

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет *компьютерных наук*
Кафедра *информационных систем*

Разработка веб-приложения «A interesting place»
Курсовой проект
по дисциплине
Технологии программирования

09.03.21 *Информационные системы и технологии*
Информационные системы и сетевые технологии

Обучающийся _____ *А.Г.Черенкова, 3 курс, д/о*
Обучающийся _____ *В.А.Евтухов, 3 курс, д/о*
Обучающийся _____ *М.А.Шамылов, 3 курс, д/о*
Руководитель _____ *Тарасов В. С.*

Воронеж 2022

Оглавление

Введение	3
Используемые определения.....	4
1. Постановка задачи	6
2. Анализ предметной области	7
3. Обоснование выбора продуктовых воронок	7
4. IDEF0 диаграмма.....	9
5. Диаграмма прецедентов.....	9
5.1. Функционал Пользователя	10
5.2. Функционал Организации	10
5.3. Функционал Администратора	10
6. Диаграмма классов.....	11
7. Диаграмма объектов	12
8. Диаграмма последовательностей.....	12
9. Диаграмма состояний	13
10. Диаграмма развертывания	14
11. Обоснование архитектуры проекта	15
12. ER – диаграмма.....	17
13. Реализация.....	17
14. Заключение.....	22
15. Список использованных источников.....	22

Введение

В современном мире существует большое количество приложений-карт, с помощью которых пользователи могут путешествовать, ориентируясь в пространстве. В этот момент человеку часто хочется посетить какие-либо заведения, например, кафе, музей и т.д. Используя в РФ картографические сервисы Яндекс в качестве источника данных о конкретных местах, пользователи должны дополнительно нажимать на конкретную точку, читать комментарии и смотреть дополнительно информацию в сети Интернет. Недавно сервисы Яндекс ввели такое обозначение, как «Хорошее место».

Каждый год Яндекс Бизнес вручает награду «Хорошее место» бизнесам с самым высоким рейтингом по отзывам и оценкам пользователей.

Знак «Хорошее место» получают организации:

- из 10% лучших в своём городе или области — отдельно для каждой категории;
- с рейтингом 4,2 и выше;
- с подтверждёнными данными — на это указывает «галочка» рядом с названием организации.

С 2021 года в проекте «Хорошее место» участвуют организации, основная категория которых относится к одной из перечисленных: Автосалон, Баня, Ветеринар, Быстрое питание, Кафе, Выставочный зал, Кондитерская, Ночной клуб, Салон красоты, Спортивный комплекс, Шиномонтаж, Хостел, Музей и многое другое.

Но, к сожалению, чтобы узнать о данном статусе заведения, необходимо искать информацию дополнительно, так же нажимая на конкретные точки на карте.

Было бы удобно иметь не просто карту, а карту, на который сразу будут выделены места со статусом «Хорошее место» для упрощенного восприятия карты во время путешествия.

По данным причинам веб-приложение с использованием картографического сервиса — это простое решение, с возможностью долгосрочной поддержки, которое доступно для большинства пользователей сети Интернет.

Используемые определения

Таблица 1. Используемые определения

Веб-сервис	идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя
Пользователь	человек, пользующийся функционалом веб-сервиса
Администратор	человек, имеющий доступ к расширенному функционалу веб-сервиса, имеющий знания о формате приема статей
Организация	торговое, промышленное, транспортное, страховое и другое объединение предпринимателей, частных лиц - акционеров для производственной, торговой или иной деятельности, приносящей доход и прибыль (дивиденды)
Хеширование паролей	особое преобразование любого объема информации, в результате которого получается некое отображение, образ, называемый хэшем (hash) — уникальная короткая символьная строка, которая присуща только этому массиву входящей информации
Back-end	программно-аппаратная часть приложения, отвечающая за функционирование внутренней части

Front-end	клиентская часть приложения, отвечающая за получение информации от сервера для её дальнейшего отображения на устройстве пользователя
Клиентская часть	средство (компьютер), принимающее данные от сервера и предоставляющее возможность взаимодействия с системой
Серверная часть	средство (компьютер), принимающее запрос от клиента, производит вычисления и формирует веб-страницу для дальнейшей отправки клиенту
Python	высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ
Flask	фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2

1. Постановка задачи

Целью данного курсового проекта является разработка самостоятельного веб-приложения, которое позволить видеть на карте «Хорошие места», а также добавлять их Организациям.

Для достижения данных целей сервис должен отвечать следующим требованиям:

1. Сервис должен иметь простой дизайн, выполненный в неброских цветах.
2. Сервис должен выполнять ряд основных функциональных задач:
 - 2.1. Создание средств для предоставления информации о карте с «Хорошими местами».
 - 2.2. Реализация формы для добавления своих мест Организацией.
 - 2.3. Возможность совершить Регистрацию/Авторизацию в системе.
 - 2.4. Возможность редактирования информации о «Хороших местах».

Для выполнения данных требований необходимо выполнить следующие задачи:

1. Проектирование веб-сервиса средствами языка UML.
2. Разработка back-end части, включающая в себя:
 - 2.1. Реализация ролей:
 - Пользователь
 - Организация
 - Администратор
 - 2.2. Реализация функциональных возможностей ролей.
 - 2.3. Подключение внешнего модуля для хранения данных.
 - 2.4. Разработка базы данных.
 - 2.5. Разработка функциональности статических и динамических страниц.
3. Разработка front-end части, включающей в себя:
 - 3.1. Создание макета дизайна.
 - 3.2. Реализация макета дизайна.
4. Проведение тестирования проекта

2. Анализ предметной области

2.1. Анализ существующих решений

2.1.1. <https://www.culture.ru/atlas/>

Достоинства:

1. Большое количество фильтров для удобного поиска конкретного места

Недостатки:

1. Не удобный интерфейс
2. Навязчивая реклама

2.1.2. <https://opentripmap.com/ru/>

Достоинства:

1. Большое количество объектов на карте
2. Гибкая система фильтра для поиска мест

Недостатки:

1. Сложно понять по карте характер отмеченного места

2.1.3. <https://www.marshruty.ru/Places/PlacesMap.aspx>

Достоинства:

1. Подробное описание мест

Недостатки:

1. Не удобный и не понятный интерфейс
2. Малое количество объектов, отмеченных на карте

3. Обоснование выбора продуктовых воронок

С помощью сервиса Яндекс. Метрика было принято решение создать три продуктовых воронки, основанных на целевых действиях пользователя веб-приложения.

Выделены три сценария пользователя:

1. Ознакомление с общедоступной информацией
2. Поиск «Хорошего места»
3. Подача заявок на добавление своего «Хорошего места» организацией

Причины выбора сценариев:

1. Общие причины:
 - Улучшение функциональных возможностей (удаление неиспользованных функциональных элементов, если такие обнаружены и добавление новых и/или необходимых элементов, и/или обновление существующих)
 - Понимание поведения среднестатистического зарегистрированного/незарегистрированного пользователя веб-

приложения (как люди попадают на сайт, что это за люди (какой возраст, какие устройства) и как они работают со страницами внутри сайта)

2. Частные причины:

Таблица 2. Частные причины выбора данных воронок

Ознакомление с общедоступной информацией	1. Анализ количества Гостей 2. Анализ кол-ва Гостей, проявивших интерес к просмотру «Хороших мест» на карте, путем ознакомления с общедоступной информацией
Поиск «Хорошего места»	1. Анализ данной последовательности действий необходим для анализа поведения пользователя при использовании поиска по карте определенного места или выбор категории
Подача заявок на добавление своего «Хорошего места» организацией	1. Подсчет кол-ва отправлений формы подачи заявок на добавление места на карту, путем нажатия пользователем кнопки «Отправить» 2. Определение сложности подачи заявок через данную форму

4. IDEF0 диаграмма

Рассмотрим основной бизнес – процесс на примере контекстной диаграммы, представленной на [рисунке 1](#). Данная диаграмма представляет собой общее видение процесса работы веб-приложения.

Работу сервиса регулирует Пользовательское соглашение, которое определяется Законами РФ и правилами приема заявок на появление на карте «Хорошего места», которые определяются Администратором приложения.

Обеспечивают работу веб-приложения Администратор и Исходная БД. На вход в Систему поступает Пользователь, причем, Пользователем выступает незарегистрированный пользователь, который в дальнейшем может принять роль Организации или Администратора. Т.е. пройдя авторизацию или регистрацию (только для Организаций), пользователь может принять одну из указанных выше ролей. На выходе Система выдает Утвержденные «Хорошие места».



Рисунок 1 - Контекстная диаграмма

5. Диаграмма прецедентов

Диаграммы прецедентов показывают действия и отношения актеров, т.е. действующих лиц системы, между собой и их действия по отношению к системе.

На [рисунке 2](#) представлены действующие лица со связями наследования. Как можно видеть, самым большим набором функций обладает Зарегистрированный, а самым малым – Незарегистрированный пользователь. Это означает, что функции Незарегистрированного пользователя наследует Зарегистрированный.

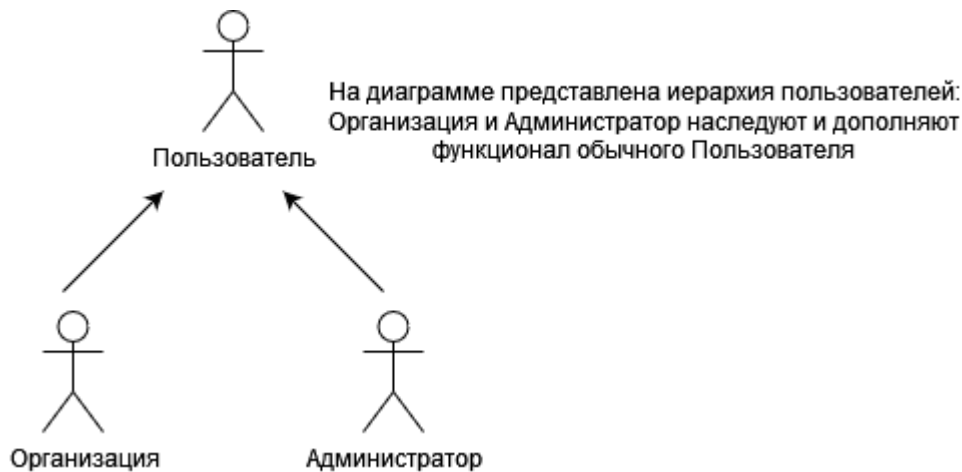


Рисунок 2 – Иерархия пользователей

На [рисунке 3](#) представлены действующие лица со связями наследования, а также их основные функции:

5.1. Функционал Пользователя

5.1.1. Доступ к просмотру Главной Страницы.

5.1.2. Доступ к просмотру меню Главной Страницы, переключение между пунктами меню Главная – «Хорошие места» – Регистрация – Авторизация.

5.2. Функционал Организации

Наследует функционал Пользователя.

5.2.1. Возможность редактировать свой профиль.

5.2.2. Возможность добавлять свое «Хорошее место».

5.3. Функционал Администратора

Наследует функционал Пользователя.

5.3.1. Возможность редактировать свой профиль.

5.3.2. Рассматривать заявки Организаций.

5.3.3. Добавлять и удалять Организации.

5.3.4. Удалять и добавлять «Хорошие места» на карту.

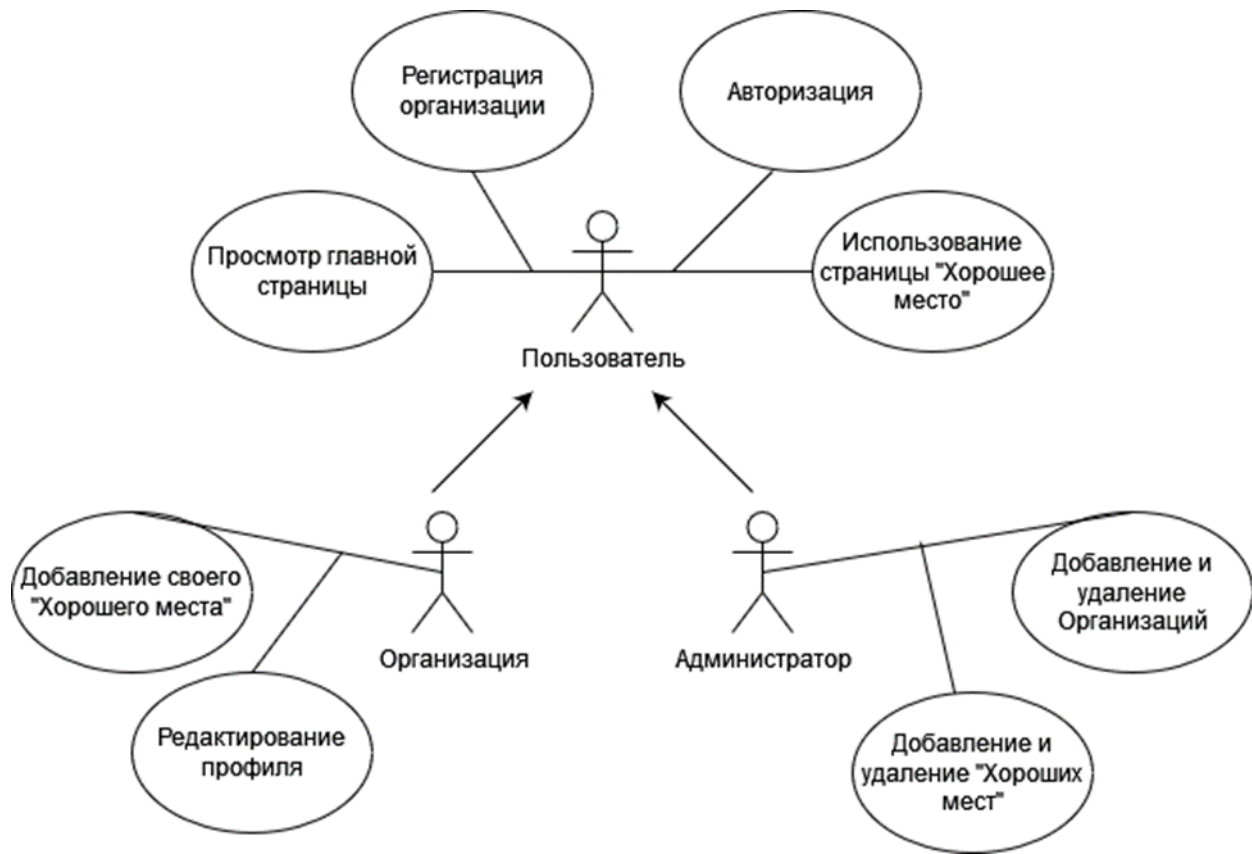


Рисунок 1 - Use-case диаграмма

6. Диаграмма классов

Диаграмма классов показывает классы системы, их атрибуты, методы и взаимодействия. Она представлена на **Ошибка! Источник ссылки не найден..**

7. Диаграмма объектов

Диаграмма объектов на [рисунке 5](#) показывают множество объектов - экземпляров классов и отношений между ними в некоторый момент времени. Она поясняет и дополняет диаграмму классов.

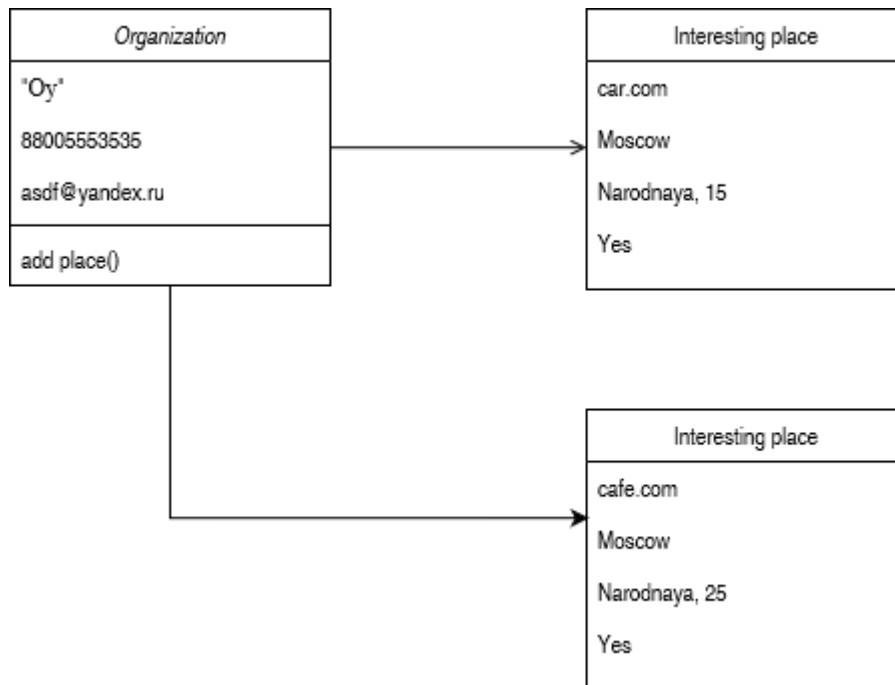


Рисунок 5 – Диаграмма объектов

8. Диаграмма последовательностей

Диаграмма последовательности отображает взаимодействие объектов в динамике. Т.е. диаграмма последовательностей отображает временные особенности передачи и приема сообщений объектами. На **Ошибка! Источник ссылки не найден.** представлен пример диаграммы для одной последовательности – подачи заявки на добавление на карту «Хорошего места».

Изначально Организация должна зарегистрироваться. Администратор производит проверку подлинности введенных данных (если не верны, то удаляет из бд данную организацию). После успешной проверки Организация может авторизоваться и подать заявку на добавление на карту своего «Хорошего места». После заполнения данных о «Хорошем месте» они переходят на подтверждение Администратору. После успешной проверки на подлинность заявки Организации приходит уведомление о том, что «Хорошее место» добавлено в бд приложения и скоро будет видно на общей карте.

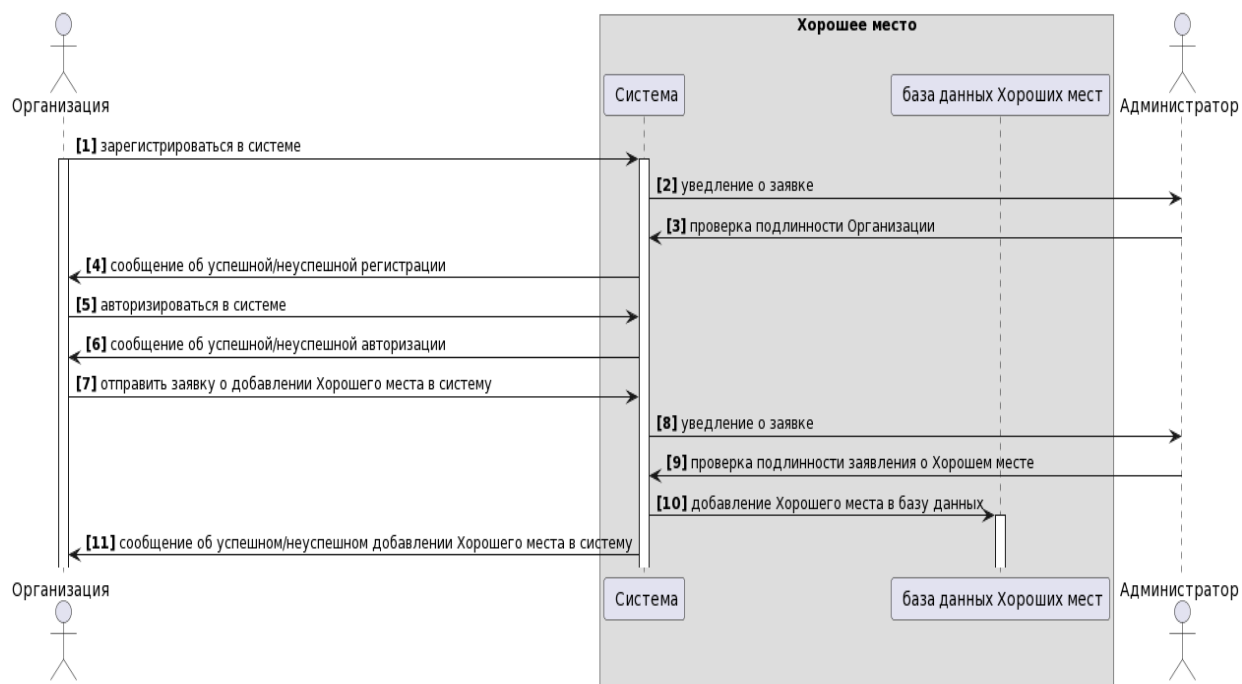


Рисунок 6 – Диаграмма последовательностей

9. Диаграмма состояний

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. На [рисунке 7](#) показано изменение состояний Формы заявки, в то время как на [рисунке 8](#) изменение состояний заявки.

Рассмотрим [рисунк 8](#):

после состояния Проверяется, заявка может принять одно из двух конечных состояний:

- Принята
- Не принята

Причем, каждому состоянию соответствует статус заявки. Состоянию Принята в соответствие ставится конечный статус «Принята», состоянию Не принята - «Не принята». Т.е. в итоге, заявка на «Хорошее место» либо будет на карте, либо нет, что соответствует **Ошибка! Источник ссылки не найден..**



Рисунок 7 – Диаграмма состояний

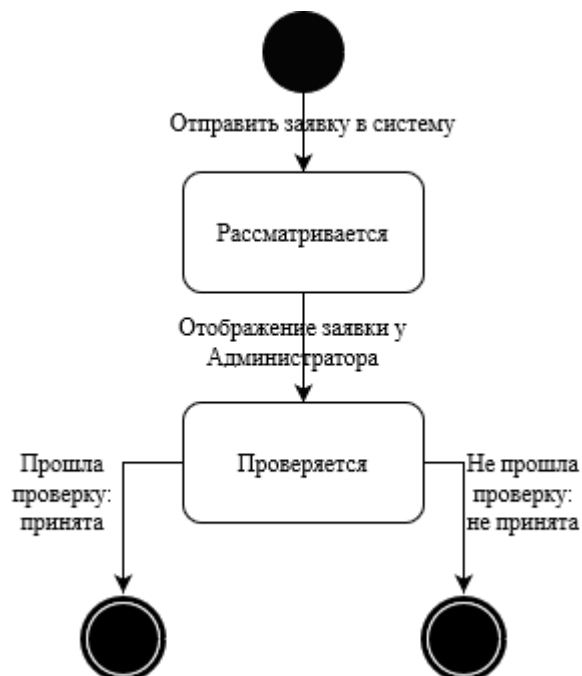


Рисунок 8 – Диаграмма состояний заявки

10. Диаграмма развертывания

Диаграмма развертывания на [рисунке 9](#) должна показывать, какие аппаратные компоненты существуют, какие программные компоненты работают на каждом узле, и как различные части этого комплекса соединяются друг с другом.

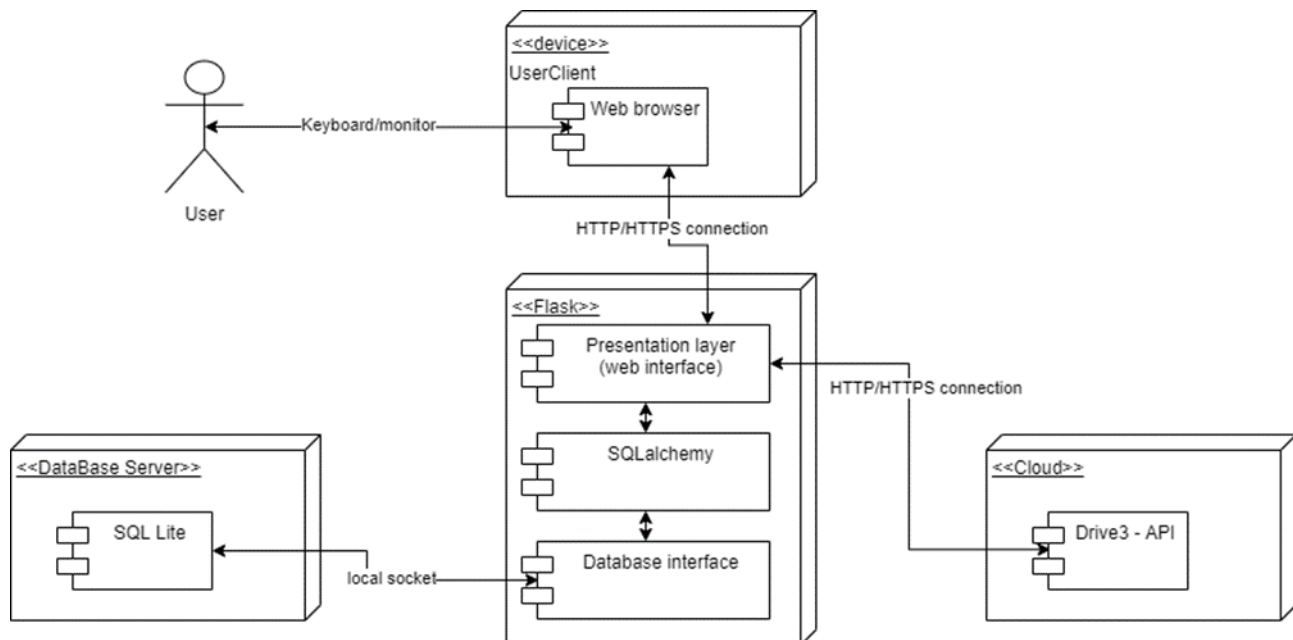


Рисунок 9 – диаграмма развертывания

11. Обоснование архитектуры проекта

Архитектура проекта: при проектировании приложения был использован паттерн проектирования MVT (одна из модификаций MVC). ORM в проекте использует паттерн Unit of Work.

Паттерн MVT (Model-View-Template) является модификацией MVC. Основное различие между этими двумя шаблонами состоит в том, что Flask сам заботится о части контроллера (программный код, который контролирует взаимодействие между моделью и представлением). Шаблон представляет собой файл HTML, смешанный с языком шаблонов Jinja2. Схематично паттерн можно определить следующим образом:

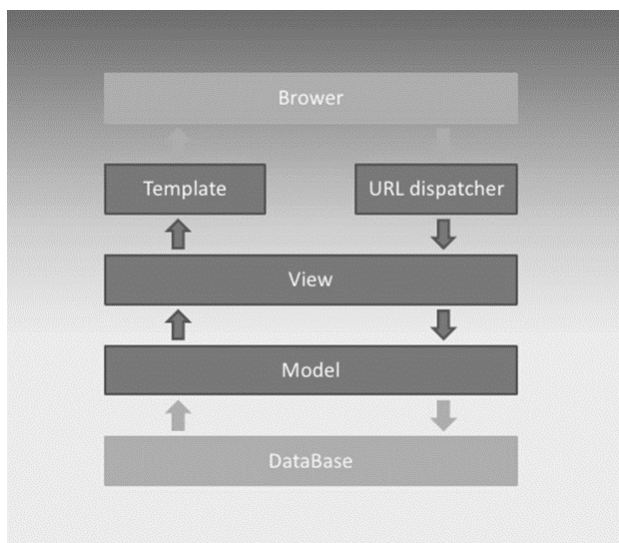


Рисунок 10 - MVT

Основные элементы паттерна:

URL dispatcher: при получении запроса на основании запрошенного адреса URL определяет, какой ресурс должен обрабатывать данный запрос.

View - это интерфейс: получает запрос, обрабатывает его и отправляет в ответ пользователю некоторый ответ. Если для обработки запроса необходимо обращение к модели и базе данных, то View взаимодействует с ними. Для создания ответа может применять Template или шаблоны. В архитектуре MVC этому компоненту соответствуют контроллеры (но не представления).

Model: описывает данные, используемые в приложении. Отдельные классы, как правило, соответствуют таблицам в базе данных.

Template: представляет логику представления в виде сгенерированной разметки html. В MVC этому компоненту соответствует View, то есть представления.

Причины выбора данного паттерна:

- четко разделена логика приложения
- облегчается поддержка и тестирование кода
- возможность автоматизации части контроллера во Flask

Аналогами выступали паттерны Model-View-Presenter и Model-View-View Model.

Причины, почему не были выбраны данные паттерны:

- Отсутствие предрасположенности выбранного фреймворка под данные паттерны
- Неоправданная сложность для данного проекта.

12. ER – диаграмма

При известных процессах, объектах и ролях необходимо определить, какие данные нужно хранить в базе. Схема базы данных определяет таблицы, поля и ограничения целостности. Данная схема представлена на [рисунке 11](#):

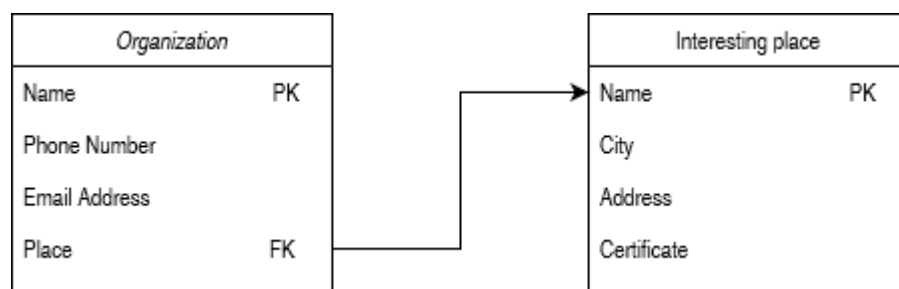


Рисунок 11 – ER диаграмма

13. Реализация

Переходя по ссылке, пользователь попадает на главную страницу приложения, где ему доступны метки с «Хорошими местами» ([рис. 12](#)).

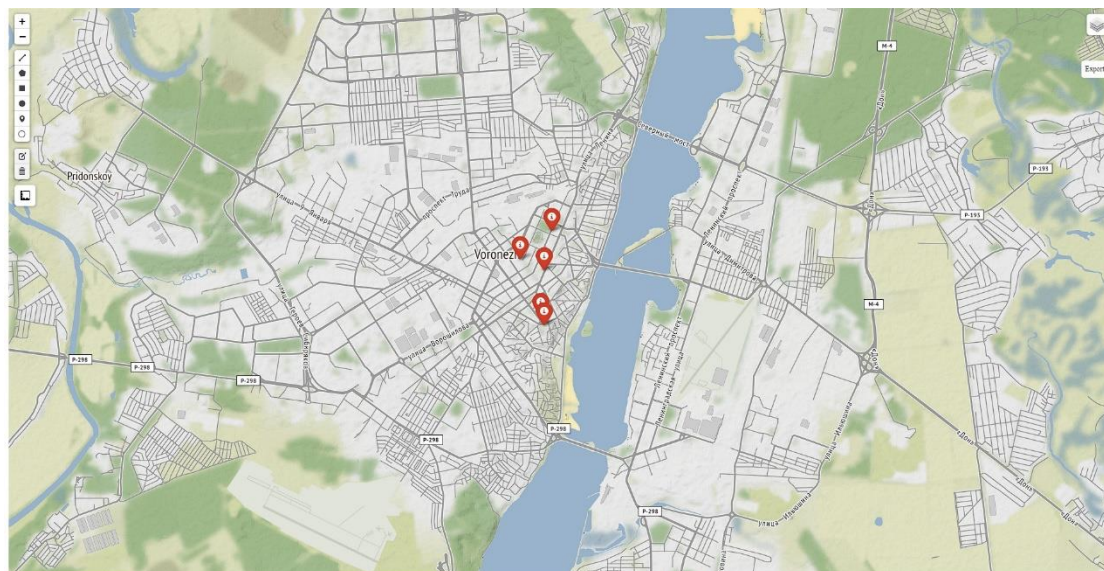


Рисунок 12 – Главная страница

При необходимости пользователь может редактировать вид карты ([рис. 13](#))

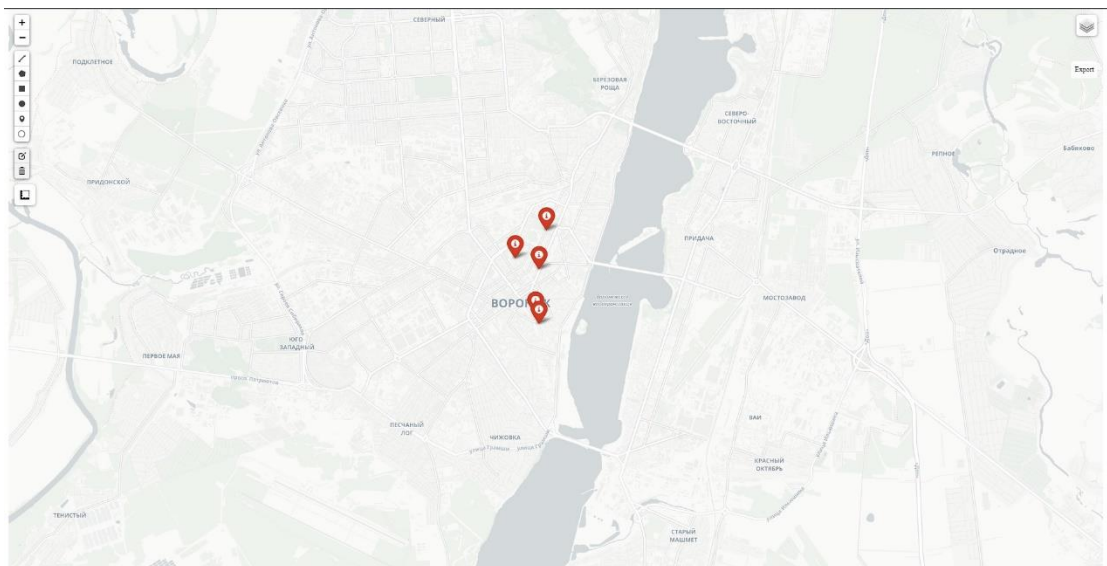


Рисунок 13 – Измененный вид карты

Организация или администратор имеет возможность входа в систему ([рис. 14](#))

Maps
Главная
Вход
Регистрация

Войдите

Введите имя

Введите пароль

Войти

Рисунок 14 – Вход в систему

Организация может подать заявку для того, чтобы ее место было показано на карте, как «Хорошее» ([рис. 15](#))

Maps Главная Выход

Введите данные организации

Введите имя организации

Введите координаты организации

Добавить заявку на рассмотрение

Рисунок 15 – Подача заявки на появление на карте «Хорошего места»

Организация имеет возможность зарегистрироваться ([рис. 16](#))

Maps Главная Вход Регистрация

Зарегистрируйтесь

Введите имя

Введите пароль

☐ Администратор

Зарегистрироваться

Рисунок 16 – Регистрация Организации

У пользователя есть возможность просмотра «Хороших мест» по категориям ([рис.17](#), [рис.18](#), [рис.19](#))

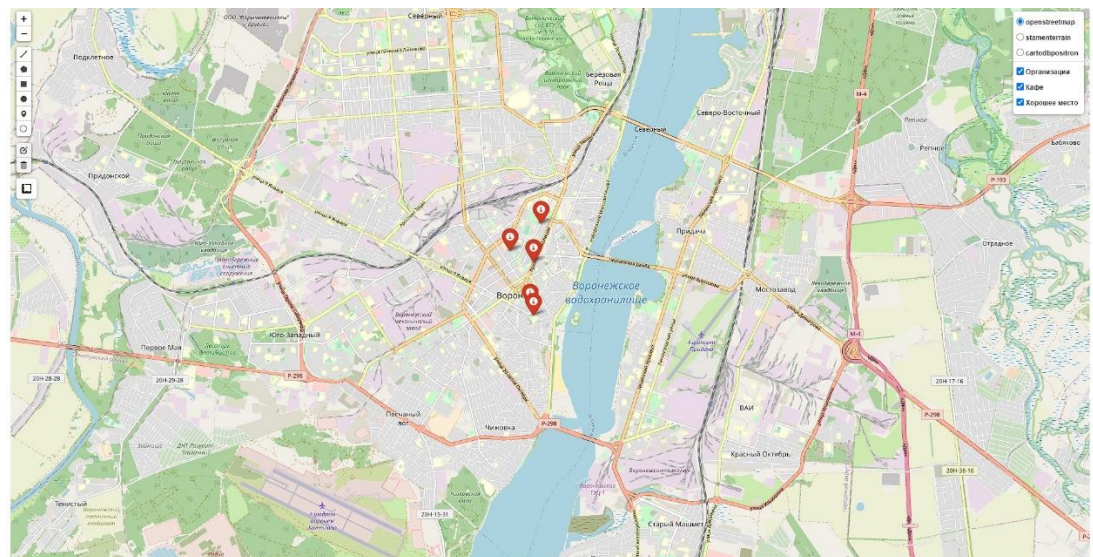


Рисунок 17 – Категории (1)

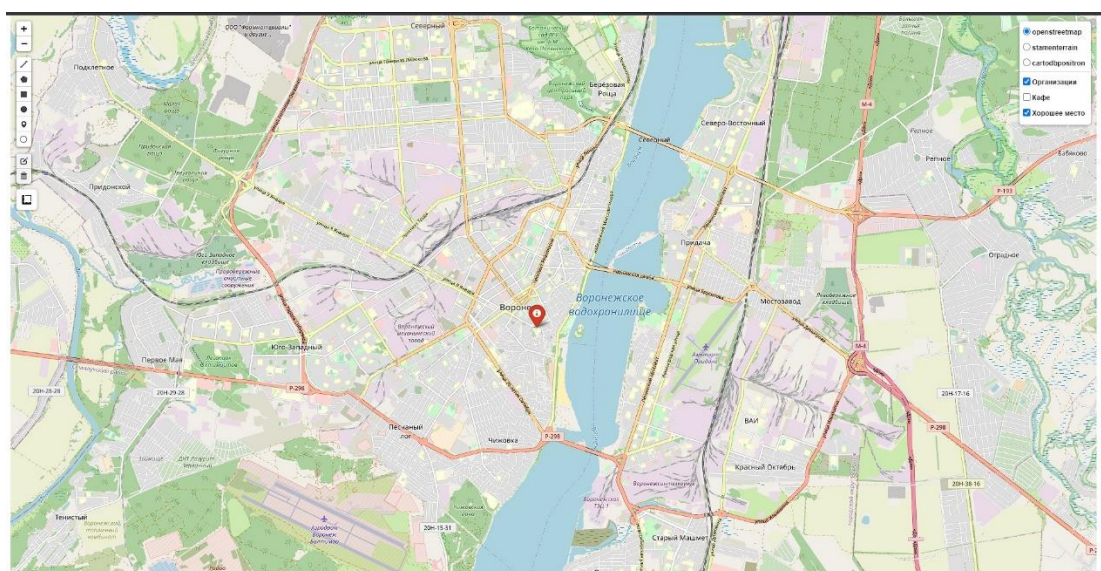


Рисунок 18 – Категории (2)

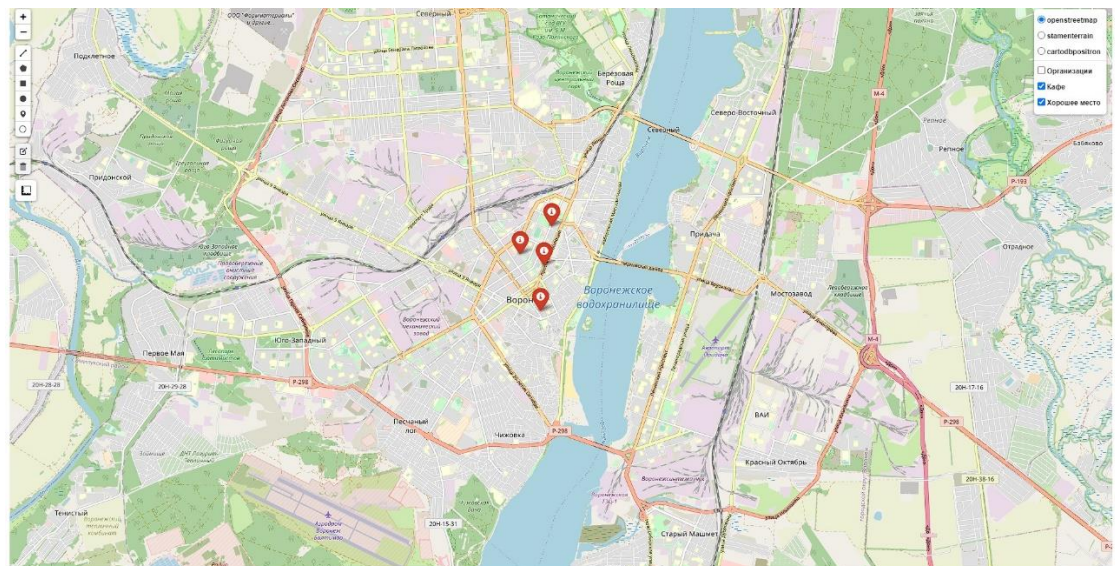


Рисунок 19 – Категории (3)

Администратор имеет возможность принять или проигнорировать заявку от Организации на появление «Хорошего места на карте» ([рис. 20](#))

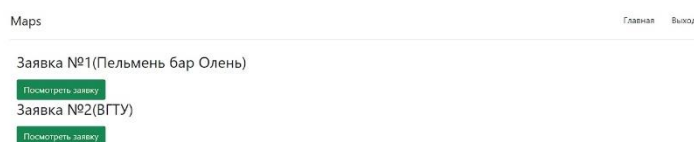


Рисунок 20 – Рассмотрение заявки

14. Заключение

В ходе данной курсовой работы была произведена разработка приложения «Хорошее место».

Серверная часть приложения была разработана на языке Python с использованием фреймворка flask и Jinja2. Клиентская часть написана на языках HTML с использованием фреймворка Bootstrap. Информация об организациях и местах хранится в базе данных SQLite.

Перед разработкой были произведены:

1. определение технических требований к web-сервису
2. проектирование web-сервиса средствами языка UML

После разработки было произведено тестирование и запись демо-видео. Приложение отвечает заявленным в Техническом Задании функциональным возможностям

15. Список использованных источников

1. Python Documentation / [сайт]. – URL: <https://www.python.org/doc/>
2. Леоненков. Самоучитель по UML / [сайт]. – URL: <http://khpi-iip.mipk.kharkiv.edu/library/case/leon/index.html>

3. SQLAlchemy / [сайт]. – URL: <https://www.sqlalchemy.org>
4. Bootsnipp / [сайт]. – URL: <https://bootsnipp.com>
5. Bootswatch / [сайт]. – URL: <https://bootswatch.com>
6. Namecheap / [сайт]. – URL: <https://www.namecheap.com/logo-maker/app/>
7. HTML CSS JavaScript MySQL PHP Bootstrap book academy примеры онлайн / [сайт]. – URL: <https://html5css.ru>
8. Хабр: Паттерны для новичков: MVC vs MVP vs MVVM / [сайт]. – URL: <https://habr.com/ru/post/215605/>
9. Хабр: Мега-Учебник Flask Глава 1: Привет, мир! (издание 2018) / [сайт]. – URL: <https://habr.com/ru/post/346306/>
10. Flask: User Guide https / [сайт]. – URL: <https://flask.palletsprojects.com/en/1.1.x/>
11. Quora: What is the difference between an MVC and an MVT framework? / [сайт]. – URL: <https://www.quora.com/What-is-the-difference-between-an-MVC-and-an-MVT-framework>
12. Real Python: Model-View-Controller (MVC) Explained – With Legos ? / [сайт]. – URL: <https://realpython.com/the-model-view-controller-mvc-paradigm-summarized-with-legos/>