УДК 004.415.2

Разработка веб-портала оценки и прогноза пожарной опасности по условиям погоды

Бондаренко Владислав Витальевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Научный руководитель:
Глаголев Владимир Александрович
Приамурский государственный университет имени Шолом-Алейхема
К.г.н., доцент кафедры информационных систем, математики и правовой информатики

Аннотация
В связи с большими массивами лесов и слабой заселенностью Дальнего Востока проблема с обнаружением пожаров является на сегодняшний день актуальной. Для решения данной проблемы создаются меры по предупреждению и предотвращению лесных пожаров. В данной статье рассматривается разработка веб-портала оценки и прогноза пожарной опасности по условиям погоды с использованием современных технологий и средств.
Ключевые слова: геоинформационная система, веб-портал, пожарная опасность, комплексный показатель, шкала Нестерова, Leaflet, C#, ASP.NET Core

Development of a web portal for assessing and predicting a fire danger via weather conditions

Bondarenko Vladislav Vitalievich
Sholom-Aleichem Priamursky State University
Student

Scientific adviser:
Glagolev Vladimir Aleksandrovich
Sholom-Aleichem Priamursky State University
Assistant professor of the Department of Computer Science

Abstract
In reason of large amount of forests and low population of the Far East, fire detection is an actual problem today. To solve this problem, measures are being taken to prevent forest fires. This article discusses the development of a web portal for assessing and predicting a fire danger via weather conditions using modern technologies and tools.

Keywords: geographic information system, web portal, fire danger, complex indicator, Nesterov scale, Leaflet, C#, ASP.NET Core

Лесной пожар является стихийным и неконтролируемым распространением огня по лесным площадям. Лесные пожары наносят огромный ущерб природе – сгорают леса, разрушается среда обитания многих живых организмов, умирают животные, уничтожаются ценные ресурсы. Помимо этого, большой ущерб наносится человеку и в первую очередь из-за загрязнения воздуха. Так существуют научные исследования, которые показывают, как вероятность возникновения патологий внутренних органов возрастает с ростом объема мелких частиц в воздухе, которые образуются во время пожаров [1, 2].

Регионы Сибири и Дальнего Востока находятся в особой зоне риска в связи с большими запасами древесины и их активной разработкой, а также малой заселенностью людьми местных территорий.

В последнее время применение геоинформационных систем (ГИС) в различных сферах жизни человека возрастает. Связано это с тем, что ГИС включает в себя современные методы обработки информации и позволяет визуализировать огромные массивы данных в виде географических карт. Это делает информацию более доступной для понимания и анализа. В связи с этим тема разработки ГИС интересует многих российских исследователей. Так, например, существуют научные работы, посвященные технологиям разработки геоинформационных систем [3-6]. Также данная тема интересует зарубежных исследователей. Некоторые из них в своих работах приводят результаты создания ГИС для мониторинга природных явлений (лесные пожары, наводнения) [7-9]. Существует исследовательский проект по созданию ГИС в городской среде [10]. Другие исследователи приложили усилия при разработке архитектуры масштабируемых ГИС [11, 12]. Также имеется исследование по разработке архитектуры базы данных [13]. Таким образом представляется возможным создание геоинформационной системы, которая будет визуализировать карту пожарной опасности Еврейской автономной области.

Существуют различные способы для обнаружения лесных пожаров (наблюдение с пожарных вышек, наблюдение пешим ходом, авиационное наблюдение с воздуха, наблюдение из космоса), но все они либо неэффективные, либо сложные и дорогостоящие. Есть способы куда более простые и экономные, основанные на применении математического аппарата. Так как основой для возникновения пожаров являются погодные условия, то можно на основании данных с метеостанций с помощью формул рассчитать вероятность возникновения пожаров в заданных регионах.

В соответствии с указаниями по противопожарной профилактике в лесах и регламентации работы лесопожарных служб, утвержденных Приказом Федеральной службы России № 289 в разделе «Регламентация лесопожарных служб» предусматривается вычисление комплексного

показателя пожарной опасности по условиям погоды [14]. Одним из способов для расчета данного показателя является метод Нестерова [15].

Расчет комплексного показателя пожарной опасности производится с использованием двух метеорологических характеристик – температура воздуха и точка росы – по следующей формуле (1):

$$КП = \sum_{1}^{n} T \times (T - t) \quad (1)$$

Суть метода состоит в накоплении комплексного показателя за период, когда отсутствуют дожди. Когда накапливается определенное количество осадков, показатель сбрасывается до нуля. Расчет производится от первого числа месяца до интересующего нас числа этого же месяца. Полученный результат сопоставляется с классом пожарной опасности по таблице Нестерова (табл. 1).

Таблица 1 – Классы пожарной опасности в лесах по условиям погоды

Класс пожарной опасности	**Величина показателя пожарной опасности (КП), в ℃**	**Степень пожарной опасности**
I	0 – 300	Очень малая
II	301 – 1000	Малая
III	1001 – 4000	Средняя
IV	4001 – 12000	Высокая
V	>12000	Чрезвычайная

Собрав метеорологические данные (температура, точка росы, количество осадков), можно высчитать значения комплексного показателя на определенных метеостанциях: Екатерино-Никольское, Облучье, Биробиджан, Ленинское, Смидович, Хабаровск, Урми, Архара, Тырма. Многоугольник, построенный по точкам в местах расположения данных метеостанций, покрывает почти всю территорию Еврейской автономной области (рис. 1).

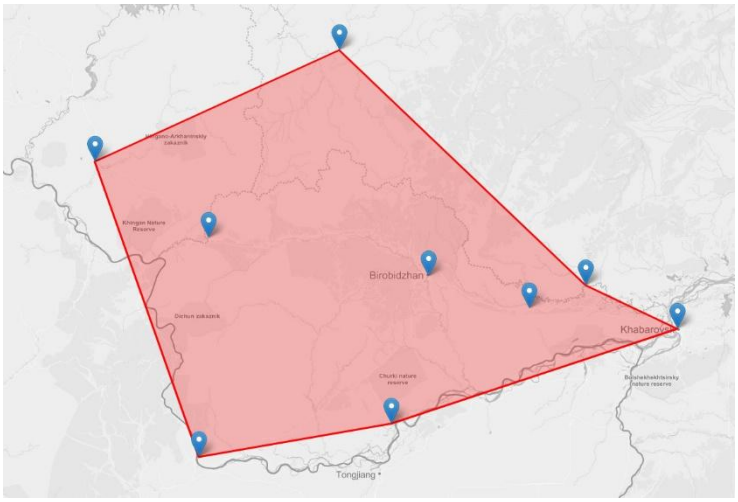

Рисунок 1 – Область покрытия метеостанциями территории Еврейской автономной области

Для того, чтобы узнать уровень пожарной опасности в любой точке на карте, необходимо вычислить промежуточные значения с помощью интерполяции методом обратно взвешенных расстояний (IDW) [16].

Для разработки интернет-портала было решено остановить свой выбор на технологии ASP.NET Core. ASP.NET Core является бесплатным фреймворком с открытым исходным кодом для разработки высокопроизводительных веб-приложений на языке программирования C#. Он имеет в своем составе перечень всех необходимых инструментов для быстрой разработки и поставляется со встроенным веб-сервером Kestrel. База данных будет создана в высокопроизводительной и открытой СУБД MariaDB. Для связи приложения с базой данных будет использоваться MySQLConnector в связке с Dapper – микро-ORM технология для доступа к данным с приведением типов. По некоторым сравнениям производительности Dapper быстрее Entity Framework более чем в 10 раз [17-19].

Для генерирования и отображения географической карты будет использоваться JavaScript библиотека Leaflet. Несмотря на то, что существуют картографические продукты от таких гигантов, как Google и Яндекс, в результате тестирования API данных продуктов для отрисовки большого количества полигонов было выявлено, что Google Maps и Яндекс Карты могут отображать только около 60-100 полигонов за раз. В таком случае пришлось бы потратить дополнительные силы для реализации метода рисования кластерами. Гораздо более легким в использовании оказался API от Leaflet [20], который изначально разрабатывался с учётом высокой производительности и легковесности.

Для того, чтобы понять из каких физических и программных компонентов будет состоять система и как они будут взаимодействовать между собой, была построена следующая диаграмма (рис. 2).

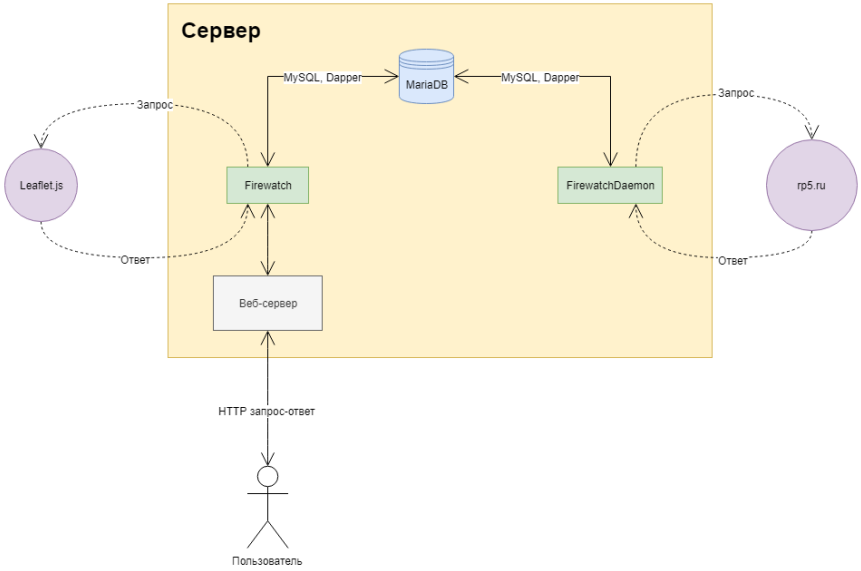

Рисунок 2 – Диаграмма, показывающая основные компоненты системы и их взаимодействие

В общем случае работа веб-портала выглядит следующим образом: сначала пользователь, зашедший на сайт, отправляет запрос на генерацию новой карты по выбранной дате. Далее этот HTTP запрос направляется на веб-сервер, который обрабатывает и передает его основному модулю системы Firewatch. Получив все необходимые данные из этого запроса, приложение обращается к базе с целью получить данные по метеостанциям, погодным условиям и географическим координатам полигонов. Когда данные имеются, веб-приложение отправляет запрос к модулю Leaflet.js на формирование новой карты и получает ответ. Готовая карта отправляется через веб-сервер обратно клиенту, где уже отображается средствами браузера.

Помимо основного приложения Firewatch, также имеется вспомогательное приложение FirewatchDaemon, которое отвечает за автоматическую актуализацию метеорологических данных. Для этого производится запрос к сайту rp5.ru, содержащему данные метеорологических измерений по каждой из метеостанций.

Далее необходимо определиться со структурой сайта. Это поможет лучше представить то, как должен выглядеть веб-портал и как пользователь будет взаимодействовать с ним. Для решения данной задачи была построена следующая диаграмма (рис. 3).

Рисунок 3 – Диаграмма, изображающая схему веб-портала

В результате были определены основные веб-страницы, на которых может находиться пользователь, и переходы между ними. Следующим шагом стало определение того, как пользователи могут взаимодействовать с веб-порталом и как разные функциональные возможности распределяются между ними. В данной системе предполагается разделение прав доступа к определенным секторам сайта, и не каждый пользователь имеет полный доступ ко всем возможностям. Для иллюстрации этого была создана диаграмма вариантов использования (рис. 4).

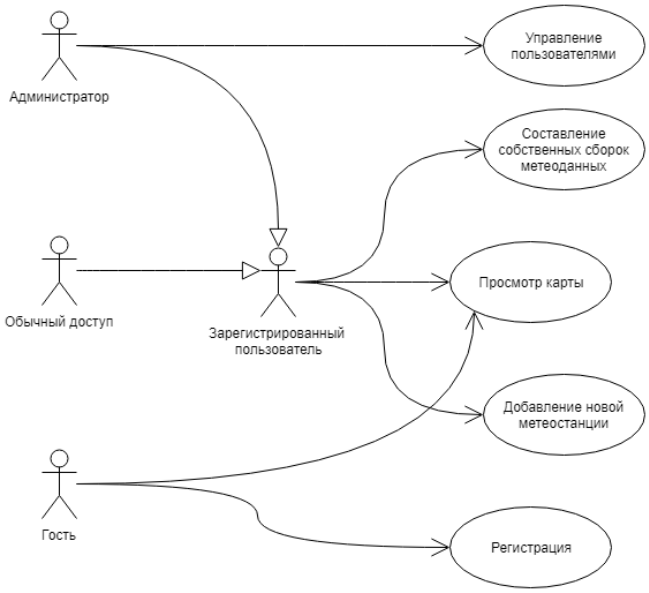

Рисунок 4 – Диаграмма вариантов использования, описывающая распределение прав доступа к системе

Главное отличие зарегистрированного пользователя от гостя – возможность создавать свои собственные сборки метеорологических данных и использовать их для генерации карты.

Следующим важным этапом стало проектирование классов, которое позволит заранее определиться со свойствами сущностей системы и взаимосвязями между ними. В результате проектирования основных сущностей, предназначенных для хранения данных, была построена следующая диаграмма (рис. 5).

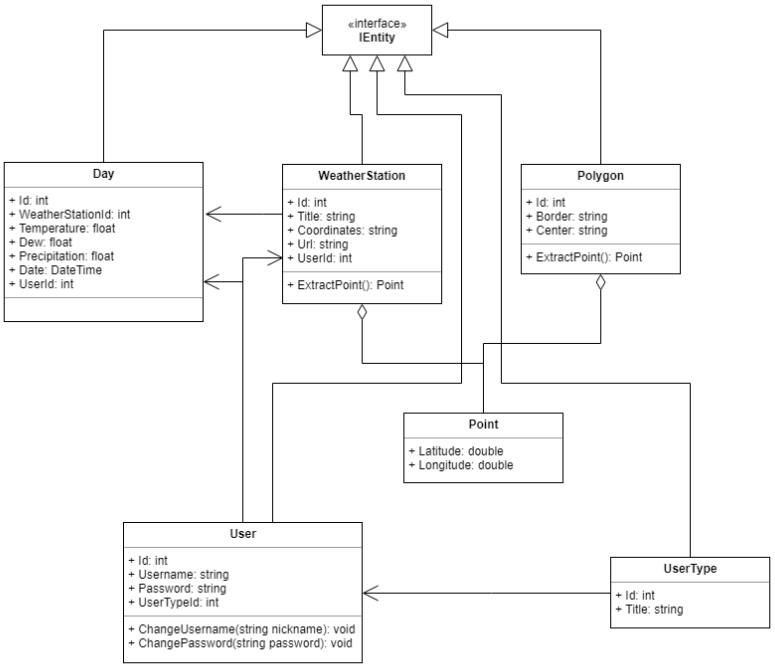

Рисунок 5 – Диаграмма классов, описывающая основные сущности системы

Все классы, изображенные на диаграмме, кроме Point, наследуются от интерфейса IEntity. Данный интерфейс реализует шаблон проектирования «Интерфейс-маркер». Его цель пометить все классы, которые наследуются от него, и таким образом сгруппировать их.

Класс WeatherStation представляет конкретный экземпляр метеорологической станции, а Polygon – полигона, отрисовываемого на карте. Оба класса имеют публичный метод ExtractPoint(), который извлекает точку месторасположения станции или полигона и возвращает ее в виде структуры Point. Структура Point содержит два поля: широта и долгота.

Для реализации системы идентификации пользователей были созданы два класса: UserType и User. Класс UserType назначает права доступа, определенные в диаграмме вариантов использования. Класс User хранит информацию о пользователе: имя, пароль и тип доступа.

Класс Day олицетворяет один из многих дней по которому ведется сбор и хранение метеорологических данных таких, как температура, точка росы, количество осадков. Для определения к какой метеостанции относятся данные, а также кто является их автором, существуют поля WeatherStationId и UserId.

При проектировании проекта было решено использовать такой шаблон проектирования, как Data Access Object (DAO), который предполагает создание отдельных сущностей, реализующих операции CRUD (Create - создание, Read - чтение, Update - обновление, Delete - удаление). Данные операции более подробно рассматриваются в научном исследовании, посвященном проектированию CRUD системы для редактирования геоинформационных данных [21]. Также был использован шаблон проектирования Singleton для того, чтобы быть уверенным, что в процессе работы программы будет создано и использовано одно единственное подключение к базе данных. Для каждой из сущностей, представленных на предыдущей диаграмме, был создан отдельный класс DAO (рис. 6).

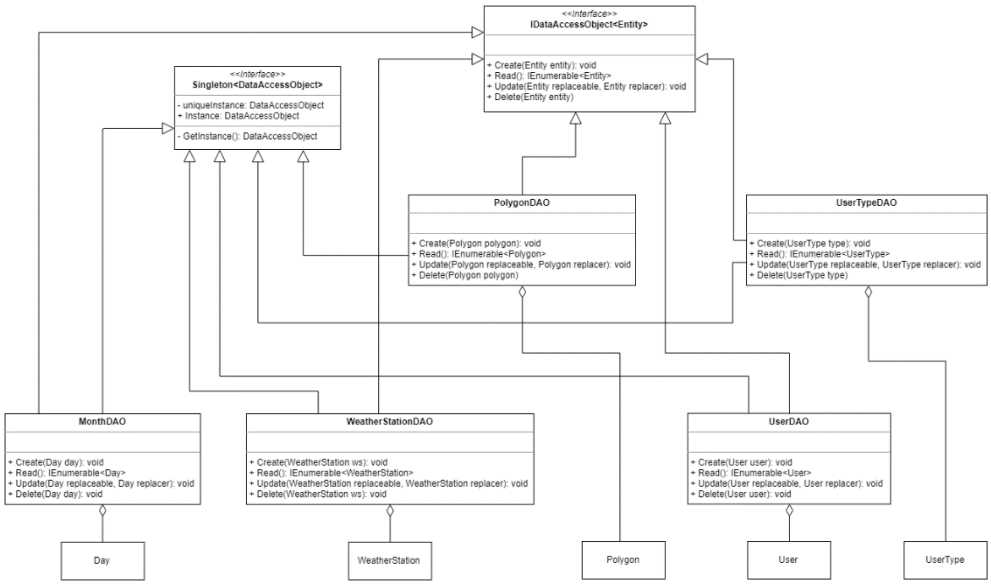

Рисунок 6 – Диаграмма классов, описывающая сущности для чтения и записи базы данных

На основании построенной диаграммы классов, описывающей основные сущности системы, была построена следующая схема базы данных (рис. 7).

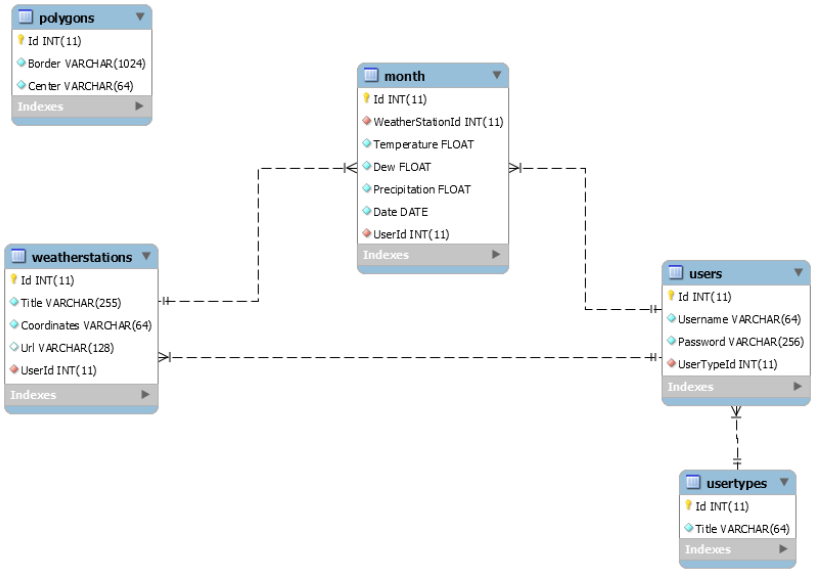

Рисунок 7 – Упрощенная схема базы данных

Представленная схема является упрощенным вариантом полной схемы базы данных, в которой вместо таблицы «Month» присутствуют 12 таблиц для каждого из месяцев. Так как для каждого месяца будут одни и те же поля, удалось создать такую схему для улучшения её восприятия. По данной схеме была создана база данных в СУБД MariaDB.

Для разработки информационной системы на языке программирования C# была использована среда разработки от Microsoft Visual Studio 2019. При создании нового проекта был выбран шаблон «ASP.NET Core Web Application», который позволит создать веб-приложение на платформе ASP.NET Core. В качестве используемой версии ASP.NET Core была выбрана самая последняя 2.2. Именно эта версия содержит важные изменения по сравнению с предыдущими и рекомендуется компанией Microsoft к использованию в личных проектах. Среди доступных шаблонов внутри «ASP.NET Core Web Application» был выбран «Web Application (Model-View-Controller)». Данный шаблон сразу настроит проект для использования в парадигме концепции «Модель-Представление-Контроллер» (Model-View-Controller или MVC). Главное назначение паттерна MVC заключается в легкости управления процессом создания комплексных веб-страниц. Весь проект разделяется на три зависимые части: модель (хранит данные), представление (отображает данные пользователю в виде HTML) и контроллер (производит обработку данных). Модель не всегда обязана присутствовать и в этом проекте было решено отказаться от нее. Все данные будут храниться и обрабатываться в парадигме шаблона проектирования Data Access Object в соответствующих классах, а контроллеры и представления будут использованы только для генерации HTML страниц.

В процессе программирования на языке C# были созданы следующие классы:

– классы основных сущностей для хранения данных из базы;

– классы, реализующие шаблон проектирования Data Access Object для операций чтения/записи из базы данных;

– классы и структуры реализующие алгоритмы для осуществления вычислений комплексного показателя пожарной опасности.

Проблемы безопасности личных данных пользователей, хранящихся в базе данных, являются очень важными проблемами. Многие из этих проблем удалось решить при разработке данного проекта, благодаря использованию сторонних программных библиотек.

Так, например, угроза атаки с помощью SQL-инъекции автоматически отпала с внедрением в проект микро-ORM библиотеки Dapper. Эта библиотека позволяет не только с легкостью создавать запросы к базе данных, но и гарантирует экранирование символов, благодаря использованию параметризованных SQL запросов. Для скрытия истинного значения паролей в проекте была использована библиотека Scrypt.NET, которая реализует один из лучших криптографических алгоритмов для шифрования паролей на сегодняшний день.

Таким образом был разработан веб-портал оценки и прогноза пожарной опасности по условиям погоды на языке программирования C# в среде разработки Visual Studio 2019. Далее приводится описание работы веб-сайта с иллюстрацией снимков экрана.

При переходе на сайт Firewatch открывается главная страница, визуально разделенная на две части: панель пользователя (слева) и область карты (справа) (рис. 8).

Рисунок 8 – Внешний вид главной страницы сайта

На панели пользователя располагаются ссылки «Вход» и «Регистрация», которые позволяют перейти на соответствующие страницы для входа в систему или регистрации в ней. Чуть ниже располагается поле для выбора даты по которой будет формироваться карта пожарной опасности. Если нажать на это поле, то появится маленькое окошко с календарем для выбора даты (рис. 9).

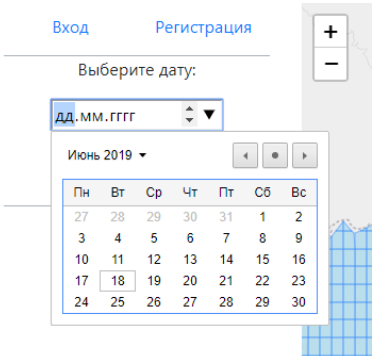

Рисунок 9 – Выпадающий календарь для выбора даты

После того, как пользователь осуществил выбор желаемой даты, необходимо нажать кнопку «Обновить карту». Система сделает все необходимые вычисления и выведет новую сгенерированную карту в область справа (рис. 10).

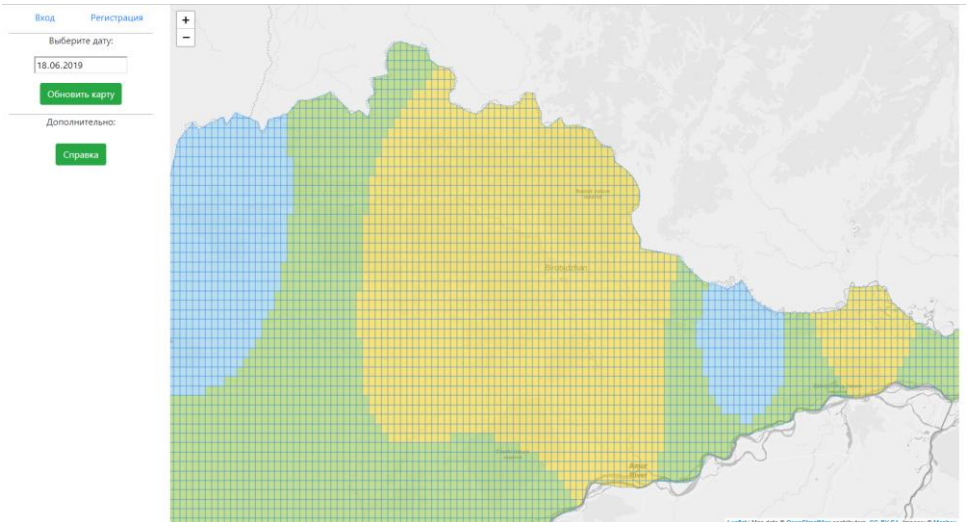

Рисунок 10 – Пример внешнего вида карты, сформированного на 18 июня 2019 года

Пользователи, впервые работающие с данной системой, могут не знать, как интерпретировать получившийся результат, для этого было создано отдельное окошко со справочной информацией, которое появляется после нажатия на кнопку «Справка» (рис. 11).

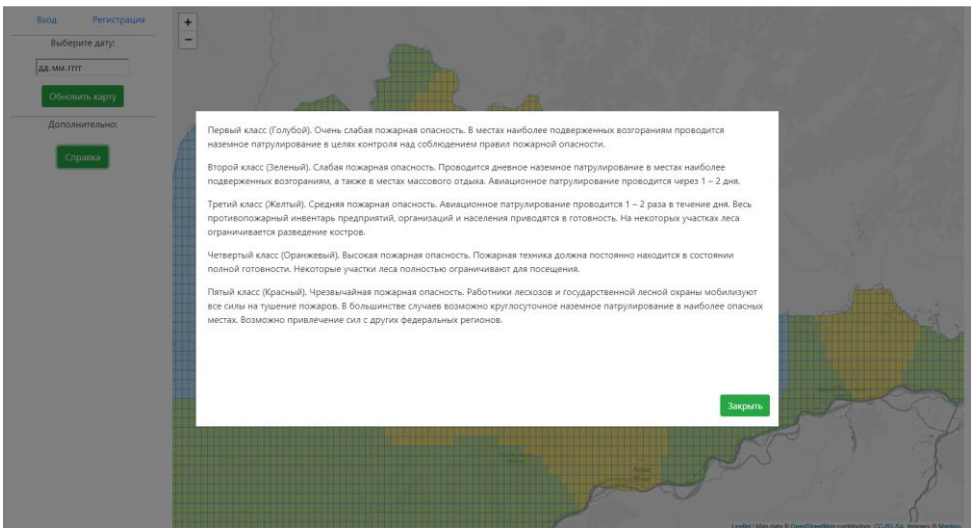

Рисунок 11 – Справочная информация, показывающая соответствие между классами пожарной опасности и работами служб лесной охраны

Для того, чтобы воспользоваться возможностью формирования собственных метеорологических данных, необходимо войти под своей учетной записью. Сделать это можно, перейдя по ссылке «Вход» в верхней части экрана. В итоге откроется новая страница (рис. 12).

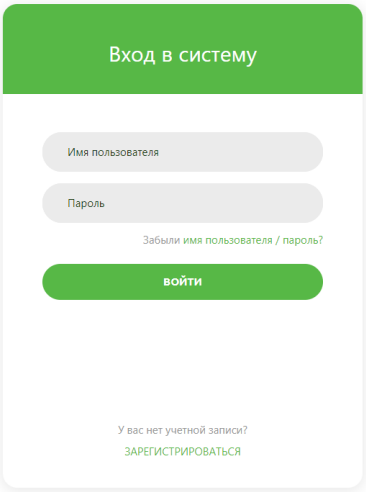

Рисунок 12 – Внешний вид страницы входа в систему

После успешного входа в систему пользователя перенаправят на главную страницу, где ему станут доступны новые элементы интерфейса для создания собственных сборок метеорологических данных (рис. 13).

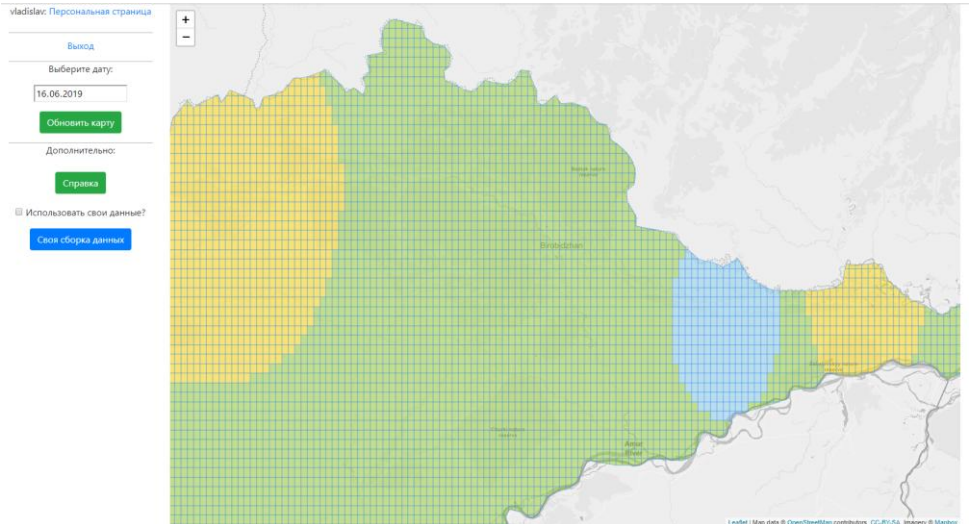

Рисунок 13 – Внешний вид главной страницы сайта после входа в учетную запись пользователя

Для того, чтобы приступить к заполнению собственных данных, необходимо нажать на синюю кнопку с надписью «Своя сборка данных». Загрузится новая страница с ссылками на таблицы из базы данных (рис. 14).

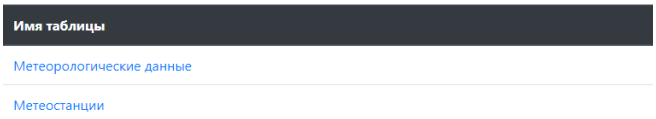

Рисунок 14 – Внешний вид страницы с таблицами из базы данных

Страница для таблицы «Метеорологические данные» разбита по месяцам на множество таблиц и выглядит следующим образом (рис. 15).

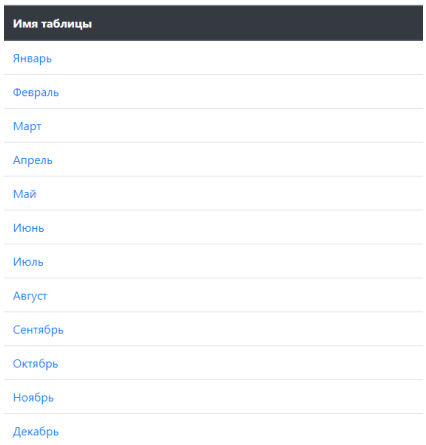

Рисунок 15 – Внешний вид страницы для таблицы «Метеорологические данные»

Выбрав месяц и нажав на соответствующую ссылку, пользователь попадает на страницу заполнения метеорологических данных по данному месяцу (рис. 16).

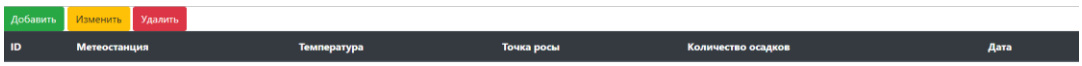

Рисунок 16 – Внешний вид страницы заполнения метеорологических данных по месяцам

Данная страница содержит табличное представление метеорологических данных, разбитое по полям, а также кнопки для добавления, изменения и удаления записей. При нажатии кнопки «Добавить» на этой же странице отобразится окно с полями для заполнения (рис. 17).

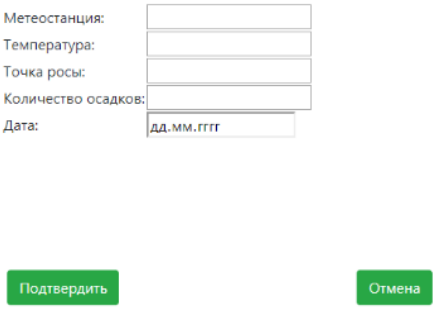

Рисунок 17 – Внешний вид окна для добавления новой записи в таблицу «Метеорологические данные»

Для того, чтобы изменить уже существующую запись в таблице, необходимо нажать на кнопку «Изменить», после чего отобразится окно для заполнения полей схожее с тем, что было при создании новой записи, но с одним новым полем «ID». Данное поле необходимо для указания системе какую существующую запись нужно изменить (рис. 18).

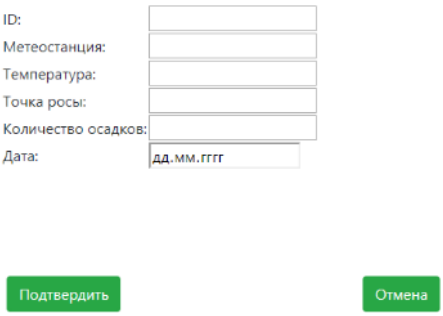

Рисунок 18 – Внешний вид окна для изменения существующей записи в таблице «Метеорологические данные»

Узнать необходимый ID записи можно из табличного представления на этой же странице. Заполнив все поля и нажав кнопку «Подтвердить», система обновит указанную запись в таблице.

Для удаления существующей записи служит кнопка «Удалить», по нажатии которой также открывается окно для указания единственного поля – «ID» (рис. 19).

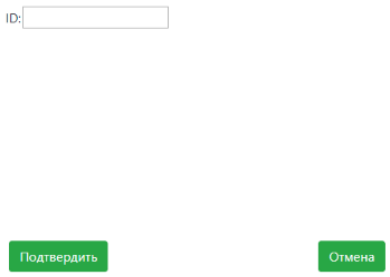

Рисунок 19 – Внешний вид окна для удаления существующей записи из таблицы «Метеорологические данные»

Указав значение поля «ID» и нажав кнопку «Подтвердить», данная запись исчезнет из таблицы навсегда.

Таблица «Метеостанции» выглядит и работает точно таким же образом, как и таблица «Метеорологические данные».

После заполнения таблиц для того, чтобы сгенерировать карту с новыми данными необходимо вернуться на главную страницу сайта и поставить галку напротив поля «Использовать свои данные?» (рис. 20).

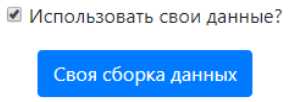

Рисунок 20 – Поле для установки собственных данных при генерации карты

Далее процесс протекает стандартно: пользователь выбирает дату из календаря и нажимает кнопку «Обновить карту». Теперь карта формируется с использованием тех данных, которые были заполнены пользователем.

Таким образом в результате данного исследования был проведен сбор и анализ теоретической информации, необходимой для моделирования карты пожарной опасности региона. Метод оценки пожарной опасности был впервые предложен Нестеровым В.Г. Он разработал «шкалу Нестерова» и в результате своих эмпирических исследований смог добиться значительных результатов в разработке модели для вычисления комплексного показателя пожарной опасности. Также были собраны дополнительные теоретические сведения об интерполяции значений, которые помогли вычислить значения комплексного показателя на всей территории Еврейской автономной области.

Был осуществлен выбор инструментов и технологий разработки наиболее подходящих для данного проекта. Также были отобраны различные программные библиотеки, которые заметно облегчили и ускорили процесс разработки.

В ходе проектирования были смоделированы основные компоненты системы и взаимосвязи между ними. На основе анализа предметной области определены основные сущности информационной системы. Созданы модели по разграничению прав доступа для пользователей и навигации по веб-порталу.

На основе диаграммы классов была спроектирована и создана база данных MariaDB.

В процессе разработки был настроен проект ASP.NET Core версии 2.2, запрограммированы основные сущности и классы для доступа к данным, реализованы алгоритмы из теоретической части, созданы веб-страницы с использованием HTML, CSS, JavaScript, в том числе Leaflet.js.

Результатом проделанной работы стало руководство пользователя, которое в подробностях описывает каждый шаг при работе с веб-порталом оценки и прогноза пожарной опасности по условиям погоды.

В конечном итоге разработанный веб-портал будет выложен в открытый доступ в сети Интернет. Данный проект разрабатывался с ориентиром на противопожарные службы и службы лесной охраны, которые смогут использовать эту систему в качестве дополнительного инструмента по предупреждению и тушению лесных пожаров. Также данный веб-портал доступен для всех заинтересованных граждан в качестве ознакомления с пожарной обстановкой на лесных территориях Еврейской автономной области.

Библиографический список

1. Добрых В.А., Шевцов Б.П., Юхно В.В., Ступак В.С., Гнатюк Ю.П., Романова Н.В., Брянцева А.И. Патология внутренних органов в условиях длительной задымленности воздуха вследствие лесных пожаров // Дальневосточный медицинский журнал. 2002. №3. С. 16-19.
2. Рябкова В. А., Брылева И. Н. Состояние здоровья населения Хабаровского края в условиях воздействия лесных пожаров // Дальневосточный медицинский журнал. 2002. №3. С. 41-44.
3. Фильков А.И., Гладкий Д.А. Разработка программного комплекса для визуализации результатов прогноза возникновения и распространения лесных пожаров в геоинформационной системе // Вычислительные технологии. 2011. №. 5. С. 89-99.
4. Гордиенко Е.П., Гордиенко Н.С., Паненко В.В. Современные технологии разработки геоинформационных систем // Техносферная безопасность. 2013. С. 99-104.
5. Слобожанина Е.А. Разработка структуры геоинформационных систем для задач экологического проектирования городской среды // Актуальные проблемы экологии и природопользования. 2017. С. 138-143.
6. Филатов Н.Н., Богданова М.С., Дерусова О.В., Литвиненко А.В., Толстиков А.В. Разработка геоинформационных систем водных объектов севера Европейской части России // Интеркарто. Интергис. 2018. №. 1. С. 19-29.
7. Deschamps A. et al. Geospatial data integration for applications in flood prediction and management in the Red River Basin // Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International. IEEE, 2002. Т. 6. С. 3338-3340.

8. Baranovskiy N., Zharikova M. A Web-Oriented Geoinformation System Application for Forest Fire Danger Prediction in Typical Forests of the Ukraine // Thematic Cartography for the Society. Springer, Cham, 2014. С. 13-22.
9. Polichtchouk Y. Geoinformation systems and regional environmental prediction // Safety science. 1998. Т. 30. №. 1-2. С. 63-70.
10. Gibin M. et al. An exploratory cartographic visualisation of London through the Google Maps API // Applied Spatial Analysis and Policy. 2008. Т. 1. №. 2. С. 85-97.
11. Lee K. Technical architecture for land monitoring portal using google maps API and open source GIS // Geoinformatics, 2009 17th International Conference on. IEEE, 2009. С. 1-5.
12. Milosavljević A., Stoimenov L., Djordjević-Kajan S. An architecture for open and scalable WebGIS // 8th AGILE Conference on GIScience, Estoril, Portugal. 2005. С. 629-634.
13. Brodaric B., Gahegan M. Distinguishing instances and evidence of geographical concepts for geospatial database design // International Conference on Geographic Information Science. Springer, Berlin, Heidelberg, 2002. С. 22-37.
14. Приказ Рослесхоза (Федеральной службы лесного хозяйства России) "Указания по противопожарной профилактике в лесах и регламентации работы лесопожарных служб" от 29.10.1993 № 289 // Рослесхоз (Федеральная служба лесного хозяйства России).
15. Метод оценки пожарной опасности в лесах по условиям погоды // Методический кабинет Гидрометцентра России URL: http://method.meteorf.ru/danger/fire/fire.html (дата обращения: 27.06.2019).
16. Inverse Distance Weighting (IDW) Interpolation // GISGeography URL: https://gisgeography.com/inverse-distance-weighting-idw-interpolation/ (дата обращения: 27.06.2019).
17. Dapper vs Entity Framework vs ADO.NET Performance Benchmarking // Exception Not Found URL: https://exceptionnotfound.net/dapper-vs-entity-framework-vs-ado-net-performance-benchmarking/ (дата обращения: 27.06.2019).
18. Entity Framework Core 2 Vs Dapper Performance Benchmark // Medium URL: https://medium.com/@engr.mmohsin/entity-framework-core-2-dapper-performance-benchmark-c29e8cce9e1b (дата обращения: 27.06.2019).
19. High Performance Object-Oriented Data Access with Dapper // Visual Studio Magazine URL: https://visualstudiomagazine.com/articles/2018/03/19/dapper-orm.aspx (дата обращения: 27.06.2019).
20. Leaflet API reference // Leaflet URL: https://leafletjs.com/reference-1.5.0.html (дата обращения: 27.06.2019).
21. Дядичев В.В., Стоянченко С.С., Дядичев А.В. CRUD система редактирования геоинформационных данных на основе фреймворков Bootstrap и Spring MVC // Известия сельскохозяйственной науки Тавриды. 2016. №. 5. С. 74-79.