

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

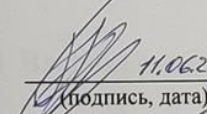
Специальность: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»
Специализация: «Эксплуатация вычислительных машин, комплексов,
систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

ЭКОЛОГИЧЕСКАЯ КАРТА ПЕНЗЕНСКОЙ ОБЛАСТИ

Студент


(подпись, дата)

Прокопданов Валерий Владимирович
(ФИО полностью)

Руководитель


(подпись, дата)

11.06.2024

Слепцов Н.В.
(фамилия, инициалы)

Нормоконтролёр


(подпись, дата)

11.06.2024

Бычков А.С.
(фамилия, инициалы)

Рецензент

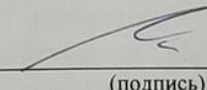
21. специалист
АО «ИПП РЗД»
(должность, место работы)


(подпись, дата)

Васильков А.В.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 27.05.2024 г. № 11)

Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой отлично

(протокол заседания ГЭК от 19.06.2024 г. № 2)

Секретарь ГЭК



Е.И. Гурин

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«19» апреля 2024 г.

**ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ**

1. Студент Прокоданов Валерий Владимирович гр. 19ВОЭ1 факультета вычислительной техники

2. Тема работы «Экологическая карта Пензенской области»

Тема утверждена приказом ПГУ № 405/о от "19" апреля 2024 г.

3. Руководитель работы к.т.н., доцент каф. ВТ «ПГУ» Слепцов Н.В.

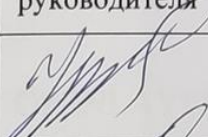
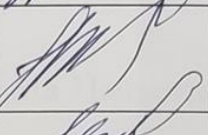
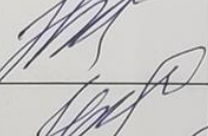
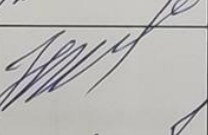
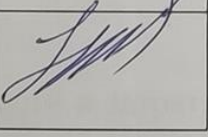
4. Задание на работу (назначение разработки, исходные данные и т.п.).

Разработать Web-приложение для мониторинга экологической обстановки. Основными функциями Web-приложения являются регистрация пользователей, добавление обнаруженных загрязнений, отображение обнаруженных загрязнений и их состояний, изменение информации об обнаруженных загрязнениях (если необходимо).

5. Перечень подлежащих разработке вопросов.

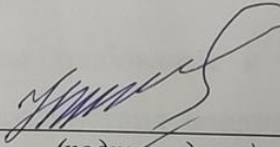
Требуется произвести анализ предметной области, разработать диаграмму развертывания, диаграмму прецедентов, диаграммы последовательностей, диаграмму состояний, схему данных. На основе которых, разработать Web-приложение «Экологическая карта Пензенской области». Также необходимо разработать экономический раздел и раздел БЖД. Оформить техническую документацию на дипломный проект.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10%	19.04.24 – 24.04.24	
2.	Выбор инструментов разработки	10%	25.04.24 – 29.04.24	
3.	Разработка UML-диаграмм	10%	30.05.24 – 07.05.24	
4.	Реализация базы данных	10%	06.05.2024 – 13.05.2024	
5.	Реализация Web-приложения	35%	14.05.24 – 31.05.24	
6.	Развёртывание приложения и отладка	15%	01.05.2024- 03.06.2024	
7.	Оформление выпускной работы	10%	04.06.2024- 07.06.2024	

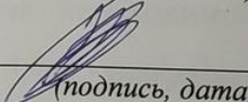
Дата выдачи задания "19" апреля 2024г.

Руководитель дипломного проекта


(подпись, дата)

Слепцов Н.В.

Задание к исполнению принял студент


(подпись, дата)

Прокоданов В.В.

Работу к защите допустить

Декан факультета



(подпись, дата)

Фионова Л.Р.

17.06.2024

Реферат

Работа содержит 118 листов, 21 рисунка, 12 таблиц, 7 источников, 3 приложений

ЭКОЛОГИЯ, ГИС, WEB-ПРИЛОЖЕНИЕ, JAVASCRIPT, PHP, OPENLAYERS, VISUAL STUDIO CODE, OPEN SERVER, MYSQL

Цель ВКР – разработка Web-приложения экологическая карта Пензенской области.

В ходе выполнения ВКР было разработано Web-приложение «Экологическая карта Пензенской области» в соответствии с техническим заданием.

В результате ВКР было разработано клиент-серверное Web-приложение, соответствующее техническому заданию, а именно:

- разработана СУБД, хранящая информацию о маркерах;
- разработано Web-приложение, которое обрабатывает информацию о маркерах.

В экономическом разделе был произведен расчет трудоемкости и затрат на разработку программного обеспечения.

В разделе БЖД были описаны требования к офисным помещениям и рабочему месту для минимизации воздействия вредных факторов на организм человека, а также был произведен расчет необходимой освещенности помещения.

Графическая часть содержит UML-диаграммы, описывающие СУБД и реализованное Web-приложение.

ПГУ 1.090501.09.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Прокопданов В.В.		11.06.24
Провер.		Слепцов Н.В.		11.06.24
Н. контр.		Бычков А.С.		11.06.24
Утв.		Митрохин М.А.		11.06.24
Экологическая карта Пензенской области Пояснительная записка				
Лит.		Лист	Листов	
		2	118	
ФВТ, каф. ВТ гр.19В031				

Содержание

Перечень сокращений и обозначений.....	5
Введение.....	6
1 Анализ предметной области.....	8
1.1 Актуальность.....	8
1.2 Обзор аналогов.....	9
1.3 Постановка задачи.....	12
1.3.1 Требования к функциональным характеристикам приложения.....	13
1.3.2 Требования к устройствам пользователей.....	14
1.3.3 Стадии и этапы разработки.....	14
2 Выбор инструментов разработки.....	16
2.1 Выбор языка программирования.....	16
2.2 Выбор среды разработки	16
2.3 Выбор сервиса для отображения карт.....	18
2.4 Выбор сервиса облачной СУБД.....	20
2.5 Выбор локальной платформы для запуска проекта.....	21
3 Архитектура приложения.....	23
3.1 Выбор архитектуры приложения.....	23
3.2 Разработка UML–диаграмм.....	24
4 Разработка приложения.....	28
4.1 Проектирование базы данных.....	28
4.2 Система аутентификации пользователя.....	30
4.2.1 Страница авторизации.....	30
4.2.2 Страница регистрации.....	32
4.3 Работа с картами.....	33
4.3.1 Отображение маркеров.....	35
4.3.2 Добавление маркеров на карту.....	39
4.4 Личный кабинет пользователя.....	40
4.5 Страницы с дополнительной информацией.....	41

5 Технико–экономическое обоснование.....	43
6 Безопасность жизнедеятельности.....	49
6.1 Потенциально опасные факторы при работе с ПЭВМ.....	49
6.2 Классификация помещений.....	50
6.3 Размещение оборудования и организация рабочих мест.....	51
6.4 Освещение.....	51
6.5 Расчет освещенности рабочего места.....	55
Заключение.....	58
Список используемых источников.....	59
Приложение А – UML-диаграммы.....	61
Приложение Б – Листинг программы.....	67
Приложение В – Презентация.....	111

Перечень сокращений и обозначений

В данной работе приняты следующие сокращения и обозначения

БД – база данных

ГИС – геоинформационная система

НК – налоговый кодекс

ОМС – обязательное медицинское страхование

ОПС – обязательное пенсионное страхование

ОРСУБД – объектно–реляционная система управления базами данных

ОСС – обязательное социальное страхование

ПО – программное обеспечение

ПФР – пенсионный фонд России

ПЭВМ – персональная электронно–вычислительная машина

РСУБД – реляционная система управления базами данных

РФ – российская федерация

СУБД – система управления базами данных

ФЗ – федеральный закон

ФОМС – федеральный фонд обязательного медицинского страхования

PDO – (PHP data object) – объект данных PHP

PHP – (Hypertext Preprocessor) – препроцессор гипертекста

UID – (User identifier) – идентификатор пользователя

UML – (Unified Modeling Language) – унифицированный язык моделирования

Введение

Геоинформационные системы привычны для использования людьми в различных сферах жизнедеятельности. Самыми популярными являются: построение маршрутов, отслеживание транспорта, просмотр погоды. Реальное применение ГИС гораздо шире, а именно:

– управление земельными ресурсами, земельные кадастры. Для решения проблем, имеющих пространственную привязку, и начали создавать ГИС. Типичные задачи — составление кадастров, классификационных карт, определение площадей участков и границ между ними и т. д.;

– инвентаризация, учет, планирование размещения объектов распределенной производственной инфраструктуры и управление ими. Например, нефтегазодобывающие компании или компании, управляющие энергетической сетью, системой бензоколонок, магазинов и т. п.;

– проектирование, инженерные изыскания, планировка в строительстве, архитектуре. Такие ГИС позволяют решать полный комплекс задач по развитию территории, оптимизации инфраструктуры строящегося района, требующегося количества техники, сил и средств;

– тематическое картографирование;

– управление наземным, воздушным и водным транспортом. ГИС позволяет решать задачи управления движущимися объектами при условии выполнения заданной системы отношений между ними и неподвижными объектами. В любой момент можно узнать, где находится транспортное средство, рассчитать загрузку, оптимальную траекторию движения, время прибытия и т. п.;

– управление природными ресурсами, природоохранная деятельность и экология. ГИС помогает определить текущее состояние и запасы наблюдаемых ресурсов, моделирует процессы в природной среде, осуществляет экологический мониторинг местности;

– геология, минерально–сырьевые ресурсы, горнодобывающая промышленность. ГИС осуществляет расчеты запасов полезных ископаемых по

результатам проб (разведочное бурение, пробные шурфы) при известной модели процесса образования месторождения;

– чрезвычайные ситуации. С помощью ГИС производится прогнозирование чрезвычайных ситуаций (пожаров, наводнений, землетрясений, селей, ураганов), расчет степени потенциальной опасности и принятие решений об оказании помощи, расчет требуемого количества сил и средств для ликвидации чрезвычайных ситуаций, расчет оптимальных маршрутов движения к месту бедствия, оценка нанесенного ущерба;

– военное дело. Решение широкого круга специфических задач, связанных с расчетом зон видимости, оптимальных маршрутов движения по пересеченной местности с учетом противодействия и т. п.;

– сельское хозяйство. Прогнозирование урожайности и увеличения производства сельскохозяйственной продукции, оптимизация ее транспортировки и сбыта [1].

В данной работе ГИС использованы для экологического мониторинга местности, а именно Пензенской области. Помимо этой сферы, систему можно использовать и для других целей, например, переориентировать под задачи военного дела или использовать для мониторинга других, не менее важных аспектов: чрезвычайных ситуаций, транспорта, минерально–сырьевых ресурсов и т.д. Так же функциональную часть данного проекта можно расширить добавив в него наблюдение за состоянием охранных зон, отслеживание состояния водных объектов

Для того, чтобы показать основной функционал геоинформационной системы, была выбрана именно тема экологии субъекта Российской Федерации, потому что тема экологии является довольно важной для рассмотрения, ведь от показателей экологии напрямую зависит качество жизни самого человека.

В настоящее время экологическая обстановка Пензенской области находится на неплохом уровне, а введение в эксплуатацию данной системы позволит

повысить рейтинг экологической обстановки и закрепиться в числе самых экологически благоприятных для жизни городов и регионов России.

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ предметной области

1.1 Актуальность

У ГИС существует множество определений, но если в них выделить общую часть, то: ГИС – это интерактивная информационная система, обеспечивающая сбор, хранение, доступ, отображение пространственно–организованных данных и ориентированная на возможность принятия научно–обоснованных управленческих решений [2].

По состоянию на 2024 год Россия находится на 112 месте из 180 в списке стран с лучшей экологией.

Для улучшения экологической обстановки в стране, в частности в Пензенской области, было принято решение создать экологическую ГИС, в которой была бы возможность отслеживать состояние природы в нашем субъекте Российской Федерации.

Экологическая ГИС — это совокупность двух методов: ГИС и дистанционного зондирования (наблюдение поверхности Земли наземными, авиационными и космическими средствами).

ГИС, в частности, экологической направленности, может быть использована людьми в различных сферах деятельности, таких как: мониторинг полезных ископаемых, природоохранных зон и мероприятий, регулирование использования земельных участков и отслеживание информации, связанной со стихийными бедствиями. Примером использования экологической ГИС может служить наблюдение за состоянием водных объектов с целью защиты от загрязнений восстановление их экосистем (при необходимости). Подобные действия также могут быть применены к земляным и воздушным участкам, будь то несанкционированная свалка, которая может привести к такой проблеме как эрозия почвы, или выбросы производствами вредных веществ, превышающие допустимые значения.

Для специалистов – экологов, ученых и простых людей экологические ГИС являются важным инструментом, предоставляющим уникальные возможности для изучения, сохранения и поддержания экологической обстановки окружающей среды. Безусловно, это невозможно без точных географических данных, получаемых с помощью метода дистанционного зондирования.

1.2 Обзор аналогов

ГИС экологической направленности довольно редки, но при глубоком внедрении в жизнь человека, они вносят свой вклад в экологию и помогают улучшить качество жизни людей. Например, в Республике Татарстан внедрена ГИС «Экологическая карта Республики Татарстан». Экологическая политика Казани получила признание как в России, так и за ее пределами. На 2024 год Казань входит в топ 3 экологически чистых городов России. «Экологическая карта Республики Татарстан» внесла немалый вклад в улучшение экологической обстановки Татарстана.

В Кирове тоже есть ГИС экологической направленности, но она гораздо хуже внедрена в жизнь человека. Как следствие не может дать таких же высоких результатов как у Татарстана.

На рисунке 1.1 приведен внешний вид экологической карты Кировской области. На карте отображается точное местоположение загрязнений с цветовой индикацией о состоянии решения проблемы. Поставив метку, также имеется возможность добавить подробное описание проблемы с фотоподтверждением.

Также на карте имеются еще три пункта: «Пробы», «Земля и имущество», «Охотнадзор», данные пункты не обладают никакими функциями, так как при их выборе, на карте не происходило изменений.

Данная ГИС плохо интегрирована в человеческую жизнь, так как проблемные места, отмеченные на карте, не изменяются уже долгое время и отсутствует информация о решении проблемы (куда передана информация о проблеме, когда передана информация о проблеме, статус решения проблемы).

Также существенным минусом является то, что данная ГИС действует только на территории Кировской области.

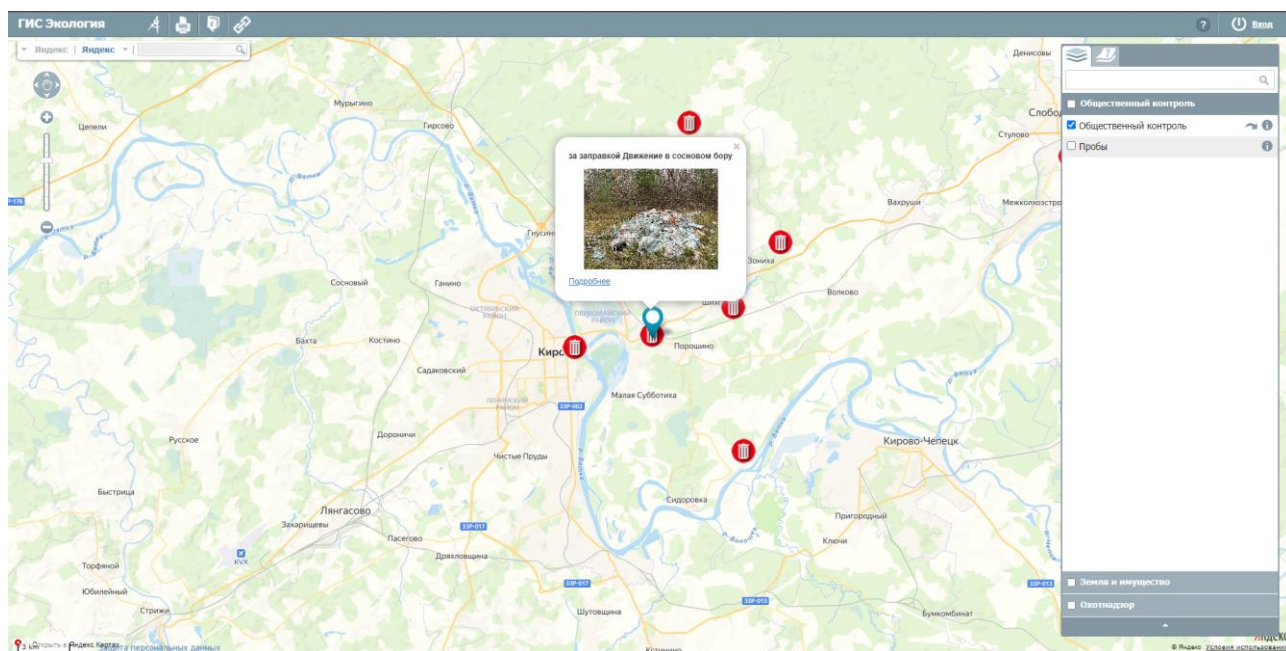


Рисунок 1.1 — Экологическая карта Кировской области

На рисунке 1.2 приведен внешний вид экологической карты Республик Татарстан. На этой карте отображается местоположение загрязнений, в том числе затонувших средств передвижения, с состоянием решения проблемы. У каждой экологической проблемы имеется подробное описание (вид отходов; тип несанкционированного размещения отходов; дата обнаружения; дата проверки; объем и площадь загрязнения; меры, предпринятые для ликвидации; органы, куда данная информация была передана).

Также на карте имеются еще несколько пунктов:

- лицензионные и согласованные участки для отслеживания территорий, где проходит добыча ископаемых и законно ли это;
- места скоплений полезных ископаемых (недропользование);
- особо охраняемые природные зоны;

Из минусов данной экологической ГИС можно выделить плохую оптимизацию (при переходе от слоя к слою происходит кратковременное зависание карты) и территориальное действие (только в пределах Республики Татарстан).

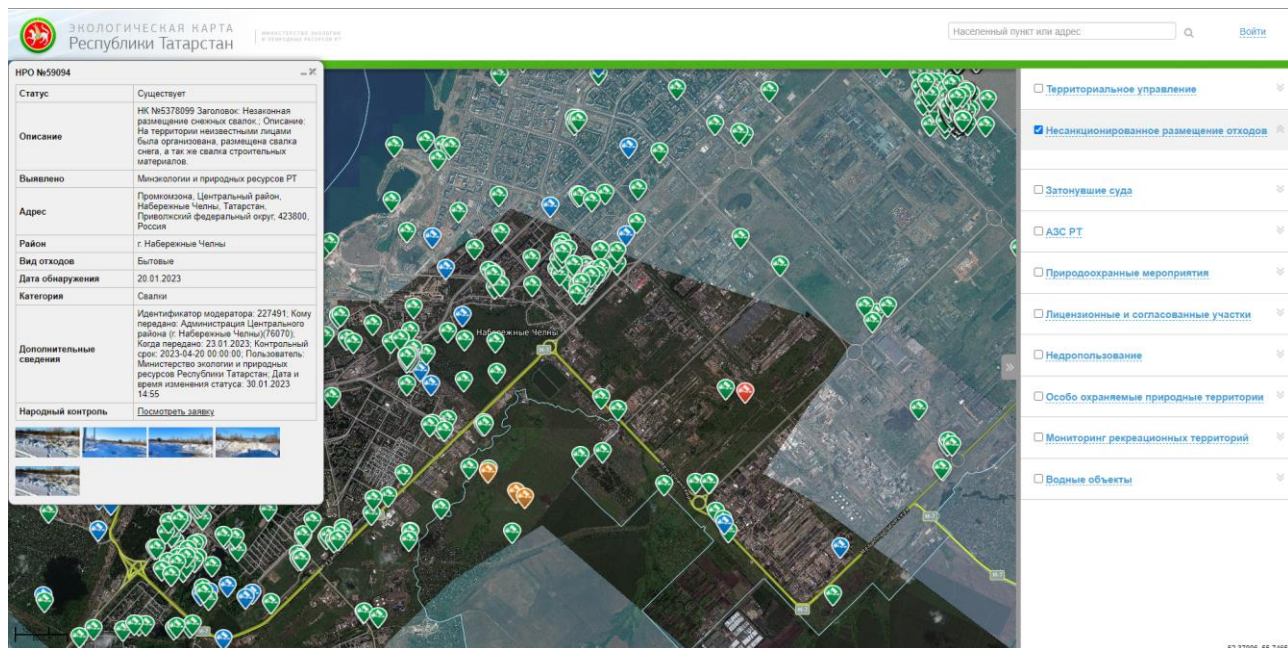


Рисунок 1.2 — Экологическая карта Республики Татарстан

Результат сравнительного анализа экологических карт приведен в таблице 1.1.

Таблица 1.1 — Результат сравнительного анализа

Функции	Название	
	Экологическая карта Кировской области	Экологическая карта Республики Татарстан
Отслеживание несанкционированных свалок	+	+
Подробное описание проблемы	—	+
Отслеживание состояния воды	—	+
Отслеживание особо охраняемые территории	—	+
Оптимизация	—	—

Функции	Название	
	Экологическая карта Кировской области	Экологическая карта Республики Татарстан
Действие в Пензенской области	—	—

Исходя из результатов сравнительного анализа можно сделать вывод о том, что разрабатываемое приложение выполняло похожий функционал, но должно быть более оптимизированным и действовать на территории Пензенской области.

1.3 Постановка задачи

1.3.1 Требования к функциональным характеристикам приложения

Главной целью данного проекта является разработка экологической карты Пензенской области, которая позволит улучшить состояние эко–среды в области.

Анализ потребностей целевых пользователей разрабатываемого приложения показал, что для их удовлетворения оно должно обладать следующими функциональными особенностями:

- возможностью регистрации и авторизации пользователей на сайте;
- возможностью фильтрации меток на карте;
- возможностью добавления меток на карту;
- возможностью добавления описания на метку;
- возможностью просмотра уже имеющихся меток на карте;
- возможностью редактирования меток;
- возможностью просмотра собственных меток в личном кабинете.

Входными данными для системы являются:

- информация о пользователе сайта (имя, email, пароль);
- информация о маркерах (координаты местоположения, описание маркера, состояние маркера).

1.3.2 Требования к устройствам пользователей

Для использования Web–приложения устройство пользователя должно обладать следующими параметрами:

- иметь постоянное подключение к сети Интернет;
- иметь экран диагональю не менее 14”;
- иметь одну из следующих операционных систем:
 - 1) Windows 8.1 и выше;
 - 2) macOS 10.10.5 и выше;
 - 3) Ubuntu 18.04 и выше;
- должен быть установлен один из следующих браузеров:
 - 1) Chrome;
 - 2) Яндекс;
 - 3) Firefox;
 - 4) Safari;
 - 5) Edge.

1.3.3 Стадии и этапы разработки

Стадии и этапы разработки представлены в таблице 1.2

Таблица 1.2 – Стадии и этапы разработки

Наименование этапов работы	Объем работы, %	Срок выполнения
Анализ предметной области	10	19.04.2024– 24.04.2024
Выбор инструментов для разработки программы	10	25.04.2024– 30.04.2024
Проектирование и составление UML диаграмм для реализации программы	10	01.05.2024– 06.05.2024
Реализация базы данных	10	07.05.2024– 13.05.2024
Реализация программы	35	14.05.2024– 28.05.2024

Наименование этапов работы	Объем работы, %	Срок выполнения
Развёртывание приложения и отладка	15	29.05.2024– 01.06.2024
Оформление выпускной работы	10	02.06.2024– 05.06.2024

2 Выбор инструментов разработки

2.1 Выбор языка программирования

В настоящее время невозможно представить себе разработку Web приложений без PHP и JavaScript. Каждый из языков обладает своими преимуществами и недостатками, которые повлияли на выбор языка для разработки:

- оба языка бесплатны в распространении;
- оба языка являются кроссплатформенными;
- JavaScript обладает мощным функционалом, в отличие от PHP (быстрая компиляция и сборка проекта, что очень важно при разработке проекта);
- у JavaScript широкая поддержка и сообщество разработчиков;
- с помощью JavaScript возможно создать сложное Web–приложение, что невозможно на PHP;
- PHP незаменим в работе с базами данных.

Учитывая преимущества, был решено использовать оба языка программирования.

2.2 Выбор среды разработки

На данный момент в мире информационных технологий существует огромное количество различных средств разработки приложений на языке JavaScript.

На сегодняшний день самыми востребованными являются:

- IntelliJ IDEA;
- Visual Studio Code;
- PyCharm.

Каждая среда разработки обладает уникальными характеристикам, о которых подробно ниже.

IntelliJ IDEA – это интегрированная среда разработки, заточенная под Java, которая направлена на сокращение времени разработки путем оптимизации повторяющихся задач.

К достоинствам IntelliJ IDEA можно отнести:

- интуитивно понятный пользовательский интерфейс;
- удобное авто дополнение.

К недостаткам IntelliJ IDEA можно отнести:

- бесплатная версия не поддерживает JavaScript;
- довольно дорогостоящая;
- потребляет значительные ресурсы компьютера.

Visual Studio Code представляет собой высоко–оптимизированный редактор кода от Microsoft, который при первом запуске позволяет пользователю настроить среду под себя, установив все необходимые для разработки расширения.

К достоинствам Visual Studio Code можно отнести:

- стоимость (бесплатно);
- низкое потребление памяти;
- расширяемость и настраиваемость.

К недостаткам Visual Studio Code можно отнести:

- ограниченный контроль исходного кода;
- требует времени для настройки с использованием расширений и плагинов.

PyCharm – это среда разработки, предназначенная для разработчиков на Python, но также может подойти и разработчикам, пишущим на JavaScript.

К достоинствам PyCharm можно отнести:

- хорошие возможности отладки;
- работа с кодом на удаленной машине и его отладка;
- работа с базами данных.

К недостаткам PyCharm можно отнести:

- платный;
- основное направление – Python;

– иногда возникают проблемы при отладке.

Проанализировав достоинства и недостатки различных сред разработки, поддерживающих язык JavaScript, выбор был остановлен на среде разработки Visual Studio Code.

2.3 Выбор сервиса для отображения карт

Для разработки приложения с картами необходимо использовать API. Существует несколько популярных API, которые позволяют создавать приложения, неотъемлемой частью которых являются карты.

Самыми популярными в настоящее время являются:

- Mapbox.
- Leafletjs;
- OpenLayers.

Mapbox – это открытая библиотека, позволяющая создавать карты как для Web-приложений, так и для обычных приложений она располагает простыми в использовании инструментами и дополнительными сервисами такими как: геокодирование, спутниковое изображение.

К достоинствам Mapbox можно отнести:

- быстрая загрузка карт;
- открытый исходный код.

К недостаткам Mapbox можно отнести:

- платный полный функционал;
- недостаточно оптимизирован для работы с большими картами.

Leaflet – это библиотека для создания интерактивных карт написанная на языке JavaScript. Имеет небольшой размер исходного файла, в котором присутствует все необходимое для создания онлайн-карт.

К достоинствам Leaflet можно отнести:

- хорошая документация;
- маленький размер;

– широкая поддержка.

К недостаткам Leaflet можно отнести:

- узкий круг разработчиков;
- исходная документация содержит только базовые примеры;
- поддерживаются только растровые карты.

OpenLayers – это открытая библиотека для создания интерактивных карт, предназначенная для работы с JavaScript. Отличительной особенностью OpenLayers является то, что можно выбрать сервер, чьи карты будут использоваться.

К достоинствам OpenLayers можно отнести:

- высокая производительность;
- хорошая документация;
- не требует дополнительных настроек и сразу готов к работе;
- мощная и гибкая;
- поддержка векторных карт.

К недостаткам OpenLayers можно отнести:

- большой объем;
- отсутствие систем геокодирования и маршрутизации.

Официальный сайт OpenLayers предоставляет большое количество примеров, дающих представление о работе библиотеки, о возможности настройки отображения карты с помощью CSS файла.

OpenLayers также подходит для рендеринга векторных данных из GeoJSON, TopoJSON, KML, GML и других географических форматов данных.

Проанализировав достоинства и недостатки различных средств для отображения карт, поддерживающих язык JavaScript, выбор был остановлен на библиотеке OpenLayers.

2.4 Выбор сервиса СУБД

Для хранения информации необходимо использовать базу данных. Существует несколько популярных БД, обеспечивающие хранение данных.

На сегодняшний день самыми востребованными являются:

- Microsoft SQL server;
- MySQL;
- PostgreSQL.

Microsoft SQL server – продукт разработанный компаниями Microsoft и Sybase.

К достоинствам Microsoft SQL server можно отнести:

- можно использовать практически на любой конфигурации железа;
- интеграция с другими решениями от Microsoft;
- позволяет хранить страницы размером до 8КБ.

К недостаткам Microsoft SQL server можно отнести:

- стоимость;
- закрытый исходный код;
- работа системы только на Windows.

MySQL – одна из популярнейших БД среди Web-приложений. Разработанная в 1994 году компанией MySQL.

К достоинствам MySQL можно отнести:

- производительность;
- кроссплатформенность;
- бесплатное распространение;
- открытый исходный код.

К недостаткам MySQL можно отнести:

- низкая производительность с большими БД;
- ограниченный функционал в сравнении с другими СУБД.

PostgreSQL – объектно-реляционная СУБД, разработанная в 1996 году Майклом Стоунбрейкером.

К достоинствам PostgreSQL можно отнести:

- производительность;
- кроссплатформенность;
- бесплатное распространение;
- открытый исходный код.

К недостаткам PostgreSQL можно отнести:

- сложность в настройке;
- высокое потребление ресурсов;
- высокие требования к аппаратным ресурсам.

Проанализировав достоинства и недостатки различных СУБД, выбор был остановлен на РСУБД MySQL.

2.5 Выбор локальной платформы для запуска проекта

Для разработки и демонстрации проекта необходимо иметь среду, позволяющую локально запустить приложение в формате Web.

На сегодняшний день самыми востребованными являются:

- XAMPP;
- Open Server;
- Denwer.

XAMPP – популярный кроссплатформенный серверный дистрибутив, включающий Apache, MariaDB, PHP, Perl. Прост в настройке и в установке. Оснащен удобной панелью управления для настройки сервера. У XAMPP'а большое сообщество пользователей подходит как для начинающих, так и для опытных разработчиков.

Open Server – это серверная среда на базе ОС Windows, включающая Apache, NGINX, MariaDB, PHP, MySQL, PhpMyAdmin. Он включает в себя удобную панель управления и предлагает широкий спектр параметров настройки. Open Server – хороший вариант для разработчиков, которым нужна гибкая и настраиваемая среда.

Denwer – русскоязычная серверная среда, разработанная специально для российских пользователей. Включающий Apache, PHP, MySQL, PhpMyAdmin и другие компоненты. Обладает дружественным пользовательским интерфейсом. Так же Denwer предоставляет документацию и поддержку на русском языке.

Проанализировав достоинства и недостатки различных локальных серверов, выбор был остановлен на Open Server.

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

3 Архитектура приложения

3.1 Выбор архитектуры приложения

В зависимости от логики распределения клиентской и серверной сторон, Web–приложения можно разделить по типам архитектуры:

- одностраничные;
- многостраничные;
- микросервисные.

Одностраничное приложение — это Web–приложение, главной особенностью которого является подгрузка всей требующейся информации при входе на страницу. Положительным моментом в использовании одностраничных приложений является то, что не происходит перезагрузки страниц, в связи с чем ускоряется работа с сайтом.

Многостраничное приложение – это Web–приложение, которое, в отличие от одностраничного приложения, использующего одну страницу, на которую подгружается вся требующаяся пользователю информация, имеет набор Web–страниц, соединенных между собой ссылками и вызывающих друг друга. Многостраничные приложения для взаимодействия с сервером перезагружают страницу. Пока страница перезагружается данные либо записываются на сервер, либо считываются с него.

Архитектура микросервисов предназначена для создания Web–приложений из отдельных сервисов, то есть одно приложение состоит из подприложений, а изменения, влияющие на какое–то отдельно взятое подприложение, никак не сказываются на остальных подприложениях.

Исходя из рассмотренного выше краткого описания архитектур Web–приложений был сделан выбор в пользу многостраничной архитектуры, так как она в полной мере позволит реализовать задуманное Web–приложение.

3.2 Разработка UML–диаграмм

Разрабатываемое Web–приложение состоит из Web–приложения пользователя и сервера базы данных. Пользователь взаимодействует с сервером базы данных для добавления и использования информации о метках. Диаграмма развертывания отображает взаимодействие клиента и сервера (рисунок 3.1).

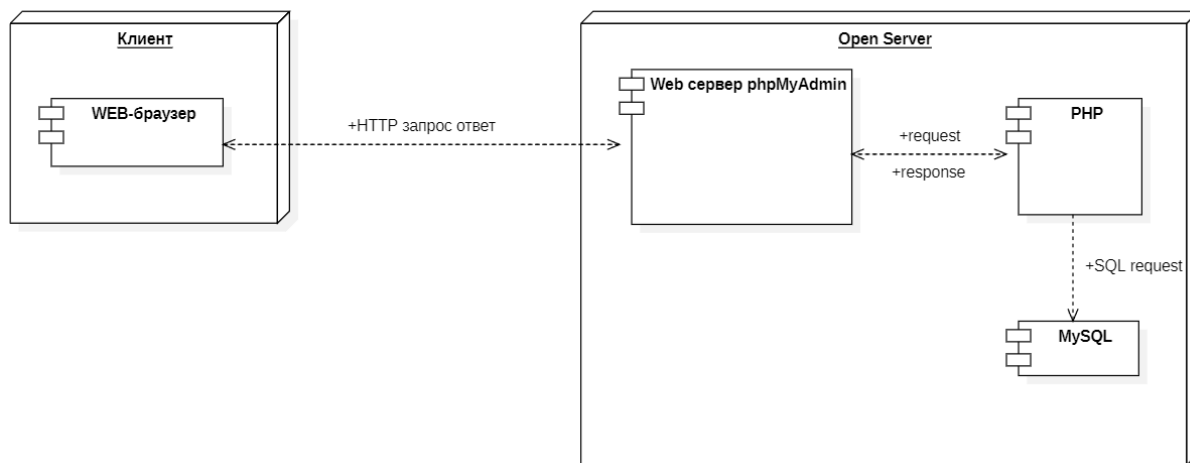


Рисунок 3.1 – Диаграмма развертывания

Чтобы представить функциональную часть приложения была разработана диаграмма прецедентов (рисунок 3.2).

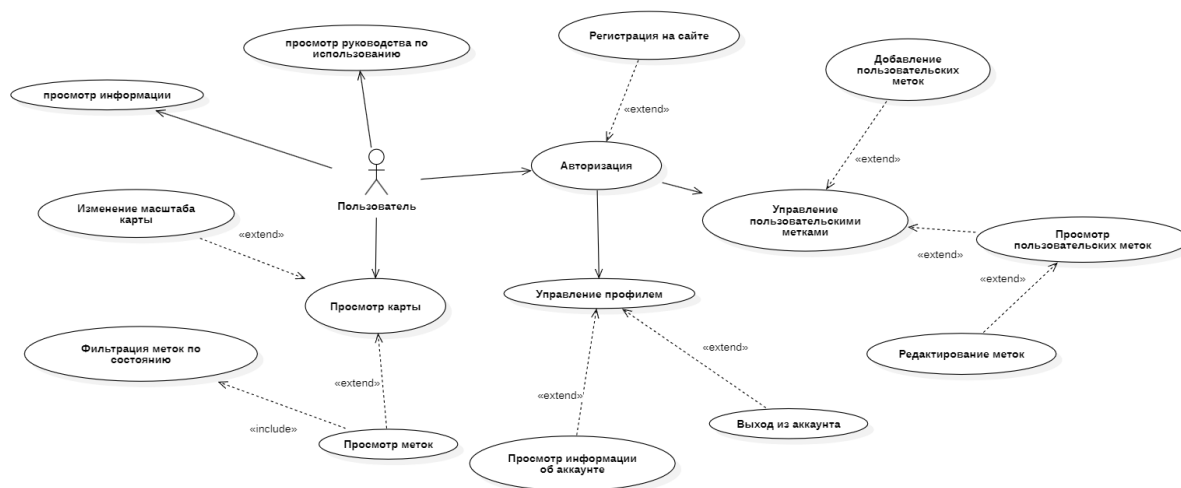


Рисунок 3.2 – Диаграмма прецедентов

Разработанная диаграмма прецедентов включает в себя: один актор (Пользователь) и 15 вариантов использования (Регистрация, Авторизация,

Управление профилем, Просмотр информации об аккаунте, Выход из аккаунта, Управление пользовательскими метками, Добавление пользовательских меток, Просмотр пользовательских меток, Редактирование меток, Просмотр карты, Просмотр меток, Фильтр меток по состоянию, Изменение масштаба карты, Просмотр информации, Просмотр руководства по использованию).

При запуске программы пользователь попадает на страницу с картой, где пользователь может фильтровать метки, просмотреть инструкцию и подробнее узнать о приложении. При нажатии на иконку профиля, пользователь попадает на страницу авторизации (если пользователь не был авторизован ранее). На этой странице необходимо авторизоваться или, если нет учетной записи, нажать на кнопку регистрации и зарегистрироваться. После авторизации (регистрации) пользователь попадает на страницу, где можно фильтровать метки, добавлять метки и, при необходимости, из аккаунта.

Для отображения функционала пользователя были разработаны диаграммы последовательностей (рисунки 3.3, 3.4).

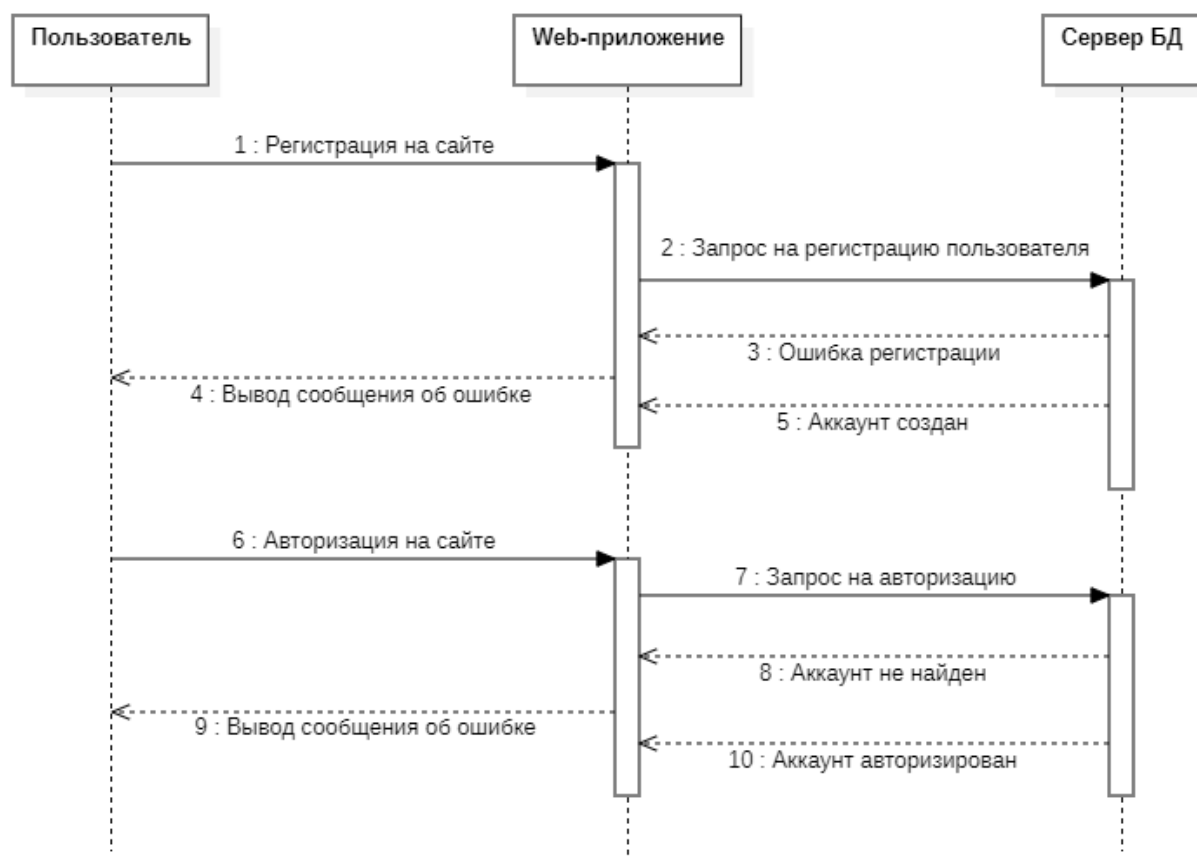


Рисунок 3.3 – Диаграмма последовательностей «Авторизация»



Рисунок 3.4 – Диаграмма последовательностей «Создание маркера»

Для отражения состояния приложения при взаимодействии пользователя с ним была разработана диаграмма состояний (рисунок 3.5).



Рисунок 3.5 – Диаграмма состояний

Для создания базы данных была разработана диаграмма данных (рисунок 3.6).

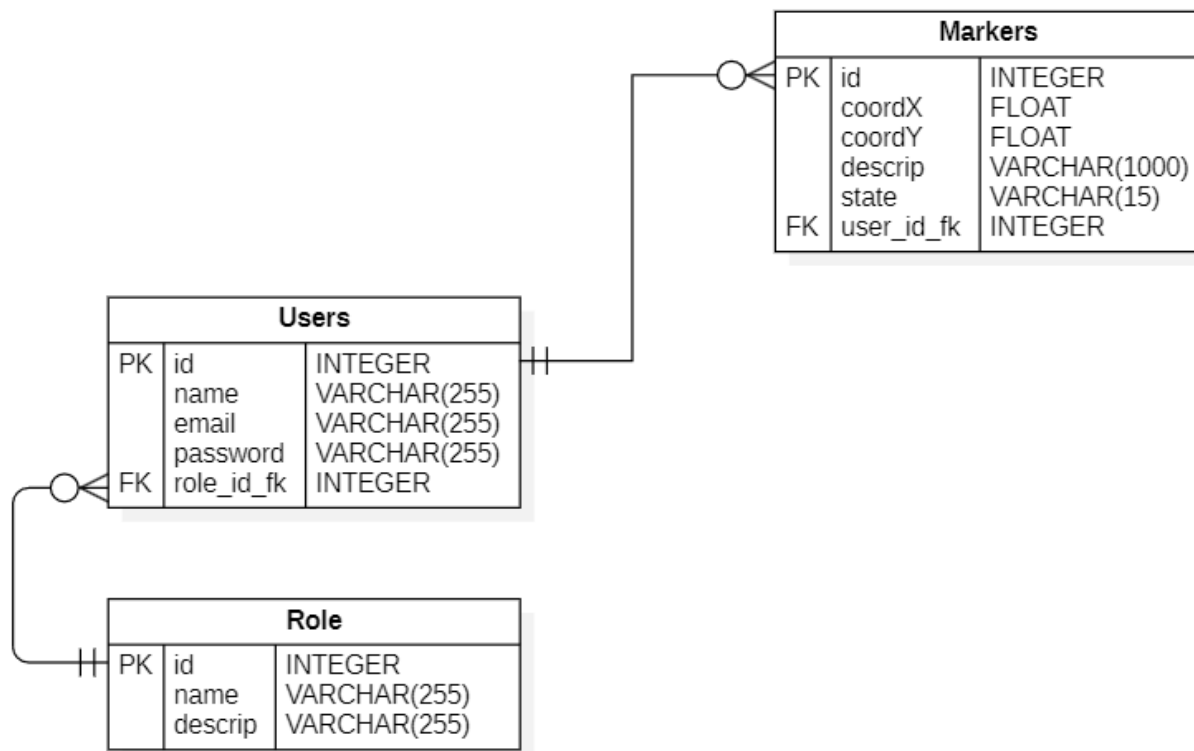


Рисунок 3.6 – Диграмма данных

Модель базы данных содержит три сущности. Таблица Users содержит информацию о пользователях, а именно: id – UID пользователя типа INTEGER, name – имя пользователя типа VARCHAR, email – электронная почта типа VARCHAR, password – зашифрованный пароль типа VARCHAR, role_id_fk – UID роли, которой обладает пользователь типа INTEGER. Таблица Role содержит информацию о ролях пользователей, а именно: id – UID роли типа INTEGER, name – название роли типа VARCHAR, descrip – описание роли типа VARCHAR. Таблица Markers содержит информацию о метках на карте, а именно: id – UID маркера типа INTEGER, coordX – широта местоположения маркера типа FLOAT, coordY – долгота местоположения маркера типа FLOAT, user_id_fk – UID пользователя, добавившего маркер.

4 Разработка приложения

4.1 Проектирование базы данных

Для задуманного функционирования сайта необходимо реализовать БД, состоящую из трех таблиц.

Описание полей таблиц БД приведено в таблицах 4.1 – 4.3.

Таблица 4.1 – Описание полей таблицы «Roles»

Имя атрибута	Нулевое значение	Ключевое поле	Тип
id	Недопустимо	PK	INTEGER
name	Недопустимо	–	VARCHAR(255)
descrip	Недопустимо	–	VARCHAR(255)

Для того, чтобы создать таблицу Roles в phpMyAdmin необходимо выполнить следующий код:

```
CREATE TABLE IF NOT EXISTS Roles (  
id int NOT NULL AUTO_INCREMENT,  
name varchar(255) NOT NULL,  
descrip varchar(255) NOT NULL,  
PRIMARY KEY (id),  
) ENGINE=InnoDB DEFAULT AUTO_INCREMENT=1;
```

Таблица 4.2 – Описание полей таблицы «Users»

Имя атрибута	Нулевое значение	Ключевое поле	Тип
id	Недопустимо	PK	INTEGER
name	Недопустимо	–	VARCHAR(255)
email	Недопустимо	–	VARCHAR(255)
password	Недопустимо	–	VARCHAR(255)
role_id_fk	Недопустимо	FK	INTEGER

Для того, чтобы создать таблицу Users в phpMyAdmin необходимо выполнить следующий код:

```
CREATE TABLE IF NOT EXISTS Users (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(255) NOT NULL,
  email varchar(255) NOT NULL,
  password varchar(255) NOT NULL,
  role_id_fk int NOT NULL,
  PRIMARY KEY (id),
  KEY role_id_fk (role_id_fk)
) ENGINE=InnoDB DEFAULT AUTO_INCREMENT=1;
```

Таблица 4.3 – Описание полей таблицы «Markers»

Имя атрибута	Нулевое значение	Ключевое поле	Тип
id	Недопустимо	PK	INTEGER
coordX	Недопустимо	–	FLOAT
coordY	Недопустимо	–	FLOAT
descrip	Недопустимо	–	VARCHAR(1000)
state	Недопустимо	–	VARCHAR(15)
user_id_fk	Недопустимо	FK	INTEGER

Для того, чтобы создать таблицу Markers в phpMyAdmin необходимо выполнить следующий код:

```
CREATE TABLE IF NOT EXISTS Markers (
  id int NOT NULL AUTO_INCREMENT,
  coordX float NOT NULL,
  coordY float NOT NULL,
  descrip varchar(1000) NOT NULL,
  state varchar(15) NOT NULL,
  user_id_fk int NOT NULL,
  PRIMARY KEY (id),
  KEY user_id_fk (user_id_fk)
) ENGINE=InnoDB DEFAULT AUTO_INCREMENT=1;
```

4.2 Система аутентификации пользователей

Для создания системы аутентификации были использованы:

- база данных MySQL;
- PHP;
- PDO;
- функции для работы с паролями.

Чтобы авторизовать пользователя необходимо создать подключение к серверу БД. В PDO для этого используется конструктор `new PDO()`, принимающий в качестве параметров настройки подключения:

```
return new \PDO(dsn: 'mysql:host=' . DB_HOST . ';port=' . DB_PORT .  
' ;charset=utf8;dbname=' . DB_NAME, username: DB_USERNAME);
```

Принимаемые параметры:

- драйвер БД. Подключение осуществляется к MySQL;
- хост сервера. Необходимо указать, так как сервер MySQL запущен локально;
- номер порта;
- имя базы данных;
- имя пользователя MySQL;
- пароль пользователя БД (не используется).

Значения принимаемых параметров для удобства записаны в константы.

```
const DB_HOST = 'localhost';  
const DB_PORT = '3306';  
const DB_NAME = 'EcoGIS';  
const DB_USERNAME = 'root';
```

4.2.1 Страница авторизации

После заполнения полей и нажатия на кнопку «Продолжить» происходит валидация данных, проверяющая их корректность.

```
if (!$user) {  
    setMessage('error', "Пользователь $email не найден");  
    redirect('/login.php');
```

```

}
if (!password_verify($password, $user['password'])) {
    setMessage('error', 'Неверный пароль');
    redirect('/login.php');
}

```

Если валидация прошла успешно, то происходит авторизация. Код функции авторизации пользователя представлен ниже:

```

function findUser(string $email): array|bool
{
    $pdo = getPDO();
    $stmt = $pdo->prepare("SELECT * FROM users WHERE email =
:email");
    $stmt->execute(['email' => $email]);
    return $stmt->fetch(\PDO::FETCH_ASSOC);
}

```

Интерфейс страницы авторизации пользователей приведен на рисунке 4.1.

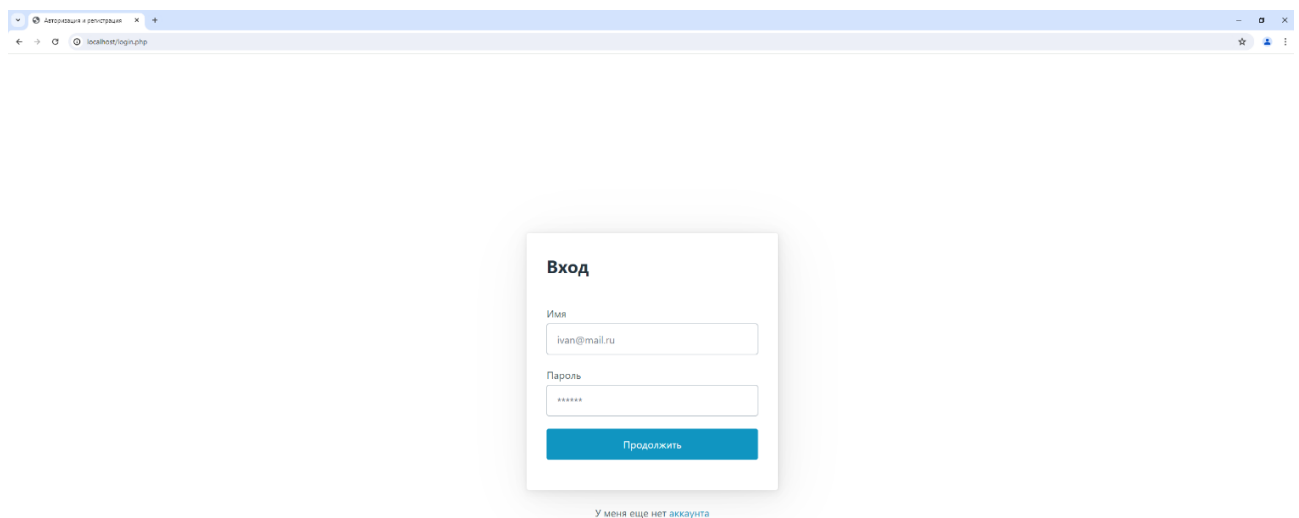


Рисунок 4.1 – Страница авторизации пользователя

Страница авторизации включает в себя следующие объекты:

- форма типа form для авторизации;
- поле ввода электронной Input type=text;

- поле ввода пароля Input type=password;
- кнопка подтверждения Button type=submit;
- ссылка на страницу регистрации href.

4.2.2 Страница регистрации

После заполнения полей и нажатия на кнопку «Продолжить» происходит валидация данных, проверяющая их корректность.

```
if (empty($name)) {
    setValidationError('name', 'Неверное имя');
}
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    setValidationError('email', 'Указана неправильная почта');
}
if (empty($password)) {
    setValidationError('password', 'Пароль пустой');
}
if ($password !== $passwordConfirmation) {
    setValidationError('password', 'Пароли не совпадают');
}
```

Если валидация прошла успешно, то происходит регистрация. Код функции регистрации пользователя представлен ниже:

```
$pdo = getPDO();
$query = "INSERT INTO Users (name, email, password) VALUES (:name,
:email, :password)";
$params = [
    'name' => $name,
    'email' => $email,
    'password' => password_hash($password, PASSWORD_DEFAULT)
];
$stmt = $pdo->prepare($query);
try {
    $stmt->execute($params);
} catch (\Exception $e) {
    die($e->getMessage());
}
```

}

Интерфейс страницы авторизации пользователей приведен на рисунке 4.2.

Регистрация

Имя
Иванов Иван

E-mail
ivan@mail.ru

Пароль

Подтверждение

Продолжить

[У меня уже есть аккаунт](#)

Рисунок 4.2 – Страница регистрации пользователя

Страница регистрации включает в себя следующие объекты:

- форма типа form для регистрации;
- поле ввода Имени и Отчества Input type=text;
- поле ввода электронной Input type=text;
- поле ввода пароля Input type=password;
- поле ввода, подтверждающее введенный ранее пароль Input type=password;
- кнопка подтверждения Button type=submit;
- ссылка на страницу авторизации href.

4.3 Работа с картами

Для отображения карты была использована библиотека OpenLayers. Чтобы отобразить карту необходимо объявить константу типа Map, в ней описать параметры отображения и слои. Код функции отображения карты приведен ниже:

```
const map = new ol.Map({  
  view: new ol.View({
```

```

        center: ol.proj.fromLonLat([45.0046400, 53.2006600]),
        zoom: 12
    }},
    layers: [
        new ol.layer.Tile({
            source: new ol.source.OSM()
        })
    ],

    target: "js-map"
});

```

Интерфейс основной страницы приведен на рисунке 4.3.

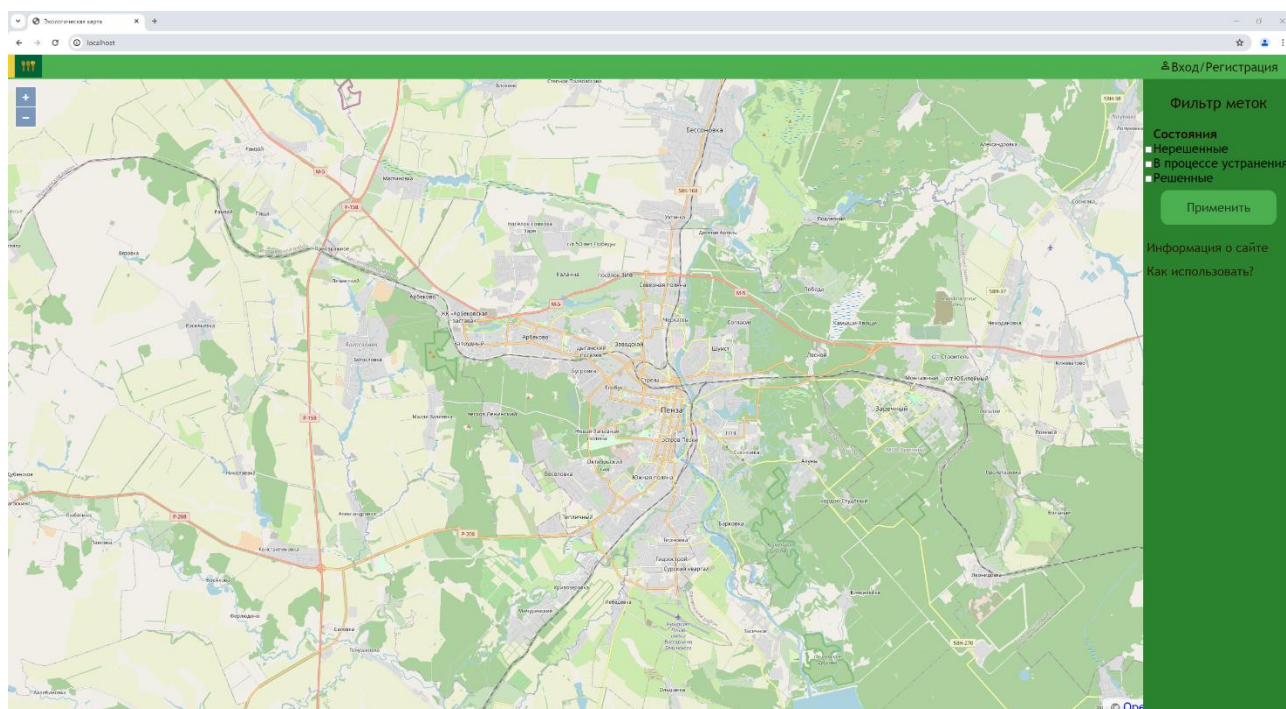


Рисунок 4.3 – Основная страница приложения

Основная страница включает в себя следующие объекты:

- ссылка на аутентификацию href;
- отображаемая карта типа ol.Map;
- поле ввода критического состояния маркера Input type=checkbox;
- поле ввода удовлетворительного состояния маркера Input type=checkbox;
- поле ввода хорошего состояния маркера Input type=checkbox;

- кнопка подтверждения выбранного состояния маркера Button type=submit;
- ссылка на страницу правил использования сайта href;
- ссылка на страницу информации о сайте href.

4.3.1 Отображение маркеров

Работа с картой подразумевает отображение маркеров. Для того чтобы отобразить маркеры используется специальный векторный слой. Код создания векторного слоя приведен ниже:

```
var markersBad = new ol.layer.Vector({
  source: new ol.source.Vector(),
  visible: true,
  title: 'MarkersBad'
});
```

Так как большое количество объектов на векторном слое может отрицательно сказаться на производительности было решено для каждого типа маркеров создать собственный слой.

Благодаря разделению типов маркеров на слои было упрощено создание фильтрации маркеров. Код функции фильтрации приведен ниже:

```
function checkCheckBox() {
  map.removeLayer(markersBad);
  map.removeLayer(markersFine);
  map.removeLayer(markersGood);
  let checkbox1 = document.getElementById('Bad');
  let checkbox2 = document.getElementById('Fine');
  let checkbox3 = document.getElementById('Good');
  if (checkbox1.checked) {
    map.addLayer(markersBad);
  }
  if (checkbox2.checked) {
    map.addLayer(markersFine);
  }
  if (checkbox3.checked) {
    map.addLayer(markersGood);
  }
}
```

```
}
```

Для того чтобы добавить маркеры на векторный слой необходимо использовать асинхронный запрос, который позволит выводить маркеры из БД не перезагружая страницу. Код запроса приведен ниже:

```
$.ajax({
    url: '/getmarkers.php',
    method: 'POST',
    dataType: 'JSON',
    success: function(data) {
        let response = JSON.stringify(data);
        let i = 0;
        while (response.indexOf("{}") != -1) {
            podsos[i] = new MARKERS_ON_MAP(response.substring(0,
response.indexOf("{}")));
            response = response.substring(response.indexOf("{}")+1);
            switch(podsos[i].getState()) {
                case 'Map_Marker_Bad': {
                    (markersBad.getSource().addFeature(podsos[i].createMarker()));
                    break;}
                case 'Map_Marker_Fine': {
                    (markersFine.getSource().addFeature(podsos[i].createMarker()));
                    break;}
                case 'Map_Marker_Good': {
                    (markersGood.getSource().addFeature(podsos[i].createMarker()));
                    break;}
            }
        }
        i++;
    }
})
```

Внешний вид страницы с отображенными маркерами приведен на рисунке 4.4.

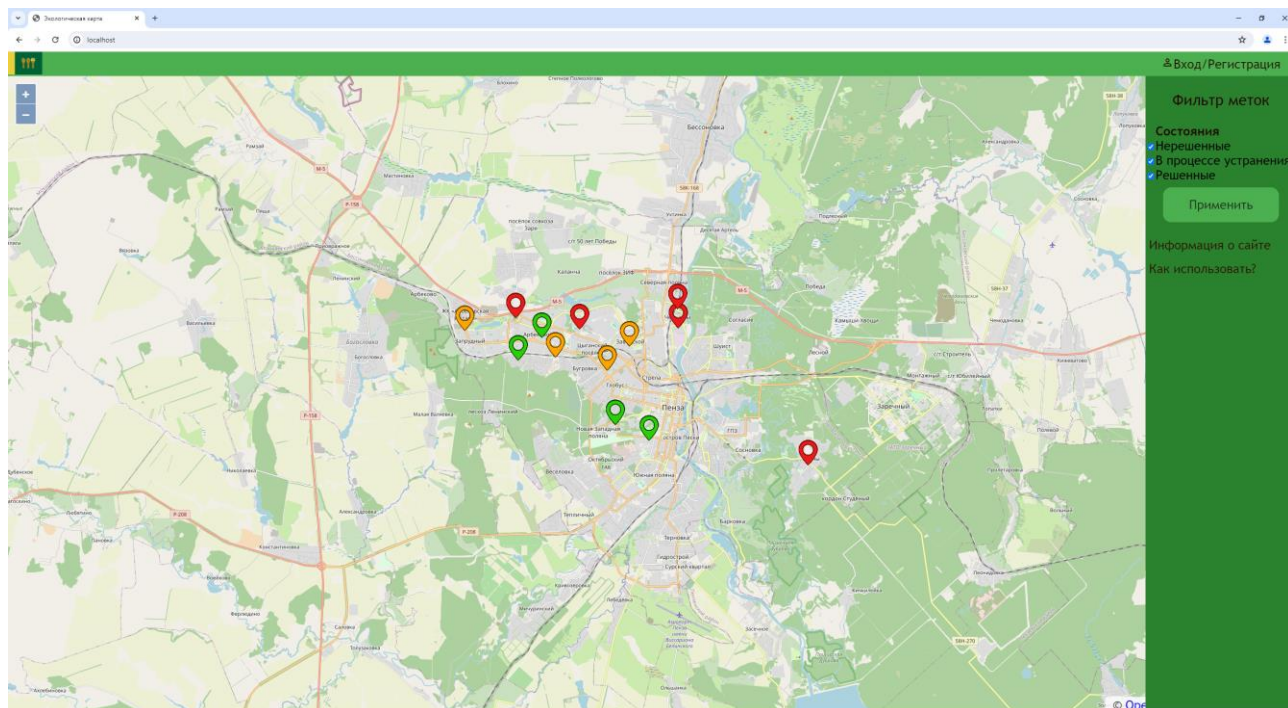


Рисунок 4.4 – Отображенные маркеры

Также маркеры имеют окна со своим описанием. Во всплывающем окне содержится описание маркера и его состояние. Код реализации всплывающего окна приведен ниже:

```
const popup = new ol.Overlay({
  element: document.getElementById('popup')
});

let f = map.forEachFeatureAtPixel(e.pixel, function(feature) {
  return feature;
});

let r = true;
for (let i = 0; i < podsos.length; i++) {
  if (podsos[i].getMarker() === f) {
    const content = f.getGeometry().getCoordinates();
    popup.setPosition(content);
    map.addOverlay(popup);
    let st;
    switch (podsos[i].getState()) {
      case 'Map_Marker_Bad':{
        st = 'Ужасно';
```

```

        break;
    }
    case 'Map_Marker_Fine':{
        st = 'Нормально';
        break;
    }
    case 'Map_Marker_Good':{
        st = 'Хорошо';
        break;
    }
}

document.getElementById('popup-content').innerHTML = `
    <p>Описание: ${podsos[i].getDescrip()}</p>
    <p>Состояние: ${st}</p>`;
document.querySelector('#popup-closer').addEventListener('click', function () {
    map.removeOverlay(popup);
});
isp = true;
r = false;
}}

```

Внешний вид страницы с выведенным описанием приведен на рисунке 4.5.

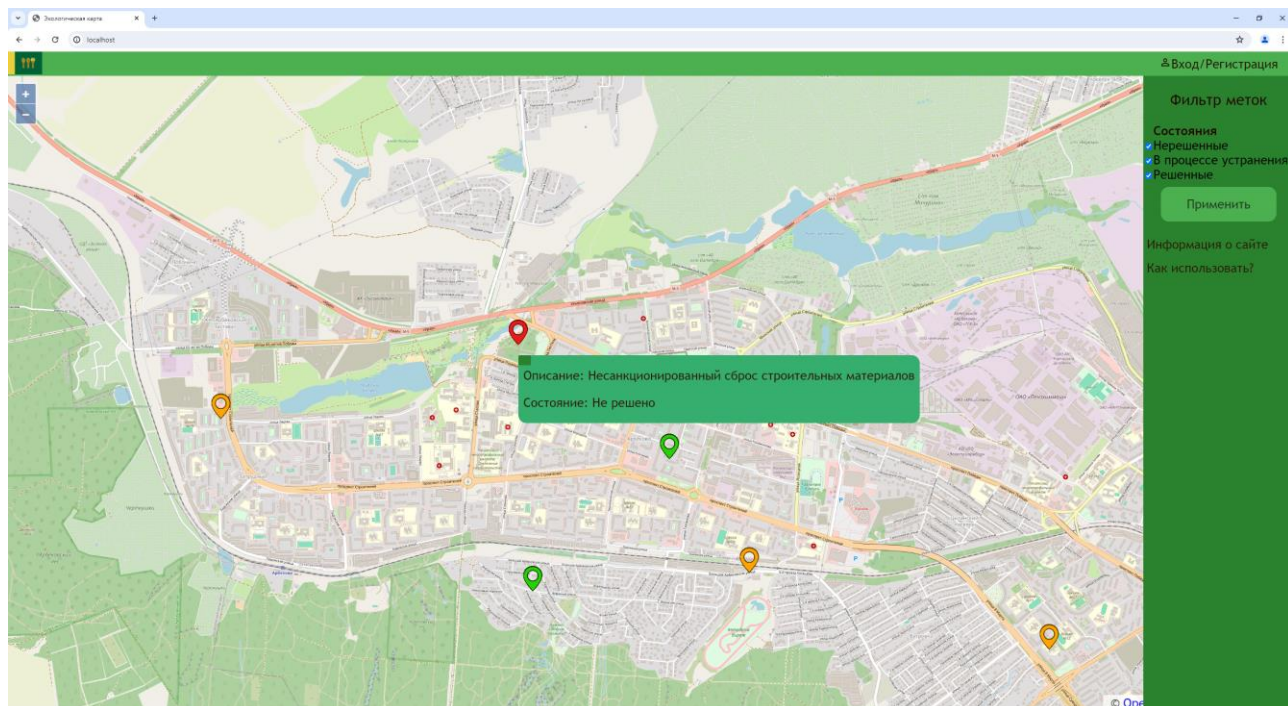


Рисунок 4.5 – Всплывающее окно с описанием

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.09.001 ПЗ

Лист

38

4.3.2 Добавление маркера на карту

Внешний вид формы для добавления маркера приведен на рисунке 4.6.

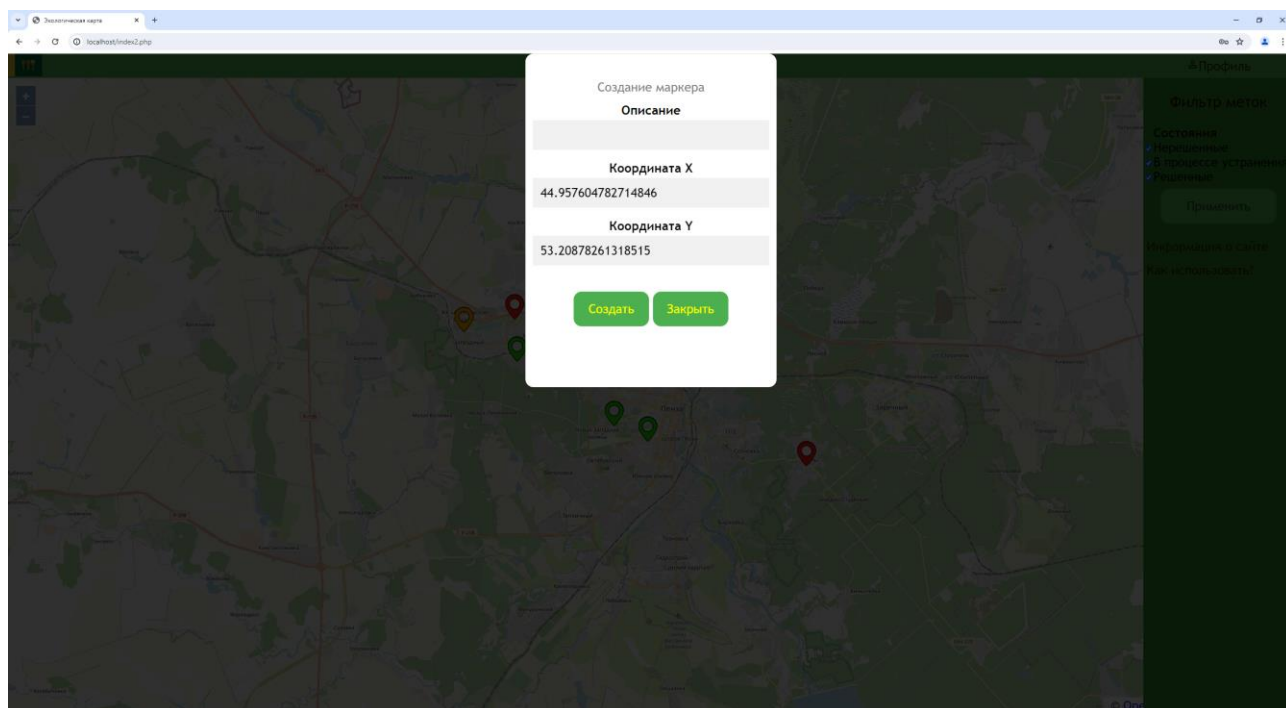


Рисунок 4.6 – Форма для добавления маркера

Добавление маркера реализовано следующим образом: после авторизации пользователь попадает на страницу, где после одиночного клика по карте появляется форма, после заполнения которой и подтверждение введенных данных на карту добавляется новый маркер. Все поля обязательны к заполнению и не могут быть пустыми.

Форма добавления маркера включает в себя следующие объекты:

- поле ввода описания маркера `Input type=text`;
- поле ввода широты местоположения маркера `Input type=text`;
- поле ввода долготы маркера местоположения `Input type=text`;
- кнопка подтверждения создания маркера `Button type=submit`;
- кнопка отмены создания маркера `Button type=reset`;
- ссылка на страницу информации о сайте `href`.

4.4 Личный кабинет пользователя

Личный кабинет пользователя необходим для возможности просмотра собственно добавленных маркеров в табличном представлении, а также для доступа к функции изменения маркеров. Пользователь с обычными правами может просматривать и изменять только свои маркеры. Пользователь с правами администратора может просматривать и изменять все маркеры. Изменение маркеров реализовано с помощью формы, в которую нужно занести актуальные на текущий момент сведения о маркере и подтвердить изменения. Внешний вид страницы личного кабинета администратора и пользователя приведен на рисунках 4.7, 4.8.

Номер записи	Долгота	Широта	Описание	Состояние
1	45.0224	53.2277	ABC	Не решенный
2	45.0219	53.2354	МОЙКА	Не решенный
3	44.9293	53.2238	Гора мусором размером с человека	Решенный
4	44.989	53.2203	10 разбитых бутылок	В процессе устранения
5	44.9794	53.1882	уауауауау	Решенный
6	44.9739	53.2104	Живая ворона	В процессе устранения
7	45.0024	53.1819	Мертвый голубь	Решенный
8	44.9387	53.2158	пока пока	В процессе устранения
9	44.8767	53.2266	хихихаха	В процессе устранения
10	44.9133	53.2145	шаукаршкур шукрашгушг	Решенный
11	44.9551	53.2273	мцаовщопщ	Не решенный
12	44.9116	53.2318	Несанкционированный сброс строительных материалов	Не решенный
13	45.1111	53.1717	несанкционированная свалка строительного мусора	Не решенный

[вернуться к добавлению маркеров](#)

Рисунок 4.7 – Личный кабинет администратора

Номер записи	Долгота	Широта	Описание	Состояние
7	45.0024	53.1819	Мертвый голубь	Решенный
10	44.9133	53.2145	шаукаршкур шукрашгушг	Решенный

[вернуться к добавлению маркеров](#)

Рисунок 4.8 – Личный кабинет пользователя

Страница личного кабинета включает в себя следующие объекты:

- таблица для вывода информации о маркерах тип table;
- кнопка выхода из аккаунта Button type=submit;
- ссылка на страницу с возможность добавления маркеров о сайте href.

На рисунке 4.9 приведен пример редактирования маркера из личного кабинета пользователя с правами администратора.

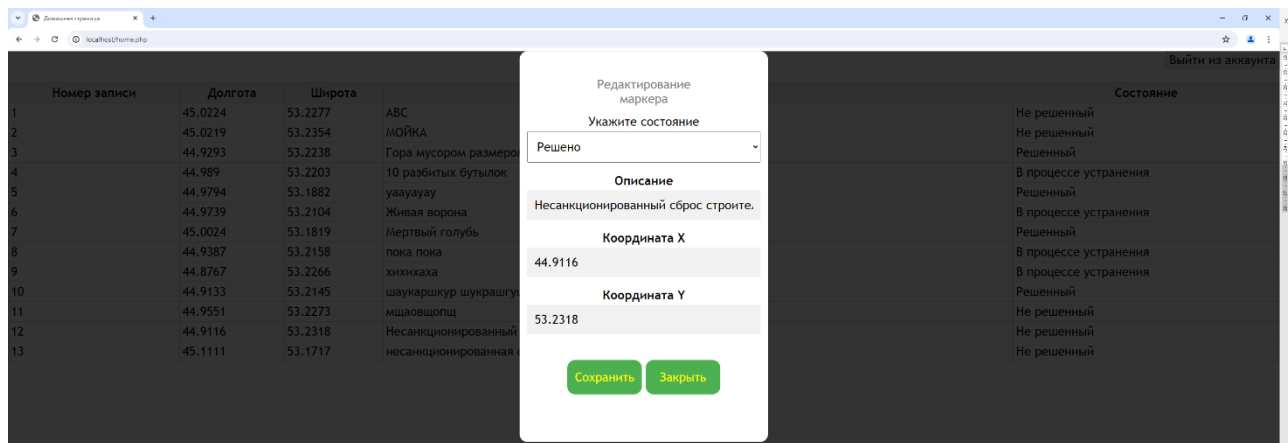


Рисунок 4.9 – Редактирование маркера из личного кабинета администратора

Форма редактирования маркера включает в себя следующие объекты:

- выпадающий список для выбора состояния маркера Input type=select;
- поле ввода описания маркера Input type=text;
- поле ввода широты местоположения маркера Input type=text;
- поле ввода долготы местоположения маркера Input type=text;
- кнопка подтверждения редактирования маркера Button type=submit;
- кнопка отмены редактирования маркера Button type=reset;
- ссылка на страницу информации о сайте href.

4.5 Страницы с дополнительной информацией

Также на сайте реализованы информационные страницы.

Страница с информацией о сайте предоставляет новому пользователю общую информацию об экологии и экологическом состоянии Пензенской области (рисунок 4.10).

Страница с правилами пользования (рисунок 4.11) предоставляет новому пользователю полный перечень требований и условий использования сайта, а также ответы на возможные вопросы, которые могут возникнуть у посетителя сайта при попытке совершения каких-либо действий (отфильтровать маркеры, добавить маркер, зарегистрироваться на сайте).

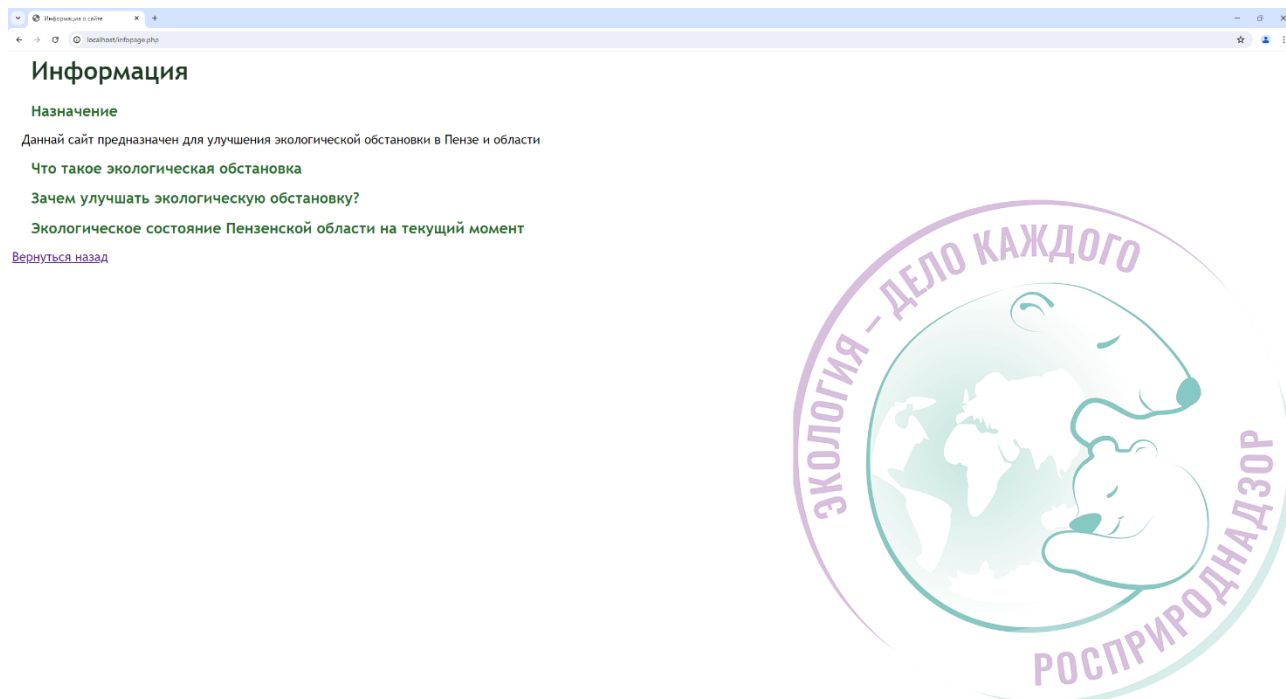


Рисунок 4.10 – Страница «Информация о сайте»

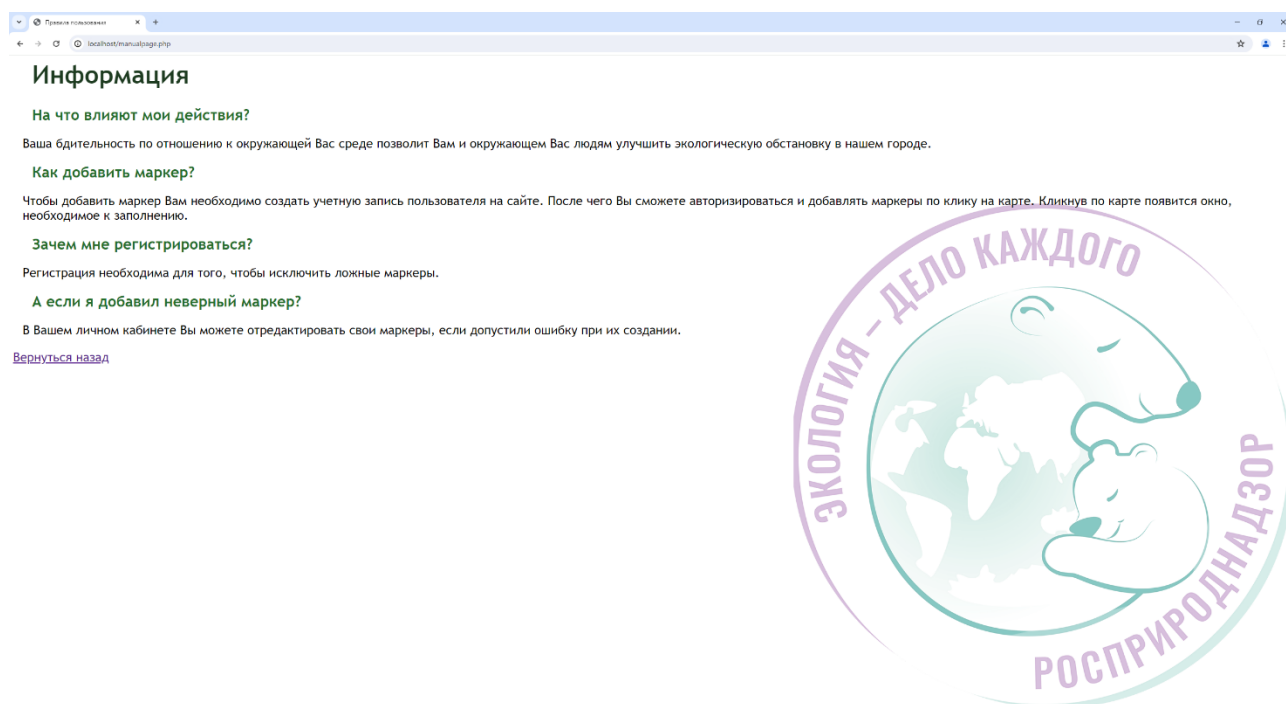


Рисунок 4.10 – Страница «Правила пользования»

					ПГУ 1.090501.09.001 ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		

5 Технико–экономическое обоснование

Для расчета себестоимости программного обеспечения необходимо учитывать следующие параметры:

- затраты на содержание помещения;
- затраты на приобретение оборудования;
- затраты на приобретение лицензионного программного обеспечения;
- затраты на дополнительные материалы;
- затраты на заработную плату;
- затраты на страховые взносы;
- затраты на обслуживание.

Разработка Web–приложения осуществлена на базе собственного производственного помещения общей площадью 30 кв.м. Содержание помещения в месяц составляет 15 000 рублей. Затраты в период разработки (2 месяца) составят 30000 рублей.

Для разработки Web–приложения необходима вычислительная техника, обладающая следующими характеристиками:

- операционная система не ниже Windows 10;
- ОЗУ не менее 8 Гб;
- место на диске не менее 50 Гб;
- видеокарта с объемом видеопамати не менее 4 Гб и поддержкой API DirectX 12;
- процессор – многоядерный с частотой не ниже 2,3 ГГц.

Исходя из описанных выше требований было решено приобрести вычислительную технику, лицензионное программное обеспечение и расходные материалы по отдельности в разных магазинах. Данные по потребности в оборудовании представлены в таблице 5.1.

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 5.1 – Оборудование для разработки

Оборудование	Источник	Кол-во	Ориентировочная цена за единицу, руб.	Потребляемая мощность, Ватт
Персональный компьютер ARDOR GAMING NEO M143	ООО «ДНС»	5	69299	500
Комплект клавиатура + мышь Logitech MK235 Grey	ПАО «М. Видео»	5	2999	2,6
Монитор Cooler Master GM27-FQ	ООО «ДНС»	10	18299	30

На основании таблицы 5.1 затраты на оборудование в период разработки составят 544480 рублей. —

Для разработки приобретено лицензионное программное обеспечение. Данные по потребности в лицензионном ПО представлены в таблице 5.1.

Таблица 5.2 – Лицензионное программное обеспечение для разработки

Лицензионное программное обеспечение	Источник	Цена, руб.	Срок действия лицензии
Microsoft Windows 11 pro	«Ситилинк»	30000	бессрочно
OpenLayers	OSGeo	0	бессрочно

Лицензионное программное обеспечение	Источник	Цена, руб.	Срок действия лицензии
phpMyAdmin	phpMyAdmin Developer Team	0	бессрочно

На основании таблицы 5.2 затраты на лицензионное программное обеспечение (в расчете на 2 месяца разработки) составят 90000 рублей.

Для разработки приобретены дополнительные материалы. Перечень дополнительных материалов представлен в таблице 5.3.

Таблица 5.3 – Перечень дополнительных материалов

Материал	Источник	Количество, штук	Цена за единицу, рублей
Стол компьютерный	ООО «ОНИКС»	5	8630
Кресло	ООО «ОНИКС»	5	6590
Лампа настольная	ООО «ДНС»	5	1999
Канцтовары	ООО «Чернильница»	5	1000
Бумага	ООО «Чернильница»	5	430

На основании таблицы 5.3 затраты на дополнительные материалы составят 78000 рублей.

Чтобы быстро и качественно разработать Web–приложение, необходимо собрать команду разработчиков. Состав коллектива и заработная плата разработчиков Web–приложения представлена в таблице 5.4.

Таблица 5.4 – Занятость и заработная плата разработчиков

Специалист	Количество, человек	Зарботная плата, рублей
Руководитель проекта	1	100000

Специалист	Количество, человек	Зарботная плата, рублей
JS Разработчик	1	80000
Backend Разработчик (PHP)	1	80000
Web Дизайнер	1	40000
Тестирующий	2	45000

Затраты на заработную плату включают в себя:

– затраты на основную заработную плату разработчика, исходящие из установленной заработной платы, занятости специалистов в процессе разработки программного продукта (в днях);

– затраты на дополнительную заработную плату, составляющую 15% от затрат на основную заработную плату

Учитывая 5–дневную рабочую неделю, количество рабочих дней в месяце в среднем равно 21.

Затраты на основную и дополнительную заработные платы разработчиков представлены в таблице 5.5.

Таблица 5.5 – Затраты на основную и дополнительную заработные платы

Исполнитель	Количество, человек	Сумма заработной платы, рублей/человек	Период занятости, дней	Основная заработная плата, рублей/человек	Дополнительная заработная плата, рублей/человек	Всего рублей
Руководитель проекта	1	100000	42	200000	30000	230000
JS Разработчик	1	80000	42	160000	24000	182000

Исполнитель	Количество, человек	Сумма заработной платы, рублей/человек	Период занятости, дней	Основная заработная плата, рублей/человек	Дополнительная заработная плата, рублей/человек	Всего рублей
Backend Разработчик (PHP)	1	80000	42	160000	24000	182000
Web Дизайнер	1	40000	42	80000	13500	93500
Тестировщик	2	45000	42	90000	28500	118500
Итого	6	390000	210	780000	148500	924500

Согласно таблице 5.5, затраты на заработную плату разработчиков составят 924500 рублей.

Отчисления на страховые взносы по нормам, установленным законодательством в ПФР, ФОМС, ФСС РФ рассчитываются согласно «Налоговому кодексу Российской Федерации (часть вторая)» от 05.08.2000 N 117–ФЗ (ред. от 29.05.2024) НК РФ Раздел XI. Страховые взносы в Российской Федерации.

Единый тариф (единые размеры) страховых взносов для плательщиков, производящих выплаты физическим лицам, установлен на:

- обязательно пенсионное страхование (ОПС);
- обязательное социальное страхование на случай временной нетрудоспособности и в связи с материнством (ОСС);
- обязательное медицинское страхование (ОМС).

В пределах установленной единой предельной величины базы для исчисления страховых взносов единый тариф страховых взносов равен 30%, свыше – 15,1%. [статья 425 налогового кодекса Российской федерации]

Суммарные отчисления на страховые взносы составят 277350 рублей.

Общая сумма затрат на разработку продукта составляет 1989575 рублей.

Ежемесячные затраты на обслуживание ПО зависят от количества пользователей. Для 20000 пользователей ежемесячные затраты составят 74000 рублей.

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

6 Безопасность жизнедеятельности

За безопасностью жизнедеятельности человека необходимо наблюдать на всех этапах реализации проекта, будь то разработка, инициация, планирование, исполнение, контроль или завершение.

Создание оптимальных условий работы в большей степени зависит от правильной оценки негативных факторов, влияющих на человека на рабочем месте. Во время нахождения на рабочем месте на человека могут влиять различные факторы производственной среды, чрезмерная физическая и умственная нагрузка, нервно-эмоциональное напряжение, а также разное сочетание этих причин. Изменения, вызванные перечисленными факторами, могут вызвать в организме человека различные по тяжести изменения.

6.1 Потенциально опасные факторы при работе с ПЭВМ

Опасные и вредные производственные факторы по природе возникновения делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические.

При работе с ПЭВМ на человека могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- опасность поражения электрическим током;
- повышенный уровень статического электричества;
- блеклость экрана дисплея;

– недостаточная освещенность рабочего места.

К химически опасным факторам, постоянно действующим на программиста, относятся активные частицы, возникающие в результате ионизации воздуха при работе компьютера.

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на программиста в течение его рабочей смены, можно отнести следующие:

- нервно-эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора [4].

6.2 Классификация помещения

Кабинет программиста является помещением без повышенной опасности категории В1–В4 (пожароопасность) I категории (выполняются легкие физические работы), поэтому должны соблюдаться следующие требования:

Таблица 6.1 – Требования к параметрам рабочего места программиста

Период года	Температура воздуха, °С	Скорость движения воздуха, м/с	Относительная влажность воздуха, %
Холодный	22–24	До 0,1	40–60
Теплый	23–25	0,1–0,2	40–60

Чтобы создать оптимальные условия труда на рабочем месте (таблица 6.1) необходимо наличие специальных приборов и поддержание их в исправном состоянии. Такими приборами могут быть радиатор отопления, вентилятор и кондиционер, которые способны эффективно поддерживать температуру воздуха, влажность и скорость движения воздуха в диапазоне допустимых значений, не всегда совпадающих с параметрами окружающей среды.

Работодатель обязан обеспечить:

- содержание электрооборудования в работоспособном состоянии и его эксплуатацию в соответствии с требованиями нормативной технической документации;
- своевременное и качественное проведение профилактических работ, ремонта, модернизации и реконструкции энергетического оборудования;
- надежность работы электроустановок и безопасность их обслуживания;
- предотвращение использования технологий и методов работы, оказывающих отрицательное воздействие на окружающую среду;
- учет и анализ нарушений в работе электроустановок, несчастных случаев и принятие мер по устранению причин их возникновения;
- разработку должностных инструкций и производственных инструкций для электротехнического персонала;
- выполнение предписаний органов Госэнергонадзора [5].

6.3 Размещение оборудования и организация рабочих мест

Рабочие места программистов должны быть расположены таким образом, чтобы естественный свет падал с боку. Рекомендуемое направление естественного света – слева, допускаемое – справа. Не допускается направление естественного света со стороны спины или лица.

Экран монитора должен находиться на расстоянии 60–70 см от глаз.

Должны быть использованы специальные компьютерные столы в сочетании с письменными.

6.4 Освещение

Главная задача при проектировании освещения – создать качественное освещение, а для этого необходимо рассчитать количество и яркость искусственных источников света. Необходимо чтобы искусственные источники света заменяли естественный источник света или дополняли его.

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

Если произвести некорректный расчет, и неправильно спроектировать освещение, сделав его тусклым, то это может привести к перенапряжению зрения, ослаблению внимания, а также может способствовать преждевременной утомляемости, если же его сделать чрезмерно ярким, то это может вызвать раздражение и повысить сухость в глазах. При этом важно не только количество освещения, но и его направленность. Пренебрежение этой характеристикой может привести к офтальмологическим проблемам или несчастным случаям, вызванным бликами на экранах мониторов.

Общие правила организации освещения заключаются в следующем:

- следует избегать большого контраста между яркостью экрана и окружающего пространства. Оптимальным считается их выравнивание;
- запрещается работа с компьютером в темном или полутемном помещении;
- освещение в помещениях с ПК должно быть смешанным: естественным (за счет солнечного света) и искусственным (за счет дополнительных источников света).

В соответствии с санитарными нормами, необходимо использовать люминесцентные лампы типа ЛД, так как цвет их излучения максимально приближен к дневному, лучше влияет на человеческий глаз, оказывая негативное влияние на органы человека и его нервную систему в меньшем в меньшем процентном соотношении в сравнении с лампами ЛБ, чей цвет излучения белый. Так же важно учесть коэффициент пульсации лампы, который так же оказывает негативное влияние на здоровье и самочувствие человека. При выборе ламп для рабочего помещения также необходимо отталкиваться от мерцания лампы. Показатель мерцания напрямую зависит от коэффициента пульсации лампы: чем выше коэффициент пульсации, тем выше показатель мерцания.

Следует отметить, что существуют специальные люминесцентные лампы, которые излучают свет различного «качества», имитируя таким образом, полный спектр естественного солнечного света. Они меньше раздражают, чем любые другие лампы искусственного освещения. [6]

Чтобы качественно спроектировать освещение необходимо распределить источники света таким образом, чтобы они:

- были равномерно распределены по комнате;
- были составлены в линии (сплошные или прерывистые).

Если рабочие места располагаются рядами (рисунок 6.1), то линии светильников должны располагаться с боку от рабочего места параллельно зрению работника.

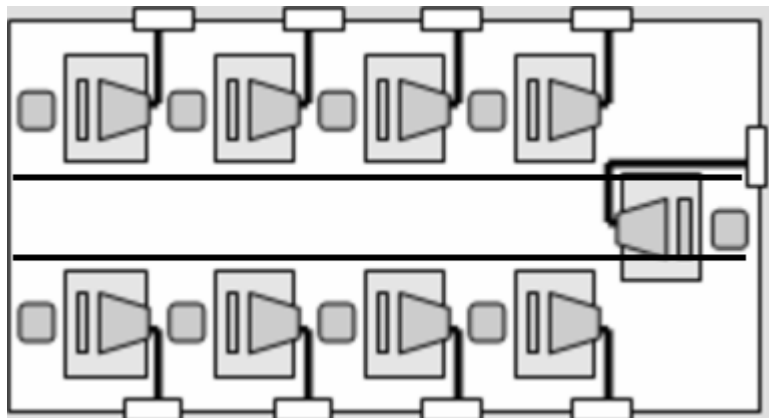


Рисунок 6.1 – Расположение рабочих мест рядами

Если рабочие места располагаются по периметру рабочего помещения (рисунок 6.2), то светильники должны располагаться непосредственно над рабочим столом работника.

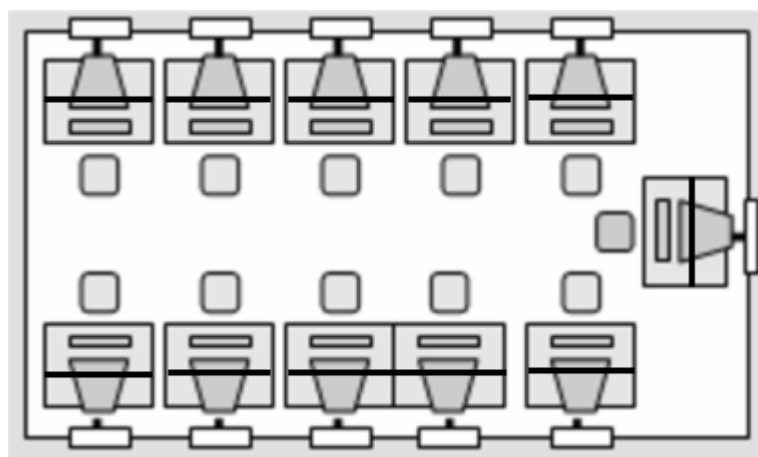


Рисунок 6.2 – Расположение рабочих мест по периметру помещения

Если правильно выстроить освещение в рабочем помещении, то это может поспособствовать повышению производительности труда на 6% при средних зрительных нагрузках и на 15% при тяжелых [6].

Если в обязанности работника входит не только работа с компьютером, но и с документами, то на его рабочее место необходимо установить лампу, у которой:

- имеется регулировка наклона осветительного элемента;
- имеется регулировка яркости.

В этом случае надо следить, чтобы свет от лампы не действовал раздражающе и не создавал бликов на экране.

В таблице 6.2 приведены нормируемые значения для освещения в рабочих помещениях.

Таблица 6.2 – Характеристики освещения в рабочих помещениях

Показатель	Нормируемое значение
Освещенность в зоне расположения рабочего документа для пользователей персональных компьютеров	300 – 500 ЛК
Освещенность экрана монитора при местном освещении	Не более 300 лк
Яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения	не более 200 кд/ кв.м.
Яркость бликов на экране монитора	Не более 40 кд/кв.м
Яркость потолка, при применении системы отраженного освещения	Не более 200 кд/кв.м.
Показатель ослепленности для источников общего искусственного освещения в производственных помещениях, предназначенных для размещения рабочих мест пользователей персональных компьютеров	Не более 20

6.5 Расчет освещенности рабочего места

Для того чтобы рассчитать, какое количество ламп необходимо использовать для рационального освещения рабочих мест, необходимо установить:

– размеры помещения:

- 1) длина помещения 6м;
- 2) ширина помещения 5м;
- 3) высота помещения 3м;

– характеристику отражающих поверхностей ρ :

- 1) побеленный потолок с коэффициентом отражения 70%;
- 2) побеленные стены с коэффициентом отражения 50%.

Для начала необходимо определить площадь помещения по формуле:

$$S = A * B. \quad (1)$$

где A – длина помещения ($A = 6$ м);

B – ширина помещения ($B = 5$ м).

Подставив значения в формулу (1), получим:

$$S = 6 * 5 = 30 \text{ м}^2.$$

Далее необходимо рассчитать индекс помещения. Индекс помещения имеет прямую зависимость от коэффициента использования светового потока, чем выше индекс помещения, тем выше коэффициент светового потока и тем эффективнее используется поток от лампы освещения.

Индекс помещения определяется по формуле:

$$I = \frac{S}{h_p * S'} \quad (2)$$

где h_p – высота подвеса светильника ($h_p = 3$ м).

Подставив значения формулу (2), получим:

$$I = \frac{30}{3 * 30} = 0,9$$

При $I=0,9$, коэффициентов отражения: потолка $\rho_{\text{п}}=0,7$, стен $\rho_{\text{с}}=0,5$, коэффициент использования светового потока $h=0,46$. Далее нужно определить необходимое количество светильников. Определяется по формуле:

$$N = \frac{S}{4}, \quad (3)$$

где N – необходимое количество светильников.

Подставив значения формулу (3), получим:

$$N = \frac{30}{4} = 8 \text{ шт.}$$

Далее необходимо рассчитать световой поток лампы по формуле:

$$F = \frac{E_{\text{н}} * S * K * Z}{N * n * \eta} * 100\%, \quad (4)$$

где n – общее число ламп в светильнике ($n = 2$);

K – коэффициент запаса, учитывающий снижение освещенности вследствие старения ламп, запыления и загрязнения светильников ($K = 1,3$);

$E_{\text{н}}$ – минимальная освещенность помещения ($E_{\text{н}} = 400$ лк)

Z – отношение средней освещенности к минимальной ($Z = 1,1$).

Подставив значения формулу (4), получим:

$$F = \frac{400 * 30 * 1,3 * 1,1}{8 * 2 * 52} * 100\% = 2062,5 \text{ лм}$$

Благодаря проведенным ранее расчетам и полученным в следствии их результатам было принято решение использовать люминесцентные лампы ЛД40-4, так как излучаемый ими световой поток, наиболее близок к рассчитанному ранее.

Для того, чтобы убедиться в правильности выбора типа ламп, необходимо выполнить проверку. Для этого необходимо найти величину отклонения фактического светового потока лампы типа ЛД40–4 ($F_{\text{ф}}$) от рассчитанного ($F_{\text{р}}$):

$$\Delta F = \frac{F_{\text{ф}} - F_{\text{р}}}{F_{\text{р}}} * 100\%, \quad (5)$$

Подставив значения формулу (5), получим:

$$\Delta F = \frac{2450 - 2062,5}{2062,5} * 100\% = 5,06\%$$

Поскольку величина ΔF попадает в допустимый диапазон (–10% +20%), следовательно, лампа выбрана верно.

Таким образом, выполнение перечисленных выше требований к организации рабочего места обеспечит комфортные условия умственного труда [7].

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения дипломного проекта было сформировано техническое задание на разработку Web–приложения «Экологическая карта Пензенской области».

Был произведен выбор инструментов разработки, в частности среды разработки Visual Studio Code, языков программирования PHP и JavaScript, СУБД MySQL и локального сервера Open Server.

Выполнено проектирование архитектуры Web–приложения, в результате чего были разработаны: диаграмма развертывания, диаграмма прецедентов, диаграммы последовательностей, диаграмма состояний и диаграмма данных.

В результате выполнения дипломного проекта в соответствии с техническим заданием было разработано Web–приложение «Экологическая карта Пензенской области», созданное в среде разработки Visual Studio Code, с помощью языков программирования PHP и JavaScript, СУБД MySQL, и локального сервера Open Server.

В экономическом разделе был произведен расчет затрат на разработку Web–приложения. В разделе БЖД была произведена оценка опасных и вредных производственных факторов, влияющих на программиста на его рабочем месте, а также произведен расчет освещенности рабочего места.

Задание на дипломное проектирование выполнено в полном объеме.

Список использованной литературы

1. Внедрение и ведение геоинформационных систем, баз данных. [Электронный ресурс]: URL: <https://42kad.ru> (дата обращения 03.06.2024).
2. УДК 528.94 Н.В. Котельникова, С.Ф. Мазуров, Л.А. Пластинин, В.П. Ступин ИрГТУ, ФГУП «ВостСибАГП», Иркутск ЗАДАЧИ ГЕОИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ОРГАНОВ ВЛАСТИ, ХОЗЯЙСТВА, ОБРАЗОВАНИЯ И БИЗНЕСА В БАЙКАЛЬСКОМ РЕГИОНЕ СИБИРИ.
3. Постановление Главного государственного санитарного врача РФ от 03.06.2003 N 118 (ред. от 21.06.2016) "О введении в действие санитарно–эпидемиологических правил и нормативов СанПиН 2.2.2/2.4.1340–03" (вместе с "СанПиН 2.2.2/2.4.1340–03. 2.2.2. Гигиена труда, технологические процессы, сырье, материалы, оборудование, рабочий инструмент. 2.4. Гигиена детей и подростков. Гигиенические требования к персональным электронно–вычислительным машинам и организации работы. Санитарно–эпидемиологические правила и нормативы", утв. Главным государственным санитарным врачом РФ 30.05.2003) (Зарегистрировано в Минюсте России 10.06.2003 N 4673).
4. ГОСТ 12.0.003–74 Опасные и вредные производственные факторы. Классификация.
5. Сравнительный анализ Правил технической эксплуатации электроустановок потребителей электрической энергии, утвержденных приказом Минэнерго РФ от 12.08.2022 N 811, и Правил технической эксплуатации электроустановок потребителей, утвержденных приказом Минэнерго РФ от 13.01.2003 N 6.
6. «Часто ты проводишь время за компьютером?». Смирнова Людмила Викторовна, Райчихинский индустриальный техникум профессий. [Электронный ресурс]: URL: <http://d.120-bal.ru/informatika/40707/index.html?page=3> (дата обращения 03.06.2024).

7. Расчет освещения по площади помещения. [Электронный ресурс]: URL: <https://www.smsm.ru/articles/raschet-osveshcheniya/> (дата обращения 03.06.2024).

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А

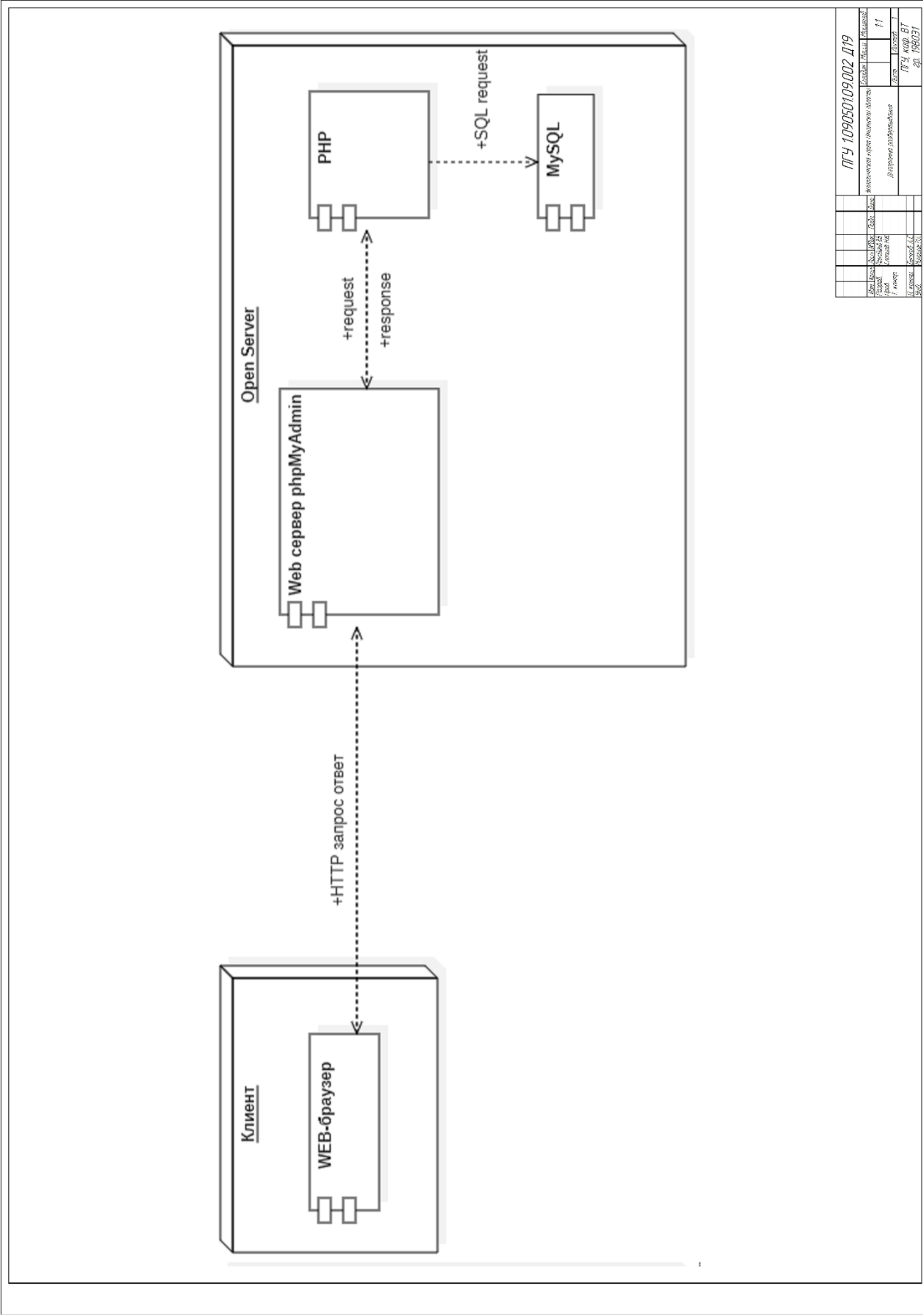
(обязательное)

Экологическая карта Пензенской области

UML-диаграммы

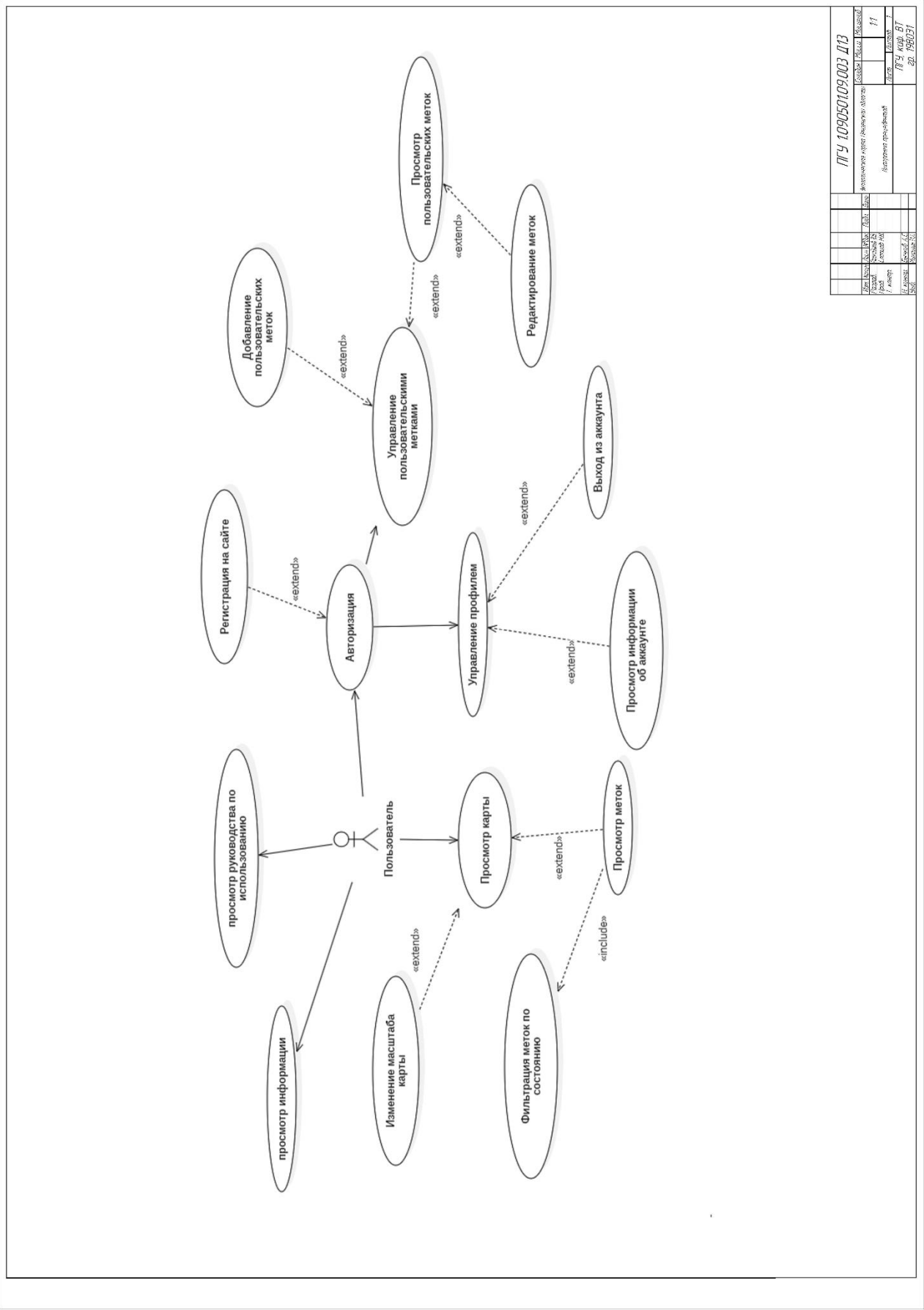
					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.1 – Диаграмма развертывания



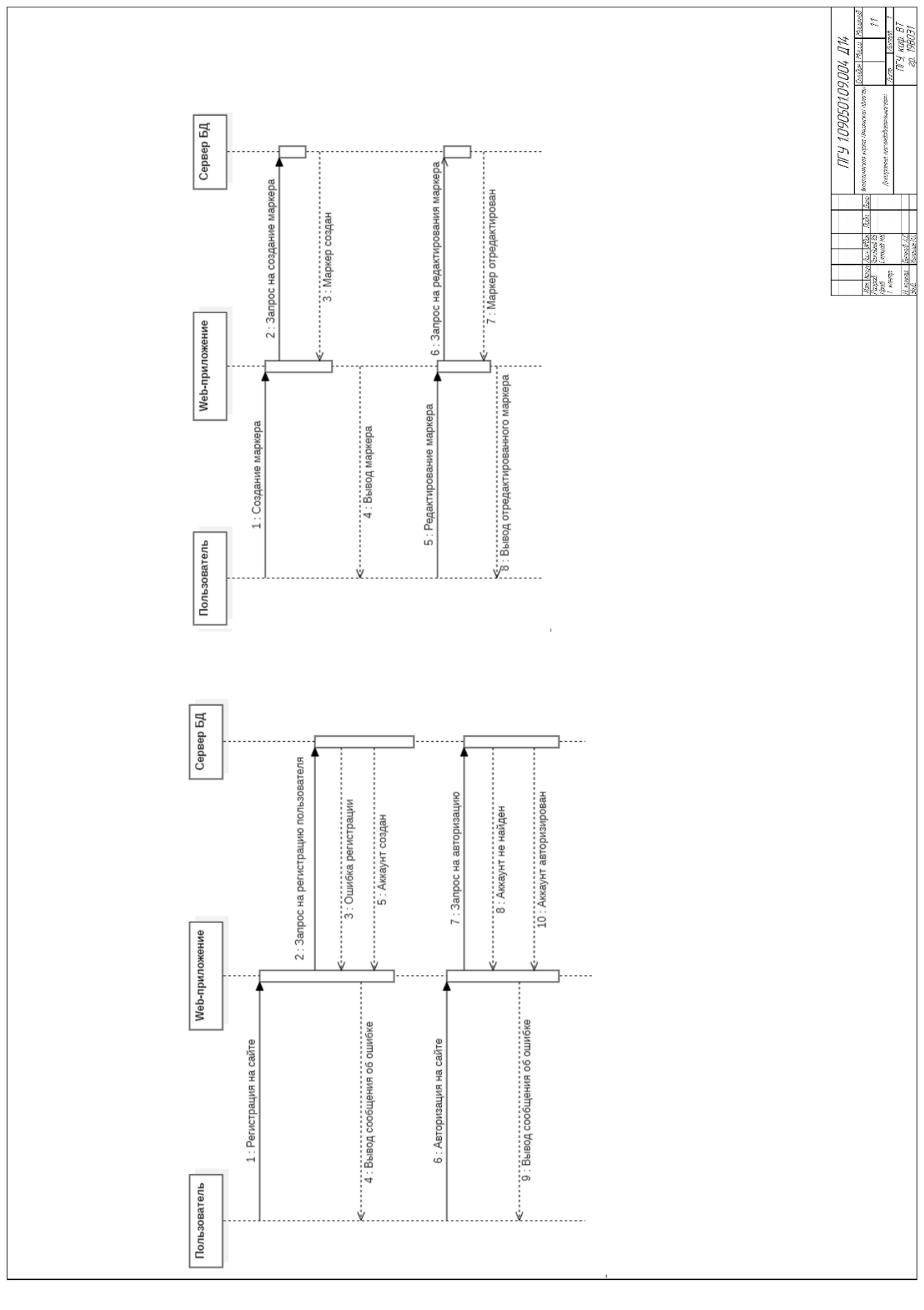
Изм.	Лист	№ докум.	Подпись	Дата
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30
31	31	31	31	31
32	32	32	32	32
33	33	33	33	33
34	34	34	34	34
35	35	35	35	35
36	36	36	36	36
37	37	37	37	37
38	38	38	38	38
39	39	39	39	39
40	40	40	40	40
41	41	41	41	41
42	42	42	42	42
43	43	43	43	43
44	44	44	44	44
45	45	45	45	45
46	46	46	46	46
47	47	47	47	47
48	48	48	48	48
49	49	49	49	49
50	50	50	50	50
51	51	51	51	51
52	52	52	52	52
53	53	53	53	53
54	54	54	54	54
55	55	55	55	55
56	56	56	56	56
57	57	57	57	57
58	58	58	58	58
59	59	59	59	59
60	60	60	60	60
61	61	61	61	61
62	62	62	62	62
63	63	63	63	63
64	64	64	64	64
65	65	65	65	65
66	66	66	66	66
67	67	67	67	67
68	68	68	68	68
69	69	69	69	69
70	70	70	70	70
71	71	71	71	71
72	72	72	72	72
73	73	73	73	73
74	74	74	74	74
75	75	75	75	75
76	76	76	76	76
77	77	77	77	77
78	78	78	78	78
79	79	79	79	79
80	80	80	80	80
81	81	81	81	81
82	82	82	82	82
83	83	83	83	83
84	84	84	84	84
85	85	85	85	85
86	86	86	86	86
87	87	87	87	87
88	88	88	88	88
89	89	89	89	89
90	90	90	90	90
91	91	91	91	91
92	92	92	92	92
93	93	93	93	93
94	94	94	94	94
95	95	95	95	95
96	96	96	96	96
97	97	97	97	97
98	98	98	98	98
99	99	99	99	99
100	100	100	100	100

Приложение А.2 – Диаграмма прецедентов



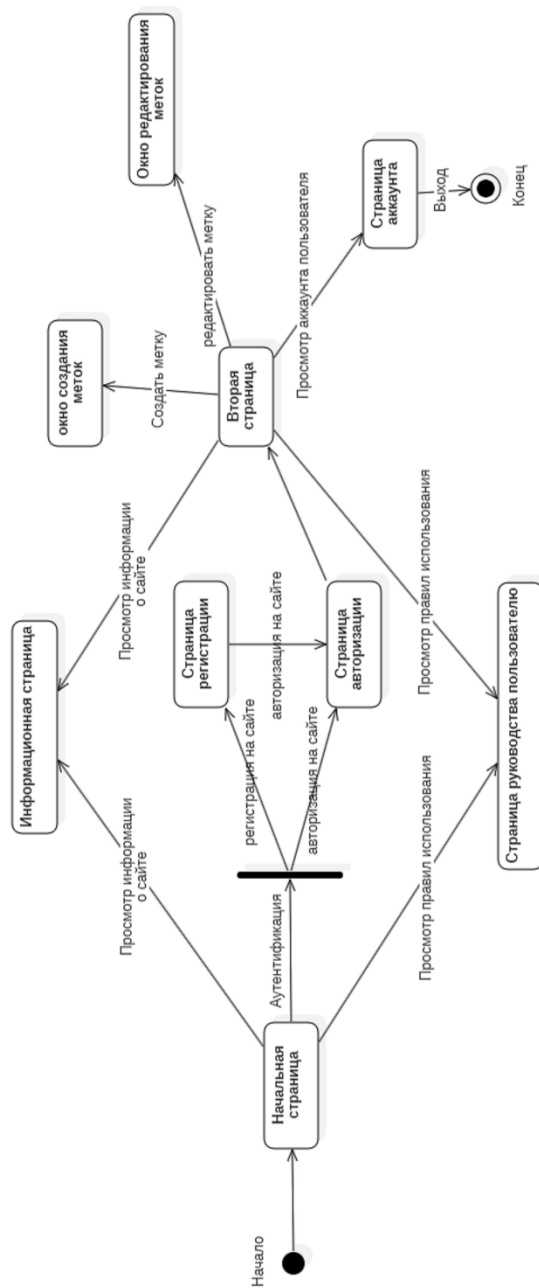
ПГУ 1.090501.09.001 ПЗ									
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель

Приложение А.3 – Диаграммы деятельности

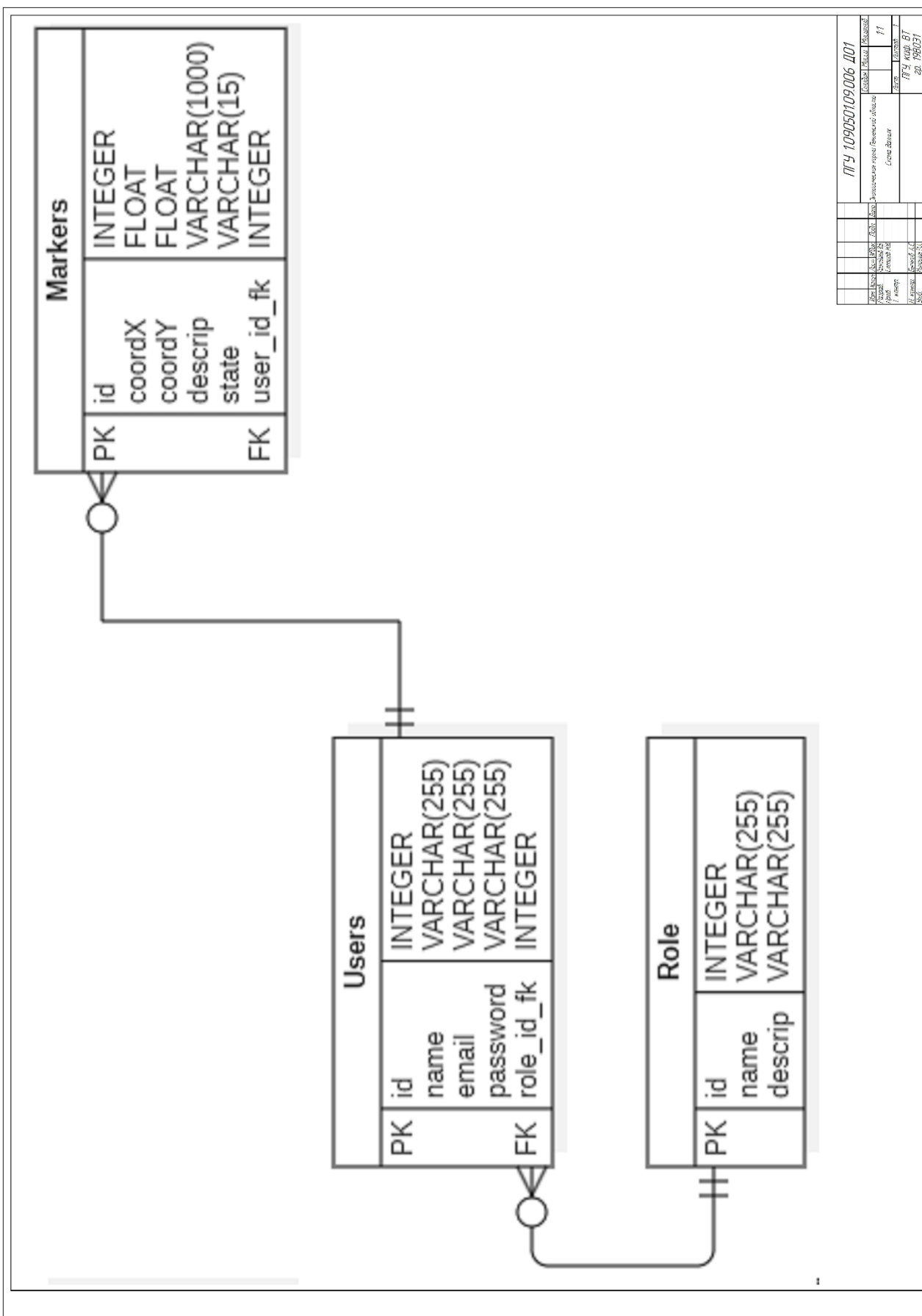


ПГУ 109050109.004 Д14									
Информация о документе (выполнено)									
Дата	Место	Содержание	Вид	Время	Состояние	Исполнитель	Исполнитель	Исполнитель	Исполнитель
14.09.2014	г. Москва	Создание маркера	1	14:00	1	1	1	1	1
Информация о редактировании									
Дата	Место	Содержание	Вид	Время	Состояние	Исполнитель	Исполнитель	Исполнитель	Исполнитель
14.09.2014	г. Москва	Редактирование маркера	1	14:00	1	1	1	1	1

Приложение А.4 – Диаграмма состояний

[illegible]

Приложение А.5 – Диаграмма данных



Приложение Б

(обязательное)

Экологическая карта Пензенской области

Листинг программы

					<i>ПГУ 1.090501.09.001 ПЗ</i>	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

index.php

```
<html lang="ru">
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Экологическая карта</title>

    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outl
ined:opsz,wght,FILL,GRAD@24,400,0,0" />
    <link rel="stylesheet" href="./libs/v6.1.1-dist/ol.css">
    <link rel="stylesheet" type="text/css" href="./styles.css">

</head>
<body>

    <nav class="topbar">
        <li style="float: left;"></li>
        <a href="./registr.php" class="regbtn" ><span
class="material-symbols-outlined">person</span>Вход/Регистрация</a>
    </nav>

    <div id="mySidepanel" class="sidepanel">
        <div id="filter" class="filter">

            <p>Фильтр меток</p>

            <b>Состояния</b>

            <label>
```

```

        <input type="checkbox" id="Bad"
name="interest"value="Bad">Нерешенные
    </label>

    <label>
        <input type="checkbox" id="Fine"
name="interest"value="Fine">В процессе устранения
    </label>

    <label>
        <input type="checkbox" id="Good"
name="interest"value="Good">Решенные
    </label>

    <button class="apply"
onclick="checkCheckBox()">Применить</button>
</div>

<div id="mySupport" class="support">
    <a href="./inforpage.php">Информация о сайте</a>
    <a href="./manualpage.php">Как использовать?</a>
</div>
</div>

<div id="popup" class="ol-popup">
    <a href="#" id="popup-closer" class="ol-popup-closer"></a>
    <div class="lcorner"></div>
    <div id="popup-content" class="popup_content"></div>
</div>

<div id="js-map" class="map"></div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

```

```

        <script src="./libs/v6.1.1-dist/ol.js"></script>
        <script src="./script/main.js"></script>

</body>
</html>

index2.php
<?php
require_once __DIR__ . '/src/helpers.php';

checkAuth();

$user = currentUser();
?>

<html lang="ru">
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Экологическая карта</title>

    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outl
ined:opsz,wght,FILL,GRAD@24,400,0,0" />
    <link rel="stylesheet" href="./libs/v6.1.1-dist/ol.css">
    <link rel="stylesheet" type="text/css" href="./styles.css">

</head>
<body>

    <nav class="topbar">

```

```

        <li style="float: left;"></li>

        <a href="./registr.php" class="regbtn" ><span
        class="material-symbols-outlined">person</span>Профиль</a>

    </nav>

    <div id="mySidepanel" class="sidepanel">
        <div id="filter" class="filter">

            <p>Фильтр меток</p>

            <b>Состояния</b>

            <label>
                <input type="checkbox" id="Bad"
                name="interest" value="Bad">Нерешенные
            </label>

            <label>
                <input type="checkbox" id="Fine"
                name="interest" value="Fine">В процессе устранения
            </label>

            <label>
                <input type="checkbox" id="Good"
                name="interest" value="Good">Решенные
            </label>

            <button class="apply"
            onclick="checkCheckBox()">Применить</button>
        </div>

        <div id="mySupport" class="support">
            <a>Информация о сайте</a>
            <a>Как использовать?</a>

```

```

        </div>
    </div>

    <div class="form_pop_up" id="form_pop_up">
        <div class="pop_up_container">
            <div class="pop_up_body" id="pop_up_body">
                <form action="/addnewmarker.php" method="post"
enctype="multipart/form-data">
                    <p>Создание маркера</p>

                    <select id="state" name="state" class="hideme">
                        <option value="Map_Marker_Bad"
selected>ужасно</option>
                        <option
value="Map_Marker_Fine">нормально</option>
                        <option
value="Map_Marker_Good">хорошо</option>
                    </select>

                    <label for="descrip"><b>Описание</b></label>
                    <input type="text" id="descrip" name="descrip" required>

                    <label for="coordX"><b>Координата X</b></label>
                    <input type="text" id="coordX" name="coordX" required>

                    <label for="coordY"><b>Координата Y</b></label>
                    <input type="text" id="coordY" name="coordY" required>

                    <button type="submit">Создать</button>
                    <button type="reset" class="btn cancel"
onclick="closeForm()">Закрыть</button>

                    <input type="text" id="user_id_fk"
name="user_id_fk" class="hideme" value="<? echo $user['id']; ?>"
required>

```

```

        </form>
    </div>
</div>
</div>

<div id="popup" class="ol-popup">
    <a href="#" id="popup-closer" class="ol-popup-closer"></a>
    <div id="popup-content" class="popup_content"></div>
</div>

<div id="js-map" class="map"></div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="./libs/v6.1.1-dist/ol.js"></script>
<script src="./script/main.js"></script>

</body>
</html>

```

login.php

```

<?php
require_once __DIR__ . '/src/helpers.php';

checkGuest();
?>

<!DOCTYPE html>
<html lang="ru" data-theme="light">
<?php include_once __DIR__ . '/components/head.php'?>
<body>

<form class="card" action="src/actions/login.php" method="post">
    <h2>Вход</h2>

```

					ПГУ 1.090501.09.001 ПЗ	Лист
						73
Изм.	Лист	№ докум.	Подпись	Дата		

```

    <?php if(hasMessage('error')): ?>
        <div class="notice error"><?php echo getMessage('error')
?></div>
    <?php endif; ?>

<label for="email">
    Имя
    <input
        type="text"
        id="email"
        name="email"
        placeholder="ivan@mail.ru"
        value="<?php echo old('email') ?>"
        <?php echo validationErrorAttr('email'); ?>
    >
    <?php if(hasValidationError('email')): ?>
        <small><?php echo validationErrorMessage('email');
?></small>
    <?php endif; ?>
</label>

<label for="password">
    Пароль
    <input
        type="password"
        id="password"
        name="password"
        placeholder="*****"
    >
</label>

<button
    type="submit"
    id="submit"

```

```

        >Продолжить</button>
    </form>

<p>У меня еще нет <a href="/register.php">аккаунта</a></p>

<?php include_once __DIR__ . '/components/scripts.php' ?>
</body>
</html>
<?php

require_once __DIR__ . '/../helpers.php';

$email = $_POST['email'] ?? null;
$password = $_POST['password'] ?? null;

if (empty($email) || !filter_var($email, FILTER_VALIDATE_EMAIL)) {
    setOldValue('email', $email);
    setValidationError('email', 'Неверный формат электронной
почты');
    setMessage('error', 'Ошибка валидации');
    redirect('/');
}

$user = findUser($email);

if (!$user) {
    setMessage('error', "Пользователь $email не найден");
    redirect('/login.php');
}

if (!password_verify($password, $user['password'])) {
    setMessage('error', 'Неверный пароль');
    redirect('/login.php');
}

```

```
$_SESSION['user']['id'] = $user['id'];
```

```
redirect('/index2.php');
```

registr.php

```
<?php
```

```
require_once __DIR__ . '/src/helpers.php';
```

```
checkGuest();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="ru" data-theme="light">
```

```
<?php include_once __DIR__ . '/components/head.php'?>
```

```
<body>
```

```
<form class="card" action="/src/actions/register.php" method="post"
enctype="multipart/form-data">
```

```
<h2>Регистрация</h2>
```

```
<label for="name">
```

```
Имя
```

```
<input
```

```
type="text"
```

```
id="name"
```

```
name="name"
```

```
placeholder="Иванов Иван"
```

```
value="<?php echo old('name') ?>"
```

```
<?php echo validationErrorAttr('name'); ?>
```

```
>
```

```
<?php if(hasValidationError('name')): ?>
```

```
<small><?php echo validationErrorMessage('name');
```

```
?></small>
```

```
<?php endif; ?>
```

```
</label>
```

					ПГУ 1.090501.09.001 ПЗ	Лист
						76
Изм.	Лист	№ докум.	Подпись	Дата		

```

<label for="email">
    E-mail
    <input
        type="text"
        id="email"
        name="email"
        placeholder="ivan@mail.ru"
        value="<?php echo old('email') ?>"
        <?php echo validationErrorAttr('email'); ?>
    >
    <?php if(hasValidationError('email')): ?>
        <small><?php echo validationErrorMessage('email');
?></small>
    <?php endif; ?>
</label>

<div class="grid">
    <label for="password">
        Пароль
        <input
            type="password"
            id="password"
            name="password"
            placeholder="*****"
            <?php echo validationErrorAttr('password'); ?>
        >
        <?php if(hasValidationError('password')): ?>
            <small><?php echo
validationErrorMessage('password'); ?></small>
        <?php endif; ?>
    </label>

    <label for="password_confirmation">
        Подтверждение
        <input

```

```

        type="password"
        id="password_confirmation"
        name="password_confirmation"
        placeholder="*****"
    >
</label>
</div>

<button
    type="submit"
    id="submit"
    >Продолжить</button>
</form>

<p>У меня уже есть <a href="/login.php">аккаунт</a></p>

<?php include_once __DIR__ . '/components/scripts.php' ?>
</body>
</html>
<?php

require_once __DIR__ . '/../helpers.php';

$name = $_POST['name'] ?? null;
$email = $_POST['email'] ?? null;
$password = $_POST['password'] ?? null;
$passwordConfirmation = $_POST['password_confirmation'] ?? null;

if (empty($name)) {
    setValidationError('name', 'Неверное имя');
}

if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    setValidationError('email', 'Указана неправильная почта');
}

```

```

if (empty($password)) {
    setValidationError('password', 'Пароль пустой');
}

if ($password !== $passwordConfirmation) {
    setValidationError('password', 'Пароли не совпадают');
}

if (!empty($_SESSION['validation'])) {
    setOldValue('name', $name);
    setOldValue('email', $email);
    redirect('/registr.php');
}

$pdo = getPDO();

$query = "INSERT INTO Users (name, email, password) VALUES (:name,
:email, :password)";

$params = [
    'name' => $name,
    'email' => $email,
    'password' => password_hash($password, PASSWORD_DEFAULT)
];

$stmt = $pdo->prepare($query);

try {
    $stmt->execute($params);
} catch (\Exception $e) {
    die($e->getMessage());
}

redirect('/login.php');

```

```

home.php
<?php
require_once __DIR__ . '/src/helpers.php';

$pdo = getPDO();

checkAuth();

$user = currentUser();
if ($user['role_id_fk'] === 1) {
    $data = $pdo->query("SELECT * FROM Markers");
}
else{
    $data = $pdo->prepare("SELECT * FROM markers WHERE user_id_fk =
:user_id_fk");
    $data->execute(['user_id_fk' => $user['id']]);
}
?>

<!DOCTYPE html>
<html lang="ru" data-theme="light">
    <title>Домашняя страница</title>
    <link rel="stylesheet" type="text/css" href="./styles.css">
<body>

    <div class="cardhome">
        <form action="src/actions/logout.php" method="post">
            <button role="button">Выйти из аккаунта</button>
        </form>
    </div>

    <div class="ptitle">
        <h1>Привет, <?php echo $user['name'] ?>!</h1>
    </div>

```

```

<?php
    echo "<table id=\"tableId\" class=\"table\"><tr><th>Номер
записи</th><th>Долгота</th><th>Широта</th><th>Описание</th><th>Состо
яние</th></tr>";

    while ($row = $data->fetch())
    {
        switch ($row["state"]) {
            case 'Map_Marker_Bad':{
                $row["state"] = 'Не решенный';
                break;
            }
            case 'Map_Marker_Fine':{
                $row["state"] = 'В процессе устранения';
                break;
            }
            case 'Map_Marker_Good':{
                $row["state"] = 'Решенный';
                break;
            }
        }
        echo "<tr>";
        echo "<td>" . $row["id"] . "</td>";
        echo "<td>" . $row["coordX"] . "</td>";
        echo "<td>" . $row["coordY"] . "</td>";
        echo "<td>" . $row["descrip"] . "</td>";
        echo "<td>" . $row["state"] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
?>

<div class="pos">
    <p><a href="./index2.php">вернуться</a> к добавлению
маркеров</p>

```

```

</div>

<div class="form_pop_up" id="form_pop_up">
  <div class="pop_up_container">
    <div class="pop_up_body" id="pop_up_body">
      <form action="/editmarker.php" method="post"
enctype="multipart/form-data">
        <p>Редактирование маркера</p>

        <label for="state">Укажите состояние</label>
        <select id="state" name="state">
          <option value="Map_Marker_Bad">Не
решено</option>
          <option value="Map_Marker_Fine">В процессе
устранения</option>
          <option
value="Map_Marker_Good">Решено</option>
        </select>

        <label for="descrip"><b>Описание</b></label>
        <input type="text" id="descrip" name="descrip"
required>

        <label for="coordX"><b>Координата X</b></label>
        <input type="text" id="coordX" name="coordX"
required>

        <label for="coordY"><b>Координата Y</b></label>
        <input type="text" id="coordY" name="coordY"
required>

        <button type="submit">Сохранить</button>
        <button type="reset" class="btn cancel"
onclick="closeForm()">Закрыть</button>

```

```

        <input type="text" id="id" name="id"
class="hideme" readonly>
        </form>
    </div>
</div>
</div>

<script src="./script/table.js"></script>

<?php include_once __DIR__ . '/components/scripts.php' ?>
</body>
</html>
<?php

```

```

require_once __DIR__ . '/../helpers.php';

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    logout();
}

```

```

redirect('/index.php');

```

infopage.php

```

<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Информация о сайте</title>
    <link rel="stylesheet" href="./infostyles.css">
</head>
<body>
    <h1>Информация</h1>

    <h2>Назначение</h2>

```

<p>Данный сайт предназначен для улучшения экологической обстановки в Пензе и области</p>

<h2>Что такое экологическая обстановка</h2>

<h2>Зачем улучшать экологическую обстановку?</h2>

<h2>Экологическое состояние Пензенской области на текущий момент</h2>

Вернуться назад
</body>
</html>

manualpage.php

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Правила пользования</title>

<link rel="stylesheet" href="./infostyles.css">

</head>

<body>

<h1>Информация</h1>

<h2>На что влияют мои действия?</h2>

<p>Ваша бдительность по отношению к окружающей Вас среде позволит Вам и окружающем Вас людям улучшить экологическую обстановку в нашем городе.</p>

<h2>Как добавить маркер?</h2>

<p>Чтобы добавить маркер Вам необходимо создать учетную запись пользователя на сайте. После чего Вы сможете авторизоваться и

добавлять маркеры по клику на карте. Кликнув по карте появится окно, необходимое к заполнению.</p>

<h2>Зачем мне регистрироваться?</h2>

<p>Регистрация необходима для того, чтобы исключить ложные маркеры.</p>

<h2>А если я добавил неверный маркер?</h2>

<p>В Вашем личном кабинете Вы можете отредактировать свои маркеры, если допустили ошибку при их создании.</p>

Вернуться назад

</body>

</html>

addnewmarker.php

<?

```
require_once __DIR__ . '/src/helpers.php';
```

```
const DB_HOST = 'localhost';
```

```
const DB_PORT = '3306';
```

```
const DB_NAME = 'EcoGIS';
```

```
const DB_USERNAME = 'root';
```

```
$coordX = $_POST['coordX'] ?? null;
```

```
$coordY = $_POST['coordY'] ?? null;
```

```
$descrip = $_POST['descrip'] ?? null;
```

```
$user_id_fk = $_POST['user_id_fk'] ?? null;
```

```
$state = $_POST['state'] ?? null;
```

```
$pdo = getPDO();
```

```
$query = "INSERT INTO Markers (coordX, coordY, descrip, user_id_fk, state) VALUES (:coordX, :coordY, :descrip, :user_id_fk, :state)";
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.09.001 ПЗ | Лист |
| | | | | | | 85 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

$params = [
    'coordX' => $coordX,
    'coordY' => $coordY,
    'descrip' => $descrip,
    'user_id_fk' => $user_id_fk,
    'state' => $state
];

$stmt = $pdo->prepare($query);

try {
    $stmt->execute($params);
} catch (\Exception $e) {
    die($e->getMessage());
}

redirect('/index2.php');

editmarker.php
<?

require_once __DIR__ . '/src/helpers.php';

const DB_HOST = 'localhost';
const DB_PORT = '3306';
const DB_NAME = 'EcoGIS';
const DB_USERNAME = 'root';

$id = $_POST['id'] ?? null;
$coordX = $_POST['coordX'] ?? null;
$coordY = $_POST['coordY'] ?? null;
$descrip = $_POST['descrip'] ?? null;
$state = $_POST['state'] ?? null;

```

```

$pdo = getPDO();

$query = "UPDATE Markers SET coordX = :coordX, coordY = :coordY,
descrip = :descrip, state = :state WHERE id = :id";

$params = [
    'id' => $id,
    'coordX' => $coordX,
    'coordY' => $coordY,
    'descrip' => $descrip,
    'state' => $state
];

$stmt = $pdo->prepare($query);

try {
    $stmt->execute($params);
} catch (\Exception $e) {
    die($e->getMessage());
}

redirect('/home.php');

getmarkers.php
<?

$servername = "localhost";
$username = "root";
$dbname = "ECOGIS";
$password = "";
$conn = mysqli_connect($servername, $username, $password, $dbname);

$sql = "SELECT * FROM Markers";
$result = $conn->query($sql);

```

```

$info = array();
while ($row = $result->fetch_assoc()) {
    $info [] = $row;
}
echo json_encode($info);

$conn->close();

```

config.php

```
<?php
```

```

const DB_HOST = 'localhost';
const DB_PORT = '3306';
const DB_NAME = 'EcoGIS';
const DB_USERNAME = 'root';

```

helpers.php

```
<?php
```

```
session_start();
```

```
require_once __DIR__ . '/config.php';
```

```
function redirect(string $path)
```

```

{
    header("Location: $path");
    die();
}

```

```
function setValidationError(string $fieldName, string $message):
```

```
void
```

```

{
    $_SESSION['validation'][$fieldName] = $message;
}

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.09.001 ПЗ | Лист |
| | | | | | | 88 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

function hasValidationError(string $fieldName): bool
{
    return isset($_SESSION['validation'][$fieldName]);
}

function validationErrorAttr(string $fieldName): string
{
    return isset($_SESSION['validation'][$fieldName]) ? 'aria-
invalid="true"' : '';
}

function validationErrorMessage(string $fieldName): string
{
    $message = $_SESSION['validation'][$fieldName] ?? '';
    unset($_SESSION['validation'][$fieldName]);
    return $message;
}

function setOldValue(string $key, mixed $value): void
{
    $_SESSION['old'][$key] = $value;
}

function old(string $key)
{
    $value = $_SESSION['old'][$key] ?? '';
    unset($_SESSION['old'][$key]);
    return $value;
}

function setMessage(string $key, string $message): void
{
    $_SESSION['message'][$key] = $message;
}

```

```

function hasMessage(string $key): bool
{
    return isset($_SESSION['message'][$key]);
}

function getMessage(string $key): string
{
    $message = $_SESSION['message'][$key] ?? '';
    unset($_SESSION['message'][$key]);
    return $message;
}

function getPDO(): PDO
{
    try {
        return new \PDO(dsn: 'mysql:host=' . DB_HOST . ';port=' .
DB_PORT . ';charset=utf8;dbname=' . DB_NAME, username: DB_USERNAME);
    } catch (\PDOException $e) {
        die("Connection error: {$e->getMessage()}");
    }
}

function findUser(string $email): array|bool
{
    $pdo = getPDO();

    $stmt = $pdo->prepare("SELECT * FROM users WHERE email =
:email");
    $stmt->execute(['email' => $email]);
    return $stmt->fetch(\PDO::FETCH_ASSOC);
}

function currentUser(): array|false
{

```

```

        $pdo = getPDO();

        if (!isset($_SESSION['user'])) {
            return false;
        }

        $userId = $_SESSION['user']['id'] ?? null;

        $stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
        $stmt->execute(['id' => $userId]);
        return $stmt->fetch(\PDO::FETCH_ASSOC);
    }

function logout(): void
{
    unset($_SESSION['user']['id']);
    redirect('/');
}

function checkAuth(): void
{
    if (!isset($_SESSION['user']['id'])) {
        redirect('/');
    }
}

function checkGuest(): void
{
    if (isset($_SESSION['user']['id'])) {
        redirect('/home.php');
    }
}

```

```

main.js
const popup = new ol.Overlay({
    element: document.getElementById('popup')
});

var isp = false;

var podsos = [];
const map = new ol.Map({
    view: new ol.View({
        center: ol.proj.fromLonLat([45.0046400, 53.2006600]),
        zoom: 12
    }),
    layers: [
        new ol.layer.Tile({
            source: new ol.source.OSM()
        })
    ],

    target: "js-map"
});

var markersBad = new ol.layer.Vector({
    source: new ol.source.Vector(),
    visible: true,
    title: 'MarkersBad'
});

var markersFine = new ol.layer.Vector({
    source: new ol.source.Vector(),
    visible: true,
    title: 'MarkersFine'
});

var markersGood = new ol.layer.Vector({

```

```

        source: new ol.source.Vector(),
        visible: true,
        title: 'MarkersGood'
    });

    let lon = 0;
    let lat = 0;

    class MARKERS_ON_MAP {
        constructor(str) {
            this.id = str.substring(8, str.indexOf('\\"', 8));
            str = str.substring(str.indexOf(", ", 2)+1);
            this.coordX = str.substring(10, str.indexOf('\\"', 10));
            str = str.substring(str.indexOf(", ")+1);
            this.coordY = str.substring(10, str.indexOf('\\"', 10));
            str = str.substring(str.indexOf(", ")+1);
            this.descrip = str.substring(11, str.indexOf('\\"', 11));
            str = str.substring(str.indexOf(", ")+1);
            this.user_fk_id = str.substring(14, str.indexOf('\\"', 14));
            str = str.substring(str.indexOf(", ")+1);
            this.state = str.substring(9, str.indexOf('\\"', 9));
        }

        createMarker() {
            var style = new ol.style.Style({
                image: new ol.style.Icon({
                    anchor: [0.5, 1],
                    src: './images/' + this.state + '.png',
                    scale: 0.1
                })
            });

            this.marker = new ol.Feature(new
            ol.geom.Point(ol.proj.fromLonLat([this.coordX, this.coordY])), {
                });

```

```

        this.marker.setStyle(style);
        return this.marker;
    }

    getMarker() {
        return this.marker;
    }

    setId(id) {
        this.id = id;
    }

    setcoordX(coordX) {
        this.coordX = coordX;
    }

    setcoordY(coordY) {
        this.coordY = coordY;
    }

    setDescrip(descrip) {
        this.descrip = descrip;
    }

    setUser_fk_id(user_fk_id) {
        this.user_fk_id = user_fk_id;
    }

    setState(state) {
        this.state = state;
    }

    getcoordX() {
        return this.coordX;
    }

```

```

getcoordY(){
    return this.coordY;
}

getState(){
    return this.state;
}

getDescrip(){
    return this.descrip;
}
}

map.on('click', function (e) {
    let f = map.forEachFeatureAtPixel(e.pixel, function(feature) {
        return feature;
    });
    let r = true;
    for (let i = 0; i < podsos.length; i++) {
        if (podsos[i].getMarker() === f) {
            const content = f.getGeometry().getCoordinates();
            popup.setPosition(content);
            map.addOverlay(popup);
            let st;
            switch (podsos[i].getState()) {
                case 'Map_Marker_Bad':{
                    st = 'Не решено';
                    break;
                }
                case 'Map_Marker_Fine':{
                    st = 'В процессе устранения';
                    break;
                }
                case 'Map_Marker_Good':{

```

```

        st = 'Решено';
        break;
    }
}

document.getElementById('popup-content').innerHTML = `
    <p>Описание: ${podsos[i].getDescrip()}</p>
    <p>Состояние: ${st}</p>`;
document.querySelector('#popup-
closer').addEventListener('click', function () {
    map.removeOverlay(popup);
});
isp = true;
r = false;
}
}

if (r) {
    if(isp)
    {
        map.removeOverlay(popup);
        isp = false;
    }
    else{
        var lonlat = ol.proj.transform(e.coordinate, 'EPSG:3857',
'EPSG:4326');
        lon = lonlat[0];
        lat = lonlat[1];
        document.getElementById("form_pop_up").style.display =
"block";
        document.getElementById('coordX').value = lon;
        document.getElementById('coordY').value = lat;
    }
}
})

```

```

let u = 0;

while (podsos == undefined || podsos == null) {
    u++;
}

$.ajax({
    url: '/getmarkers.php',
    method: 'POST',
    dataType: 'JSON',
    success: function(data){
        let response = JSON.stringify(data);
        let i = 0;

        while (response.indexOf("{}") != -1) {
            podsos[i] = new MARKERS_ON_MAP(response.substring(0,
response.indexOf("{}")));
            response = response.substring(response.indexOf("{}")+1);
            switch(podsos[i].getState()){
                case 'Map_Marker_Bad': {

(markersBad.getSource().addFeature(podsos[i].createMarker()));
                    break;
                }
                case 'Map_Marker_Fine': {

(markersFine.getSource().addFeature(podsos[i].createMarker()));
                    break;
                }
                case 'Map_Marker_Good': {

(markersGood.getSource().addFeature(podsos[i].createMarker()));
                    break;
                }
            }
        }
    }
});

```

```

        }
        i++;
    }
}
}))

function checkCheckBox() {
    map.removeLayer(markersBad);
    map.removeLayer(markersFine);
    map.removeLayer(markersGood);

    let checkbox1 = document.getElementById('Bad');
    let checkbox2 = document.getElementById('Fine');
    let checkbox3 = document.getElementById('Good');

    if (checkbox1.checked) {
        map.addLayer(markersBad);
    }

    if (checkbox2.checked) {
        map.addLayer(markersFine);
    }

    if (checkbox3.checked) {
        map.addLayer(markersGood);
    }
}

function closeForm() {
    document.getElementById("form_pop_up").style.display = "none";
}

function createPopup(feature) {
    let r = 0;
    for (let i = 0; i < podsos.length; i++) {
        if (podsos[i].getMarker() === feature) {

```

```

        console.log(podsos[i].getDescrip());
        r = i;
    }
}

feature.popup = new ol.Popup("pop",
    new ol.geom.Point(ol.proj.fromLonLat([this.coordX,
this.coordY])),
    null,
    '<div
class="markerContent">'+podsos[r].getDescrip()+ '</div>',
    null,
    true,
    function() { controls['selector'].unselectAll(); }
);

map.addPopup(feature.popup);
}

function destroyPopup(feature) {
    feature.popup.destroy();
    feature.popup = null;
}

table.js
const table = document.getElementById('tableId');

const rows = table.getElementsByTagName('tr');

Array.from(rows).forEach((row, index) => {
    console.log(index);

    row.addEventListener('click', () => {
        const cells = row.getElementsByTagName('td');
        document.getElementById("form_pop_up").style.display =
"block";
    });
});

```

```

        document.getElementById('id').value = cells[0].innerHTML;
        document.getElementById('coordX').value =
cells[1].innerHTML;
        document.getElementById('coordY').value =
cells[2].innerHTML;
        document.getElementById('descrip').value =
cells[3].innerHTML;
        if (cells[4].innerHTML == 'Map_Marker_Bad') {

document.getElementById('state').getElementsByName('option')[0].s
elected = 'selected'
        } else {
            if (cells[4].innerHTML == 'Map_Marker_Fine') {

document.getElementById('state').getElementsByName('option')[1].s
elected = 'selected'
            } else {

document.getElementById('state').getElementsByName('option')[2].s
elected = 'selected'
            }
        }
    });
});

```

```

function closeForm() {
    document.getElementById("form_pop_up").style.display = "none";
}

```

styles.css

```

* {
    box-sizing: border-box;
    font-size: 25px;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Arial, sans-serif;

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | ПГУ 1.090501.09.001 ПЗ | Лист |
| | | | | | | 100 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

}

body {
    margin: 0 auto;
    padding: 0;
}

.map {
    padding-top: 48px;
    height: 100%;
    width: 100%;
}

li {
    list-style: none;
}

.pos p{
    position: relative;
    text-align: center;
    text-decoration: none;
}

.pos a{
    color: #27c2ba;
    text-decoration: none;
}

.pos a:hover{
    text-decoration:underline;
}

.topbar {
    width: 100%;
    height: 48px;
    position: absolute;
    z-index: 1;
}

```

```

        background-color:#4CAF50;
    }

.topbar a {
    width: 300px;
    position: absolute;
    right: 0;
    padding: 9px 7px;
    text-decoration: none;
    font-size: 25px;
    color: #1f1302;
    display: block;
    text-align: center;
}

.topbar a:hover {
    background-color: #1f1302;
    color:#4CAF50;
}

.sidepanel {
    height: 100%;
    width: 300px;
    position: fixed;
    top: 48px;
    right: 0;
    z-index: 1;
    background-color: #2b822e;
}

.sidepanel a {
    display: block;
    padding: 9px, 7px;
    color: #02300e;
}

```

```

.sidepanel p {
    text-align: center;
    font-size: 30px;
    color: #1e0702;
}

.sidepanel b {
    padding-left: 20px;
    color: #1e0702;
}

.sidepanel label {
    display: block;
    left: 0;
}

.sidepanel button {
    cursor: pointer;
    display: block;
    width: 230px;
    margin: 10px auto 0px auto;
    padding: 20px 0px;
    font-weight: 500;
    border: none;
    color: #1d3b20;
    border-radius: 16px;
    background-color: #4CAF50;
}

.support {
    padding-top: 22px;
}

.support a {

```

```

        cursor: pointer;
        width: 300px;
        position: relative;
        right: 0;
        padding: 9px 7px;
        text-decoration: none;
        display: block;
        color: #2c1806;
        text-align: left;
    }

.support a:hover {
    background-color: #1f1302;
    color: #4CAF50;
}

.form_pop_up {
    display: none;
    width: 100%;
    height: 100%;
    position: fixed;
    top: 0;
    left: 0;
    background-color: rgba(0, 0, 0, 0.8);
    z-index: 2;
}

.pop_up_container {
    width: 100%;
    height: 100%;
}

.pop_up_body {
    margin: auto;
    width: 500px;

```

```

        background-color: white;
        border-radius: 16px;
        text-align: center;
        padding: 10px 15px 10px 15px;
        position: relative;
    }

    .pop_up_body label {
        display: block;
    }

    .pop_up_body select {
        width: 100%;
        padding: 15px;
        margin: 5px 0 22px 0;
    }

    .pop_up_body p {
        margin: 25px auto 0px auto;
        width: 330px;
        padding: 17px 20px;
        border-radius: 16px;
        border: none;
        font-weight: 500;
        color: gray;
    }

    input[type=text] {
        width: 100%;
        padding: 15px;
        margin: 5px 0 22px 0;
        display: inline-block;
        border: none;
        background: #f1f1f1;
    }

```

```
.pop_up_body button {
    cursor: pointer;
    display:inline;
    width: 150px;
    margin: 30px auto 0px auto;
    padding: 20px 0px;
    border: none;
    color: yellow;
    border-radius: 16px;
    background-color:#4CAF50;
}
```

```
.pop_up_close {
    position: absolute;
    top: 15px;
    right: 15px;
    font-size: 21px;
}
```

```
.hideme {
    display: none;
    visibility: hidden;
}
```

```
.cardhome button{
    cursor: pointer;
    position: absolute;
    top: 0;
    right: 0;
}
```

```
.ptitle h1 {
    position: relative;
    text-align: center;
}
```

```

}

.table {
    width: 100%;
    margin-bottom: 20px;
    border: 1px solid #dddddd;
    border-collapse: collapse;
}

.table th {
    font-weight: bold;
    padding: 5px;
    background: #efefef;
    border: 1px solid #dddddd;
}

.table td {
    border: 1px solid #dddddd;
    padding: 5px;
}

.popup_content {
    padding-top: 1px;
    padding-bottom: 1px;
    padding-left: 10px;
    padding-right: 10px;
    width: auto;
    font-size: 10px;
    color: #2c1806;
    background-color: #36ae6e;
    border-radius: 16px;
}

.lcorner {
    position: relative;
    width: 25px;
    height: 20px;

```

```

    top: 20px;
    left: 0;
    background-color: #2b822e;
    border-radius: 3px;
}

infostyles.css
* {
    font-size: 25px;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Arial, sans-serif;
}

body {
    background: url("../images/back.png") no-repeat;
    background-position: right bottom;
}

body h1 {
    font-size: 50px;
    margin-left: 1cm;
    color: #1d3b20;
}

body h2 {
    font-size: 30px;
    margin-left: 1cm;
    color: #256e2b;
}

body p {
    margin-left: 0.5cm;
}

```

```

app.css
html[data-theme="light"] .card {
    background: #fff;
}

body {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    height: 100vh;
}

.card {
    box-shadow: 0.0145rem 0.029rem 0.174rem rgba(0, 0, 0, 0.01698),
    0.0335rem 0.067rem 0.402rem rgba(0, 0, 0, 0.024),
    0.0625rem 0.125rem 0.75rem rgba(0, 0, 0, 0.03),
    0.1125rem 0.225rem 1.35rem rgba(0, 0, 0, 0.036),
    0.2085rem 0.417rem 2.502rem rgba(0, 0, 0, 0.04302),
    0.5rem 1rem 6rem rgba(0, 0, 0, 0.06),
    0 0 0 0.0625rem rgba(0, 0, 0, 0.015);
    background: #141e26;
    border-radius: 0.25rem;
    padding: 40px;
}

form {
    min-width: 500px;
    max-width: 700px;
}

.avatar {
    width: 300px;
    height: 300px;
    display: block;
    object-fit: cover;
    border-radius: 50%;
}

.home {
    display: flex;
    flex-direction: column;
    align-items: center;
    row-gap: 20px;
    padding: 100px;
}

.notice {
    padding: 10px;
    border-radius: 6px;
    margin: 20px 0;
}

```

```
.notice.error {
    border: 2px solid #762c2c;
    background: #f3d3d3;
    color: #762c2c;
}

.notice.success {
    border: 2px solid #2c742b;
    background: #e1fee3;
    color: #2c742b;
}
```

| | | | | | | |
|------|------|----------|---------|------|-------------------------------|------|
| | | | | | <i>ПГУ 1.090501.09.001 ПЗ</i> | Лист |
| | | | | | | 110 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Приложение В

(обязательное)

Экологическая карта Пензенской области

Презентация

| | | | | | | |
|------|------|----------|---------|------|-------------------------------|------|
| | | | | | <i>ПГУ 1.090501.09.001 ПЗ</i> | Лист |
| | | | | | | 111 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

Экологическая карта Пензенской области

Выполнил студент группы 19в0э1 Прокоданов В.В.

Научный руководитель: к.т.н., доцент кафедры ВТ Слепцов Н.В.

ЦЕЛЬ РАБОТЫ

Разработать экологическую карту для мониторинга состояния окружающей среды. Мониторинг осуществляется за счет сбора и хранения информации, которую пользователи могут внести на сайт.



| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.09.001 ПЗ

Лист

112

АКТУАЛЬНОСТЬ ПРОЕКТА

По состоянию на 2024 год Россия находится на 112 месте из 180 в списке стран с лучшей экологией.

Для улучшения экологической обстановки в стране, в частности в Пензенской области, было принято решение создать экологическую ГИС, в которой была бы возможность отслеживать и влиять на состояние природы в нашем субъекте Российской Федерации.

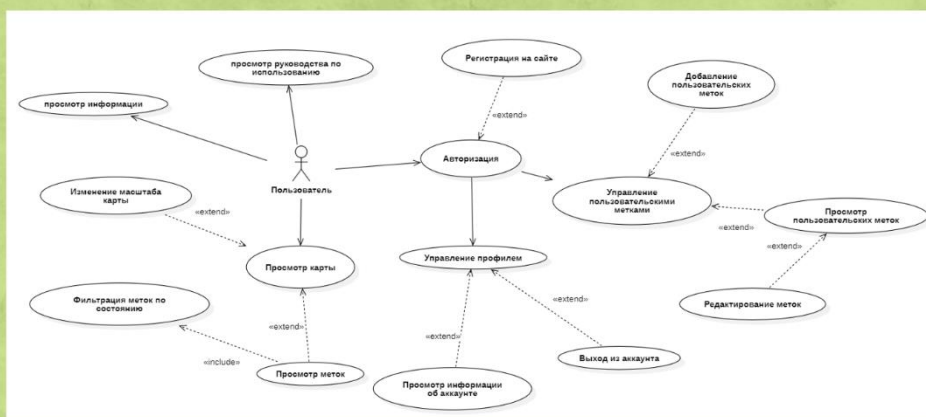
СРЕДСТВА РАЗРАБОТКИ

- JavaScript
- PHP
- Open Server
- MySQL
- OpenLayers

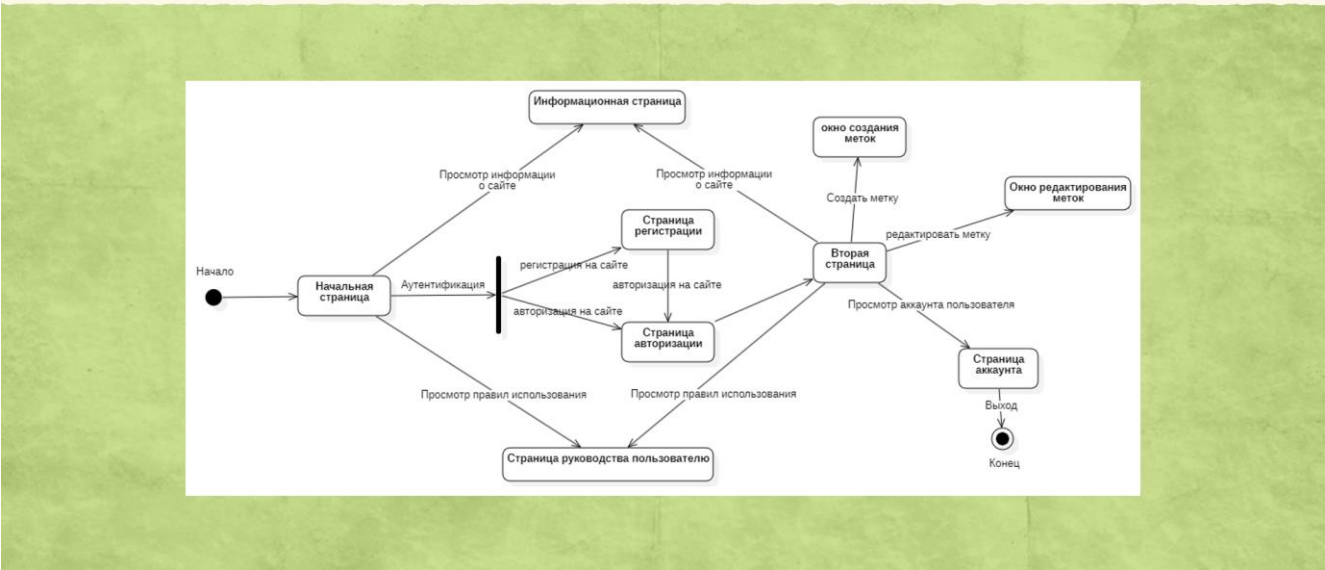


АРХИТЕКТУРА ПРИЛОЖЕНИЯ

UML-ДИАГРАММА ПРЕЦЕДЕНТОВ



UML-ДИАГРАММА СОСТОЯНИЙ



АВТОРИЗАЦИЯ И РЕГИСТРАЦИЯ

Вход

Имя

Пароль

Продолжить

У меня еще нет аккаунта

Регистрация

Имя

E-mail

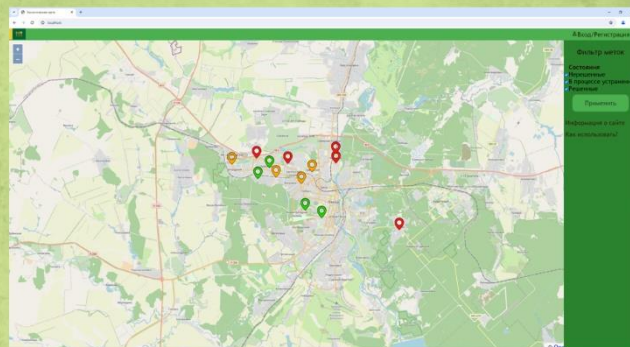
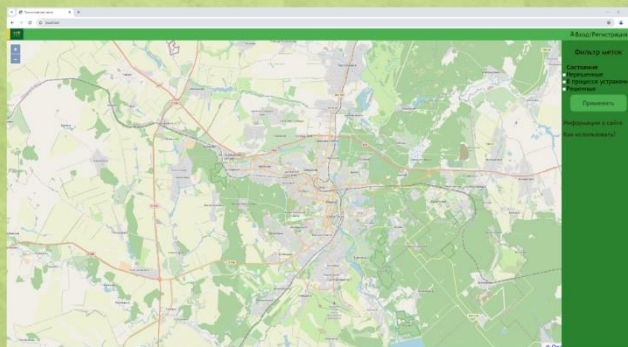
Пароль

Подтверждение

Продолжить

У меня уже есть аккаунт

ОСНОВНАЯ СТРАНИЦА ПРИЛОЖЕНИЯ



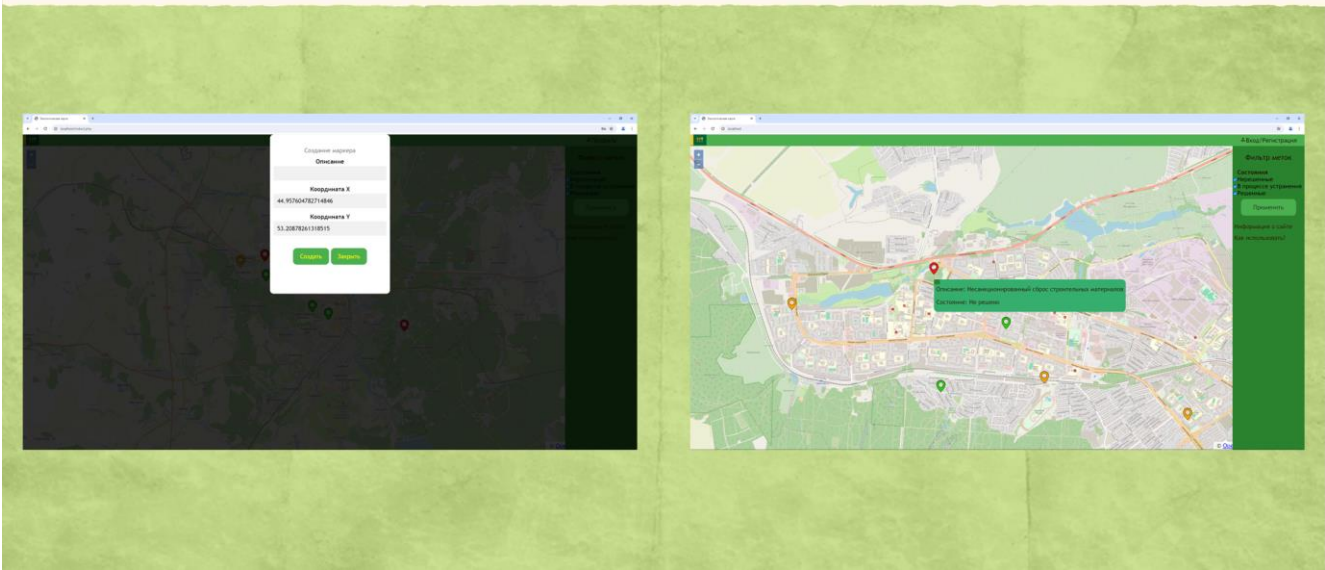
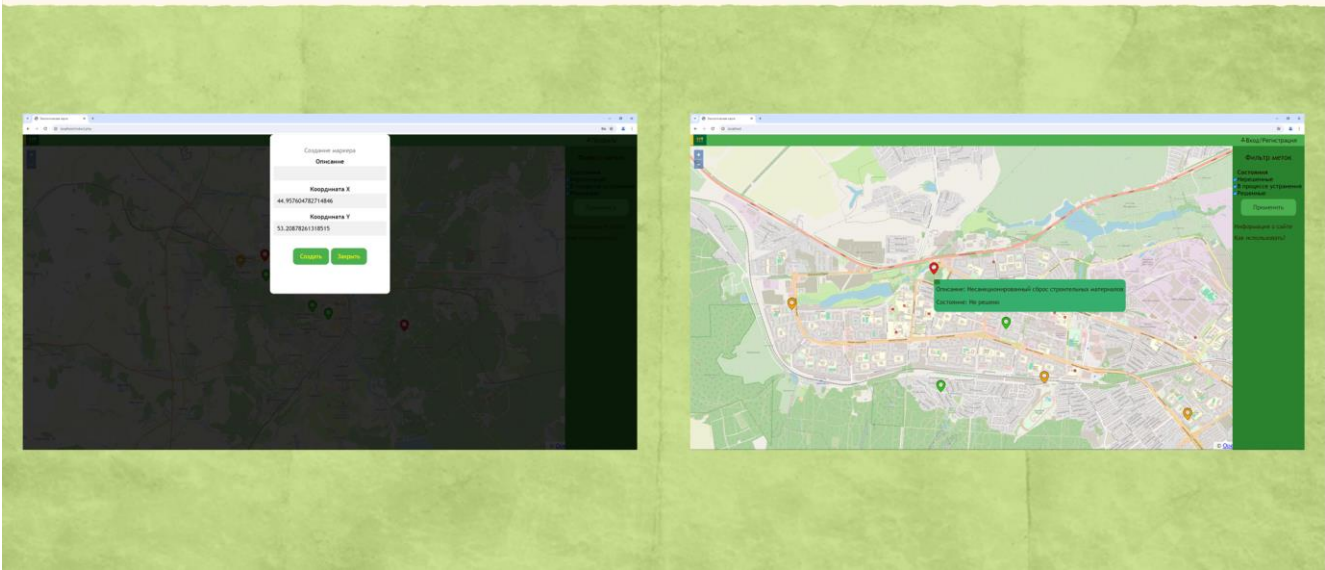
| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.09.001 ПЗ

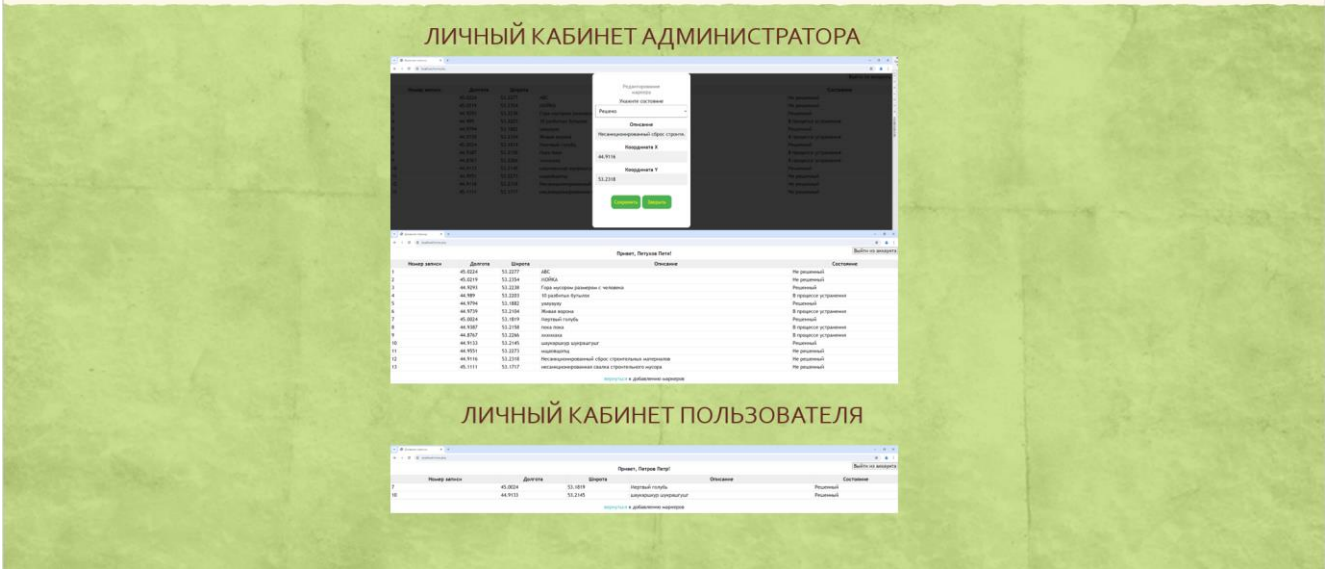
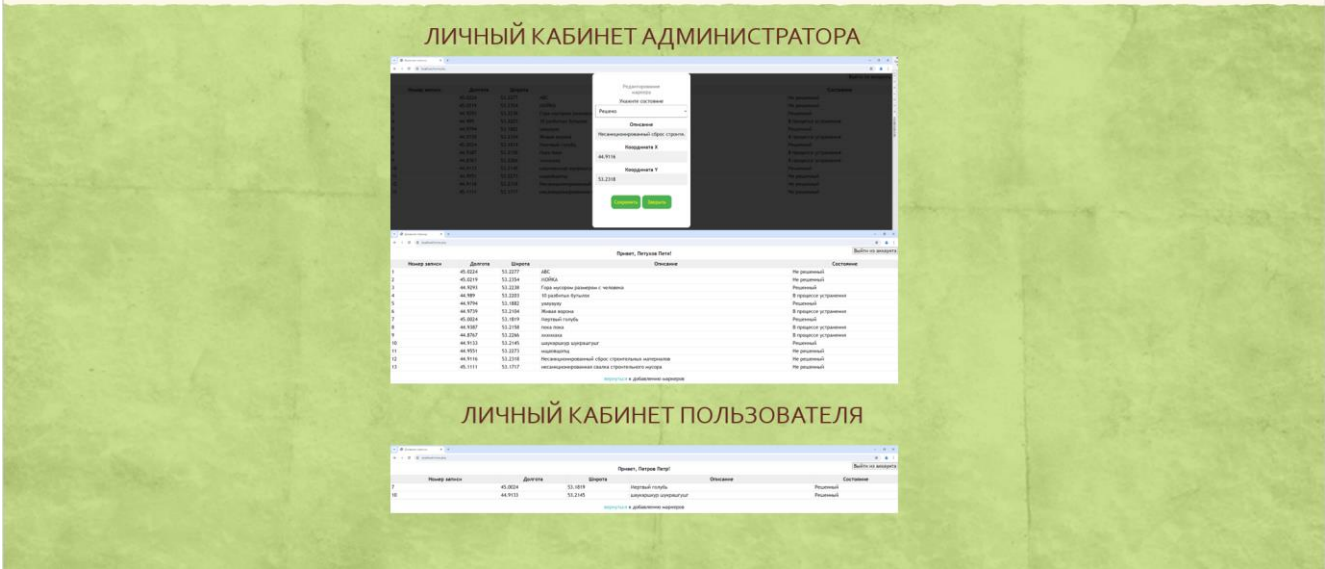
Лист

116

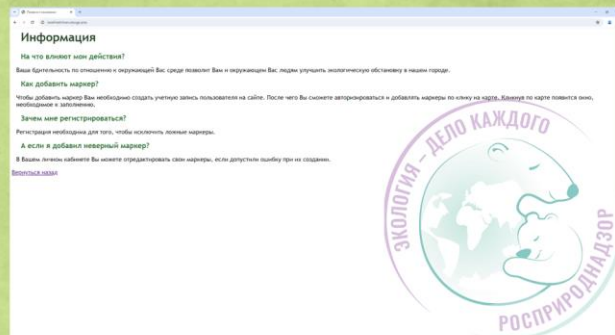
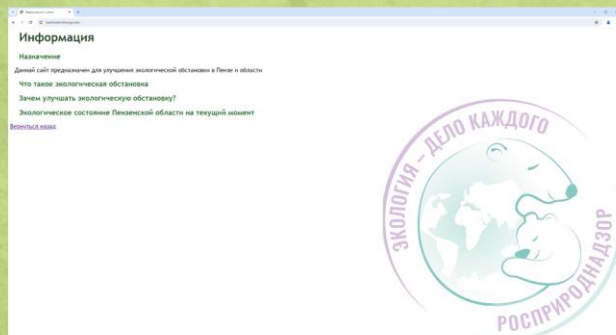
ДОБАВЛЕНИЕ МЕТОК



ЛИЧНЫЕ КАБИНЕТЫ АДМИНИСТРАТОРА И ПОЛЬЗОВАТЕЛЯ

[illegible][illegible]

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ О ПРИЛОЖЕНИИ



ЗАКЛЮЧЕНИЕ

Развитие проекта

Добавление нейросети

Прямое привлечение необходимых служб

Расширение зон мониторинга:

- заповедные зоны
- водные объекты
- объекты, загрязняющие воздушное пространство



| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПГУ 1.090501.09.001 ПЗ

Лист

118

СПАСИБО ЗА ВНИМАНИЕ!