсы и минусы программирования на Java" [Электронный ресурс]
// medium - Режим доступа: <https://medium.com/nuances-of-programming/плюсы-и-минусы-программирования-на-java-2861f4c2a0d5>
24. "What is PostgreSQL?" [Электронный ресурс]// PostgreSQL - Режим
доступа: <https://www.postgresql.org/about/>
25. "PostgreSQL — объектно-реляционная система управления базами
данных" [Электронный ресурс] // Web Creator - Режим доступа: <https://web-creator.ru/articles/postgresql>
26. "Функции на языке запросов (SQL)" [Электронный ресурс]//
PostgresPro - Режим доступа: <https://postgrespro.ru/docs/postgresql/9.4/xfunc-sql>
27. "Пример использования триггеров в PostgreSQL" [Электронный
ресурс]// Записки программиста - Режим доступа: <https://eax.me/postgresql-triggers/>
28. "Правила PostgreSQL" [Электронный ресурс]// PostgresPro - Режим
доступа: <https://postgrespro.ru/docs/postgresql/9.6/sql-createrule>
29. "Типы индексов" [Электронный ресурс]// PostgresPro - Режим
доступа: <https://postgrespro.ru/docs/postgresql/9.6/indexes-types>
30. "Архитектура PostgreSQL. Изоляция и многоверсионность"
[Электронный ресурс]// Егор Рогов, Павел Лузанов, 2017 - Режим доступа:
https://edu.postgrespro.ru/dba1/dba1_05_arch_mvcc.pdf
31. "Расширение" [Электронный ресурс]// PostgresPro - Режим доступа:
<https://postgrespro.ru/docs/postgresql/9.6/sql-createextension>
32. "Наследование" [Электронный ресурс]// PostgresPro - Режим доступа:
<https://postgrespro.ru/docs/postgrespro/9.5/ddl-inherit>
33. "SSO на микросервисной архитектуре. Используем Keycloak. Часть
№1" [Электронный ресурс] Хабр - Режим доступа:
<https://habr.com/ru/company/X5RetailGroup/blog/486778/>
34. Keycloak [Электронный ресурс]// Wikipedia. - Режим доступа:
<https://ru.wikipedia.org/wiki/Keycloak>

35. "Keycloak - Open Source Identity and Access Management" [Электронный ресурс]// Keycloak Docs - Режим доступа: <https://www.keycloak.org/documentation>
36. Стандарт JSON Web Token [Электронный ресурс]// Standards Track - Режим доступа: <https://datatracker.ietf.org/doc/html/rfc7519>
37. "Access Tokens" [Электронный ресурс]// Auth0 Docs - Режим доступа: <https://auth0.com/docs/tokens/access-tokens>
38. "Refresh Tokens" [Электронный ресурс]// Auth0 Docs - Режим доступа: <https://auth0.com/docs/tokens/refresh-tokens>
39. "Безопасность JSON Web Tokens (JWT)" [Электронный ресурс]// Cyber Polygon - Режим доступа: <https://cyberpolygon.com/ru/materials/security-of-json-web-tokens-jwt/>
40. "Лекция 7. Сервис–ориентированная архитектура (SOA)" [Электронный ресурс]// Архитектура информационных систем - Режим доступа: <https://sites.google.com/site/arhitekturainformacionnyhsist/lekcia-7-servis-orientirovanna-arhitektura-soa>
41. "Использование MS Project для управления проектами" [Электронный ресурс]// Хабр – Режим доступа: <https://habr.com/post/151593/>
42. "Бизнес-процессы" [Электронный ресурс]. Школа бизнеса "Управляй будущим" – Режим доступа: <https://www.u-b-s.ru/publikacii/biznes-processy.html>
43. Развитие моделирования бизнес-процессов [Электронный ресурс]. – Режим доступа: <https://www.scienceforum.ru/2015/802/15636>
44. Нотация BPMN 2.0: ключевые элементы и описание [Электронный ресурс]. – Режим доступа: <https://www.comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/>
45. Алгоритм [Электронный ресурс]. – Режим доступа: <http://www.machinelearning.ru/wiki/index.php?title=Алгоритм>

46. ГОСТ 19.701-90 ЕСПД Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/gost-19-701-90-espd>
47. Понятие языка программирования [Электронный ресурс]. – Режим доступа: <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya>
48. Интернет технологии [Электронный ресурс]. – URL: <https://www.internet-technologies.ru/articles/php-ruby-python>
49. Язык C# и платформа .NET Core [Электронный ресурс] Metanit.com Сайт о программировании - Режим доступа: <https://metanit.com/sharp/tutorial/1.1.php>
50. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с
51. Справочник по html и css [Электронный ресурс] – Режим доступа: <http://htmlbook.ru/>
52. Мартин Роберт К. «Чистый код. Создание, анализ и рефакторинг» - Питер, 2010. – 464 с
53. Агуров, Павел С#. Сборник рецептов / Павел Агуров. - М.: "БХВ-Петербург", 2012. - 432 с.
54. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 с.
- Информационные технологии. Основы работы с реляционной БД Oracle. - М.: McGraw-Hill, 2002. - 200 с.
55. Информационные технологии. Основы работы с реляционной БД Oracle. - М.: McGraw-Hill, 2002. - 200 с.
56. Мюллер, Р.Дж. Базы данных и UML. Проектирование / Р.Дж. Мюллер. - М.: ЛОРИ, 2002. - 420 с.
57. Modern Web Development with ASP.NET Core 3: An End to End Guide Covering the Latest Features of Visual Studio 2019, Blazor and Entity Framework, 2nd Edition, 2020. - 802p
58. Practical Entity Framework [Текст]// Gorman, Brian, 2020 - 649p

59. React Native Learn once, write anywhere. [Электронный ресурс]// -
Режим доступа: <https://reactnative.dev/>

60. Introduction to Expo. [Электронный ресурс]// Режим доступа:
<https://docs.expo.io/>

61. Кроссплатформенная разработка мобильных приложений в 2020
году [Электронный ресурс]// Режим доступа: <https://habr.com/ru/post/491926/>