

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра информатики и вычислительной техники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Информационная система для определения выгодности постройки
заведения общественного питания**

Направление: 09.03.02 «Информационные системы и технологии»

Профиль: «Информационные системы и технологии»

Квалификация: бакалавр

Студент гр. 607-91:

Хисматов Эльмир Зиннурович

Научный руководитель:

к.ф.-м.н., доцент кафедры ИВТ

Лысенкова Светлана Александровна

Допущен к защите:

«__» _____ 2023 г.

зав. кафедры ИВТ, к.т.н., доцент

Федоров Дмитрий Алексеевич

Сургут, 2023

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра информатики и вычислительной техники

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
студенту гр. 607-91 направления 09.03.02 «Информационные системы и технологии»

Хисматову Эльмиру Зиннуровичу

Тема: Информационная система для определения выгоды постройки заведения общественного питания.

Целевая установка: Проектирование и разработка информационной системы для определения выгоды постройки заведения общественного питания. Разработать модель базы данных, концептуальную модель предметной области. Разработать диаграмму использования, BPMN-диаграмму, диаграмму классов, диаграмму компонентов. Разработать пользовательский интерфейс информационной системы. Подготовить пояснительную записку.

Исходные данные: Литературные источники, Internet.

Начало проектирования: февраль 2023

Конец проектирования: июнь 2023

Содержание работы

1. Изучение предметной области и обзор аналогов.
2. Описание требований к обеспечению информационной системы.
3. Построение моделей и диаграмм (разработка концептуальной модели предметной области, диаграммы использования, BPMN-диаграммы, диаграммы классов, диаграммы компонентов).
4. Разработка пользовательского интерфейса.
5. Подготовка отчета.

Отчетный материал

1. Пояснительная записка на 50-70 листах.
2. Таблицы и рисунки.

Литература

1. Жукова, Ю. А. Информационные технологии в сфере общественного питания / Ю. А. Жукова // В мире научных открытий : Материалы V Международной студенческой научной конференции, Ульяновск, 20–21 мая 2021 года. Том III. Часть 1. – Ульяновск: Ульяновский государственный аграрный университет им. П.А. Столыпина, 2021. – С. 169-172. – EDN CEEUTB.
2. Габараева, З. И. Информационные технологии в общественном питании / З. И. Габараева, Д. Г. Мустафаева // Современные тенденции развития информационных технологий в научных исследованиях и прикладных областях : Сборник докладов III Международной научно-практической конференции, Владикавказ, 28–29 апреля 2022 года. – Владикавказ: Северо-Кавказский горно-металлургический институт (государственный технологический университет), 2022. – С. 21-23. – EDN VGVJKG.
3. Сергеева, И. И. Информационные системы и технологии в общественном питании / И. И. Сергеева, Е. А. Реденкова // Экосистема цифровой экономики: проблемы, реалии и перспективы : Сборник научных трудов национальной научно-практической конференции, Орел, 23–25 апреля 2018 года / Под редакцией Л.И. Малявкиной. – Орел: Орловский государственный университет экономики и торговли, 2018. – С. 172-178. – EDN YXKLAR.

Подписи:

Задание получил студент _____ /Хисматов Э.З./

Руководитель ВКР _____ /Лысенкова С.А./

Зав. кафедрой ИВТ, к.т.н _____ /Федоров Д.А./

РЕФЕРАТ

Работа содержит 53 страницы, 19 рисунков, 4 таблицы, 35 используемых источников, 6 приложений.

Ключевые слова: веб-приложение, веб-сервис, заведение общественного питания, сайт, модель предметной области, BPMN диаграмма, use-case диаграмма, спецификация прецедентов, диаграмма классов, диаграмма компонентов, информационная система.

Целью выпускной квалификационной работы является проектирование и разработка информационной системы для определения выгодности постройки заведения общественного питания.

В первой главе проведено исследование предметной области и выполнен сравнительный анализ существующих аналогов, выявлены основные недочеты аналогов, относительно разработанной информационной системы.

Во второй главе определены все необходимые виды обеспечения: функциональное, информационное, лингвистическое, программное, техническое, методическое, математическое.

В третьей главе произведено проектирование системы и реализованы схемы в нотации UML, BPMN и др., отражающие архитектуру и цель системы.

В четвертой главе описана разработанная информационная система, пользовательский интерфейс.

Достигнутые результаты:

- Построена концептуальная модель предметной области, use-case диаграмма, описание прецедентов, BPMN-диаграмма, диаграмма классов, диаграмма компонентов;
- Разработан прототип информационной системы для определения выгодности постройки заведения общественного питания

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

API – Application Programmin Interface (интерфейс прикладного программирования)

ИС – информационная система

СЕ – Catering Establishment (заведение общественного питания)

МПО – модель предметной области

ПО – программное обеспечение

UI – User Interface (пользовательский интерфейс)

BPMN – Business Process Model and Notation (Модель бизнес-процесса и нотация)

БД – база данных

CSS – Cascading style sheets (каскадные таблицы стилей)

HTML – Hypertext markup protocol (протокол передачи гипертекста)

UML – unified modeling language (универсальный язык моделирования)

SQL – Structured Query Language (язык запросов к базам данных)

ПК – персональный компьютер

JSON – JavaScript Object Notation (текстовый формат обмена данными)

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И ОБЗОР АНАЛОГОВ	10
1.1 Описание предметной области	10
1.2. Обзор аналогов	12
1.2.1. Google Maps	12
1.2.2. Яндекс Карты.....	14
1.2.3. 2GIS.....	16
1.3. Сравнение аналогов	17
ГЛАВА 2. ТРЕБОВАНИЯ К СИСТЕМЕ	19
2.1. Функциональные требования к информационной системе	19
2.2. Требования к информационному обеспечению информационной системы	19
2.3. Требования к лингвистическому обеспечению системы	19
2.4. Требования к алгоритмическому обеспечению системы	20
2.4.1. Выбор места и просмотр результата.....	20
2.4.2. Комментирование	22
2.5. Требования к программному обеспечению системы	23
2.5. Требования к техническому обеспечению системы	24
2.6. Требования к методическому обеспечению	25
2.7. Требования к математическому обеспечению	25
ГЛАВА 3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	27
3.1. Таблица SWOT-анализа.....	27
3.2. Концептуальная модель предметной области	28
3.3. Use case диаграмма	29
3.4. Описание прецедентов.....	30
3.5. BPMN диаграмма	33
3.6. Диаграмма классов	34
3.7. Диаграмма компонентов.....	39
ГЛАВА 4. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	41
4.1. Пользовательский интерфейс	41

4.2. Запрос через API и запрос к базе данных.....	44
ЗАКЛЮЧЕНИЕ	46
СПИСОК ЛИТЕРАТУРЫ.....	47
ПРИЛОЖЕНИЕ 1. Модуль event_reverse_geocode.js	52
ПРИЛОЖЕНИЕ 2. Модуль init.py	59
ПРИЛОЖЕНИЕ 3. Модуль main.py.....	60
ПРИЛОЖЕНИЕ 4. Модуль zapros_people.py	61
ПРИЛОЖЕНИЕ 5. Модуль coord_repository.py	62
ПРИЛОЖЕНИЕ 6. Модуль menu.js	65

ВВЕДЕНИЕ

Заведения общественного питания уже давно стали неотъемлемой частью жизни современного человека. Ведь всегда удобно, когда можно в любой момент зайти в кафе и перекусить, чем идти в магазин, купить продуктов и после процесса готовки наконец-то поесть.



Рис. 1 Численность предприятий общественного питания в России

По оценкам BusinessStat (рис. 1) численность предприятий общественного питания за период 2017-2019 увеличилось на 3%: с 186,9 до 193,1 тыс. В 2020 году из-за пандемийной обстановки в мире их число уменьшилось на 6% [14].

А с начала 2021 года количество заведений общественного питания увеличилась на 8,3% и продолжает расти, так как данная отрасль смогла преодолеть трудности, связанные с пандемией коронавируса и снова начать наращивать свои ресурсы [15].

И все эти изменения заставляют владельцев бизнеса в данной сфере искать новые средства обработки информации о своих конкурентах и той области, в которой собираются строить новое заведение. Особое внимание уделяется

поисково-информационным картографическим службам, ведь в последнее время они пользуются большой популярностью среди пользователей смартфонов. Ведь достаточно кликнуть на карте и очень быстро появится много информации о выбранном месте [16].

Актуальность работы заключается в том, что на данный момент отсутствует какая-либо информационная система, показывающая важные данные одновременно для владельцев заведений, ресторанных критиков и экспертов, такие как:

- Выбор типа заведения при помощи фильтра;
- Показ собираемых данных в формы;
- Расчёт полученных данных прямо в формах;
- Хранение оставленных отзывов в БД;
- Показ результата выгодности в процентах.

Целью выпускной квалификационной работы является разработка информационной системы, позволяющей собрать вышеуказанные данные и показать предполагаемую выгодность постройки заведения общественного питания. Под выгодностью понимается то, насколько успешно будет будущее заведение. Принесет ли оно прибыль или уже уйдет в долги и его придется закрыть. И эта выгодность будет показана в виде процента успешности, которая отражается в ходе работы системы. Все основные функции системы будут реализованы в виде веб-сервиса. Сайт не требует каких-либо лишних действий. Все что требуется от пользователя это подать в качестве входных данных лишь место предполагаемой стройки заведения, а на выходе он получит все данные о количестве конкурентов, наличии организаций рядом и так далее.

Чтобы разработать информационную систему, ставятся следующие задачи:

1. Изучить предметную область и построить ее концептуальную модель;
2. Определить аналоги и провести их анализ;

3. Описать все необходимые виды обеспечения;
4. Спроектировать информационную систему;
 - 4.1. Составить таблицу SWOT анализа;
 - 4.2. Построить диаграмму вариантов использования use case;
 - 4.3. Описать возможные прецеденты работы с программой;
 - 4.4. Реализовать поведенческую модель в нотации BPMN;
 - 4.5. Построить диаграмму классов;
 - 4.6. Построить диаграмму компонентов;
5. Разработать информационную систему;
 - 5.1. Определить все необходимые модули программы;
 - 5.2. Разработать код;
 - 5.3. Разработать сайт под систему;
 - 5.4. Реализовать на сайте работу всех модулей.

Апробация работы:

1. Участие IX Всероссийской конференции молодых ученых «Наука и инновации XXI века» (г. Сургут, 2023 г.)). Название проекта «Информационная система для определения выгодности постройки заведения общественного питания».
2. Участие в XXVII Открытой региональной студенческой научной конференции имени Г.И. Назина «Наука 60-й параллели». Название проекта «Информационная система для определения выгодности постройки заведения общественного питания».
3. Участие в VI Национальной научно-практической студенческой конференции «Проблемы и решения автоматизации XXI века». Название проекта «Информационная система для определения выгодности постройки заведения общественного питания»

ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И ОБЗОР АНАЛОГОВ

1.1 Описание предметной области

Заведение общественного питания — это место, где можно заказать и употребить еду и напитки, приготовленные профессиональными поварами и барменами. Обычно такие заведения предлагают различные виды блюд и напитков, которые могут быть поданы на месте, а также предоставляют услугу доставки еды на дом или на работу [13].

Заведения общественного питания могут иметь различные формы собственности и виды организации, такие как рестораны, кафе, бары, закусочные, фаст-фуды, кофейни и т.д. Эти заведения могут быть ориентированы на различные типы кухонь и стилей обслуживания, включая национальную кухню, диетическое питание, вегетарианскую или веганскую кухню, быстрое питание и т.д.

Заведения общественного питания являются важным элементом индустрии общепита и играют значительную роль в социальной жизни общества. Кроме того, эти заведения также являются важным источником дохода для многих людей, включая владельцев, менеджеров, поваров, официантов и других работников, связанных с этой отраслью [17].

И чтобы преуспевать в этой индустрии владельцам требуются различные методы анализа данных. Многие из них используют поисковые информационно-картографические службы для определения места постройки нового заведения [18].

Разрабатываемая система сможет помочь владельцам с этим вопросом. В ней будут различные вместе работающие модули в роли которых выступают сущности проекта.

В качестве основной сущности выступает «Заведение общественного питания». Основными понятиями, связанными с данной сущностью, являются:

- Основная информация о заведении общественного питания (место, тип заведения, основные данные);
- Место, в котором предполагается постройка;
- Тип заведения;
- Основные данные (среднее количество людей, количество конкурентов и организаций, цена аренды);
- Сайт с информационной системой.

1.2. Обзор аналогов

Разрабатываемая информационная система носит название «CE-helper». Она имеет как прямые, так и косвенные аналоги, которые реализуют некоторые функции данной системы. Рассмотрим по одному аналогу от каждого типа приложений и оценим их по 6 критериям (Таблица 1).

1.2.1. Google Maps

Google Maps [2] - это онлайн-карта, разработанная компанией Google, которая позволяет пользователям искать места и получать информацию о них, создавать маршруты и планировать поездки. С помощью Google Maps можно найти местоположение организаций и учреждений, таких как рестораны, гостиницы, магазины, банки и медицинские учреждения, а также искать общественный транспорт, автомобильные маршруты и пешеходные маршруты.

Кроме того, на карте можно увидеть информацию о трафике, оценки и отзывы пользователей об организациях, фотографии и видео, а также другую полезную информацию. На рис. 2 представлен интерфейс веб-сервиса Google Maps:

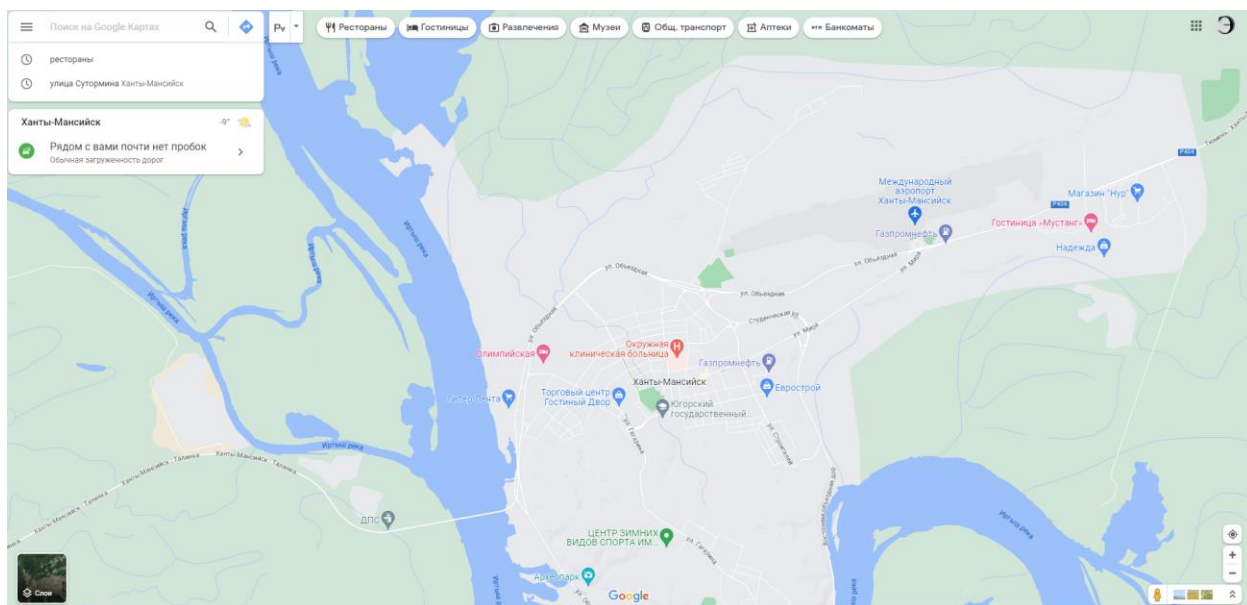


Рис. 2. Основной раздел интерфейса Google Maps

На рис. 3 представлен раздел с использованием фильтра «рестораны»

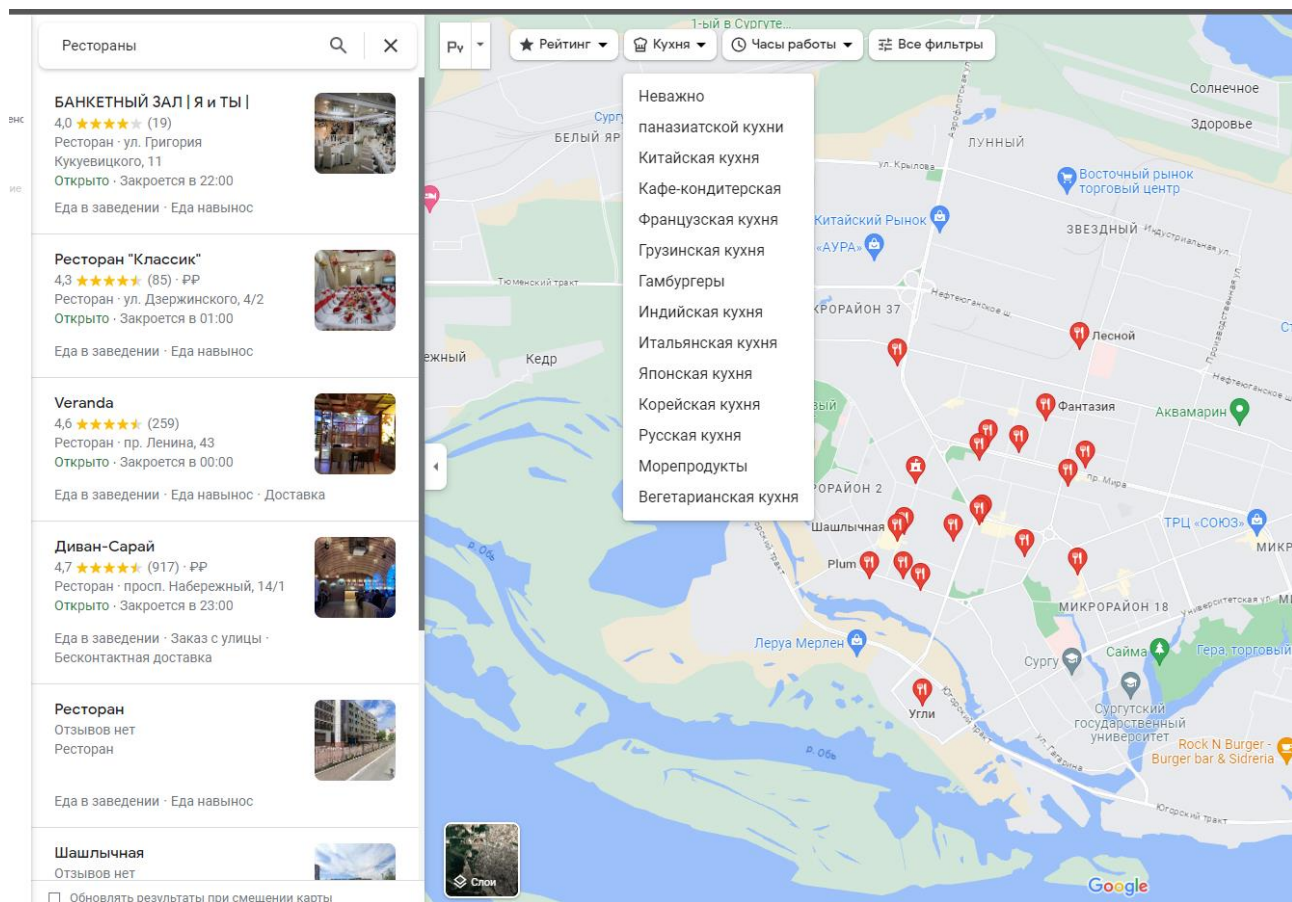


Рис. 3 Поиск по ресторанам

Рассмотрим преимущества и недостатки использования Google Maps в качестве инструмента для владельцев заведений общественного питания:

Преимущества:

1. Широкий охват аудитории: Google Maps является одним из самых популярных сервисов для поиска и планирования маршрутов, поэтому он может быть очень полезен для привлечения новых клиентов в ресторан.
2. Детальная информация о заведении: владельцы ресторанов могут добавить на Google Maps информацию о своем заведении, включая адрес, телефон, часы работы, меню и т.д. Это позволяет потенциальным клиентам легко найти и оценить заведение.
3. Обзоры и рейтинги: пользователи Google Maps могут оставлять отзывы и оценки на заведения, что может повысить или понизить репутацию

ресторана. Владельцы заведений могут использовать эту информацию для улучшения качества обслуживания и продуктов.

4. Интеграция с другими сервисами: Google Maps интегрируется с другими сервисами Google, такими как Google My Business и Google AdWords, что позволяет владельцам ресторанов управлять своими профилями и рекламными кампаниями.

Недостатки:

1. Нет контроля над информацией: информация о заведении на Google Maps может быть неправильной или устаревшей, и владелец ресторана не всегда может ее изменить.
2. Рейтинг может быть завышенным или заниженным: рейтинг заведения на Google Maps может быть завышенным или заниженным из-за неправильных или предвзятых отзывов.
3. Конкуренция: Google Maps может отображать информацию о конкурирующих заведениях, что может снижать число клиентов, посещающих ваш ресторан.
4. Зависимость от интернета: Google Maps требует подключения к интернету, что может быть проблемой в местах с плохой связью.

1.2.2. Яндекс Карты

Яндекс Карты [3] — это онлайн-картографический сервис, предоставляемый компанией Яндекс. Он позволяет пользователям просматривать интерактивные карты, планировать маршруты, искать информацию о предприятиях, местах и достопримечательностях, а также получать рекомендации по местам для посещения на основе личных предпочтений и рейтингов. На рис.3 показан основной интерфейс Яндекс Карт.

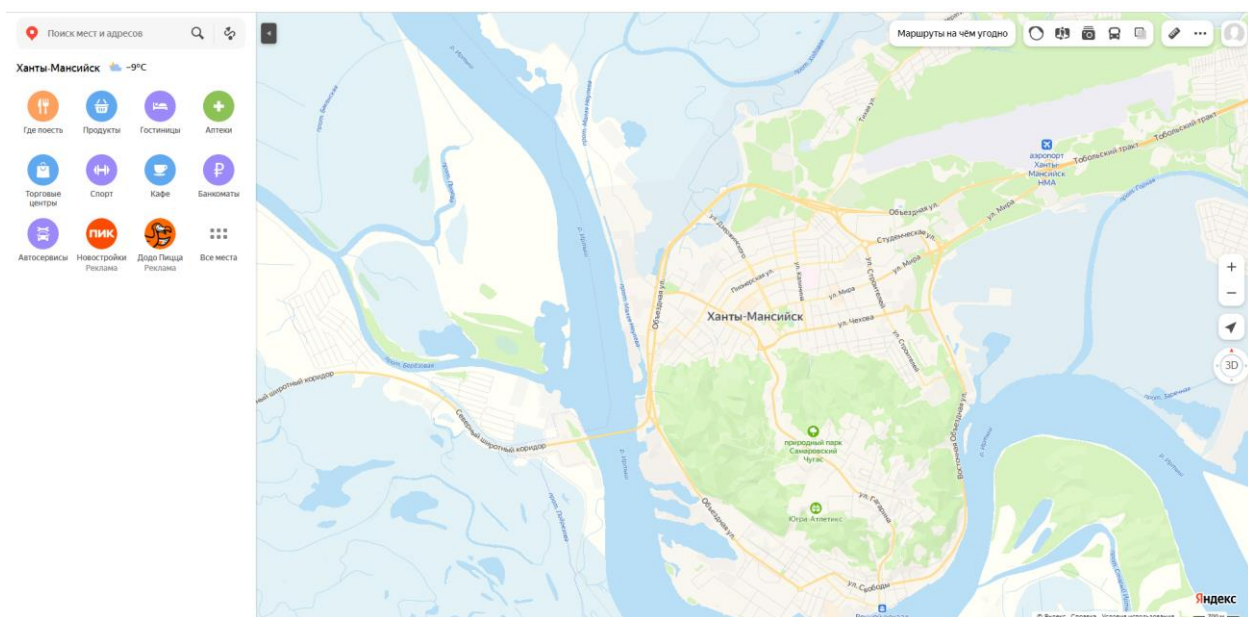


Рис. 4. Основной раздел интерфейса Яндекс Карты

На рис. 5 показан поиск с применением фильтра раздела «Кафе»

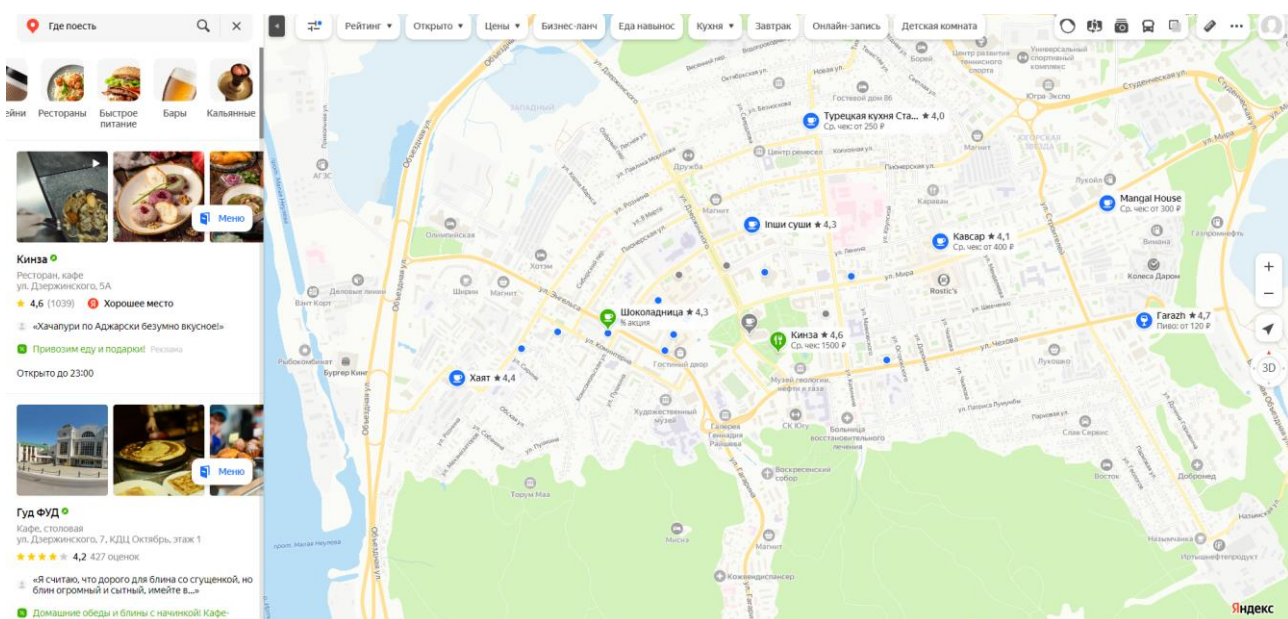


Рис. 5 Работа фильтра поиска

Преимущества:

1. Один из популярнейших сервисов в России, что означает частое обновление информации.
2. Сервис полностью бесплатен.
3. Удобный интерфейс.
4. Возможность показа различных слоев на карте (плотность людей, пробки и так далее).

Недостатки:

1. Также как и сервис Google Maps не работает без интернета.
2. Недостаточная популярность за пределами России, потому что сервис ориентирован на российский рынок.

1.2.3. 2GIS

2GIS [4] — это онлайн-картографический сервис, разработанный компанией DoubleGIS. Он предоставляет пользователям подробные интерактивные карты, а также информацию о местоположении предприятий и организаций, включая адреса, телефоны, часы работы и другие подробности.

Одним из главных преимуществ 2GIS является то, что сервис предоставляет большое количество информации о местоположении предприятий и организаций, включая не только контактную информацию, но и отзывы пользователей, фотографии, цены и другие детали. Это может быть полезно для поиска нужной услуги или места, а также для оценки качества обслуживания.



Рис. 6. 2 GIS

На рис.7 показан широкий спектр фильтров по категории «Где поесть»

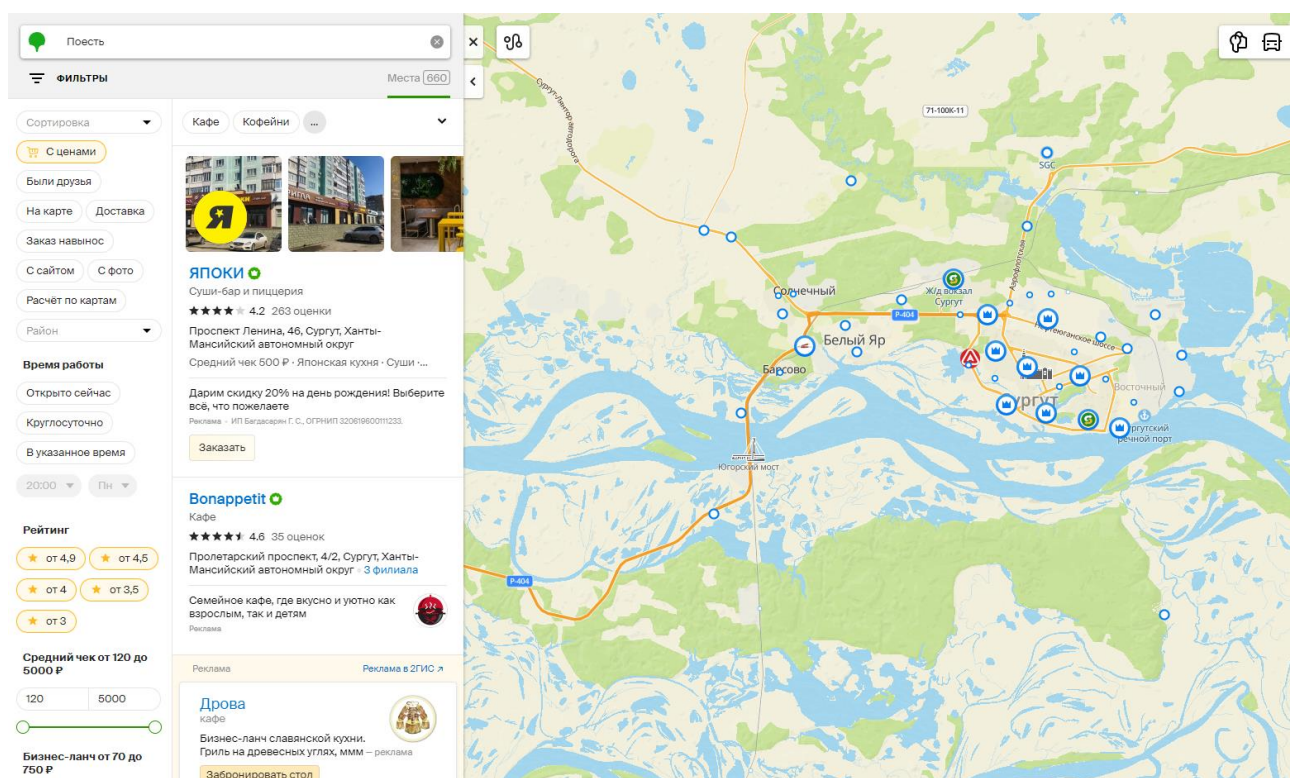


Рис. 7 Фильтр от 2GIS

Преимущества:

1. Одним из главных преимуществ является возможность работы оффлайн. Для этого требуется скачать карту с необходимым городом.
2. Широкий фильтр поиска и большое меню различных показаний у показываемого объекта.

Недостатки:

1. Карты необходимо постоянно обновлять, чтобы избежать получения недостоверной информации;
2. Менее популярен в России чем Яндекс Карты.

1.3. Сравнение аналогов

На таблице 1 показано сравнение аналогов по выбранным критериям с разрабатываемым веб-сервисом «CE-helper»:

Табл. 1. Сравнение аналогов

Критерий	CE-helper	Google Maps	Яндекс Карты	2 GIS
----------	-----------	-------------	--------------	-------

Использование оффлайн	Необходимо подключение к Интернету	Необходимо подключение к Интернету	Необходимо подключение к Интернету	Можно скачать карту и использовать ее оффлайн
Простота интерфейса	Прост и понятен	Прост и понятен	Прост и понятен	Прост и понятен
Фильтр по типу заведения	Реализован в виде выпадающего меню	Реализовано только для ресторанов	Реализовано в отдельном меню	Реализовано в отдельном меню
Количество людей	Показ в точных цифрах	Показ в виде цветовой схемы	Показ в виде цветовой схемы	Показ в виде цветовой схемы
Одновременный сбор нескольких типов данных	Одновременный показ результата поиска	Только если в одной категории	Только если в одной категории	Только если в одной категории
Формула успешности	Показывает процент успеха	Не реализовано	Не реализовано	Не реализовано

Как видно из таблицы, веб-сервис «CE-helper» имеет преимущества по всем выбранным показателям, а именно по одновременному сбору данных, ведь он не ограничивается только одной категорией, может показать количество людей в точных цифрах, формула успешности реализована только в разрабатываемом веб-сервисе. Поэтому «CE-helper» является наилучшим решением для владельцев заведений общественного питания в выборе информационной средства для анализа области предполагаемой постройки.

ГЛАВА 2. ТРЕБОВАНИЯ К СИСТЕМЕ

2.1. Функциональные требования к информационной системе

К информационной системе для сбора данных предъявляются следующие функциональные требования:

- Выбор типа заведения при помощи фильтра;
- Показ собираемых данных в формы;
- Расчёт полученных данных прямо в формах;
- Хранение оставленных отзывов в БД;
- Показ результата выгоды в процентах.

2.2. Требования к информационному обеспечению информационной системы

Источниками информации для информационной системы является система API от Яндекс Карты, а также база данных со значениями количества людей и цены аренды.

Входными данными являются выбор места и посылаемые запросы от системы, с учетом выбора типа заведения.

Выходными данными является показ выбранных для системы данных и процент успешности.

2.3. Требования к лингвистическому обеспечению системы

Лингвистическое обеспечение информационной системы для определения выгоды постройки заведения общественного питания включает в себя следующие языки программирования и технологии:

- На этапе проектирования программного обеспечения с целью создания проектной документации применяется универсальный язык моделирования UML, на основе которого планируется разработать следующие диаграммы:
 - Модель предметной области;

- Use case диаграмма;
- Спецификация прецедентов;
- Описание бизнес-процесса происходит на универсальном языке BPMN;
- Диаграмма классов;
- Диаграмма компонентов;
- «SQLite DB» версии 3.12.2. Это встраиваемая система управления базами данных, в которых хранятся данные о количестве людей, необходимые для правильной работы системы.

2.4. Требования к алгоритмическому обеспечению системы

В данном пункте представлено несколько основных алгоритмов взаимодействия пользователя с информационной системой.

2.4.1.Выбор места и просмотр результата

Процесс выбора места и просмотра результата пользователя должен быть реализован согласно следующему алгоритму (рис. 8):

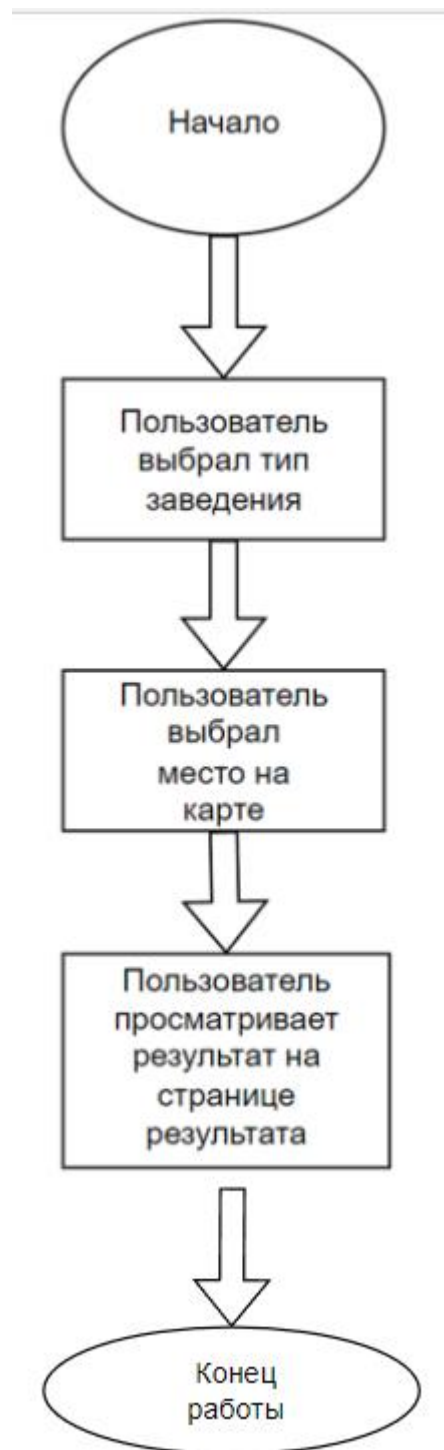


Рис. 8 Основной алгоритм выбора места заведения пользователем

После того как пользователь открыл сайт, он нажимает на кнопку «Начать работу». Авторизация для работы не требуется. Затем на рабочей странице сайта, он выбирает тип заведения, например «Кафе». Далее он кликает на карте выбирая места и жмет кнопку «Показать результат». Пользователь переходит на страницу результата и смотрит собранные данные.

2.4.2. Комментирование

Процесс комментирования и просмотра отзывов пользователем должен быть реализован согласно следующему алгоритму (рис. 9):

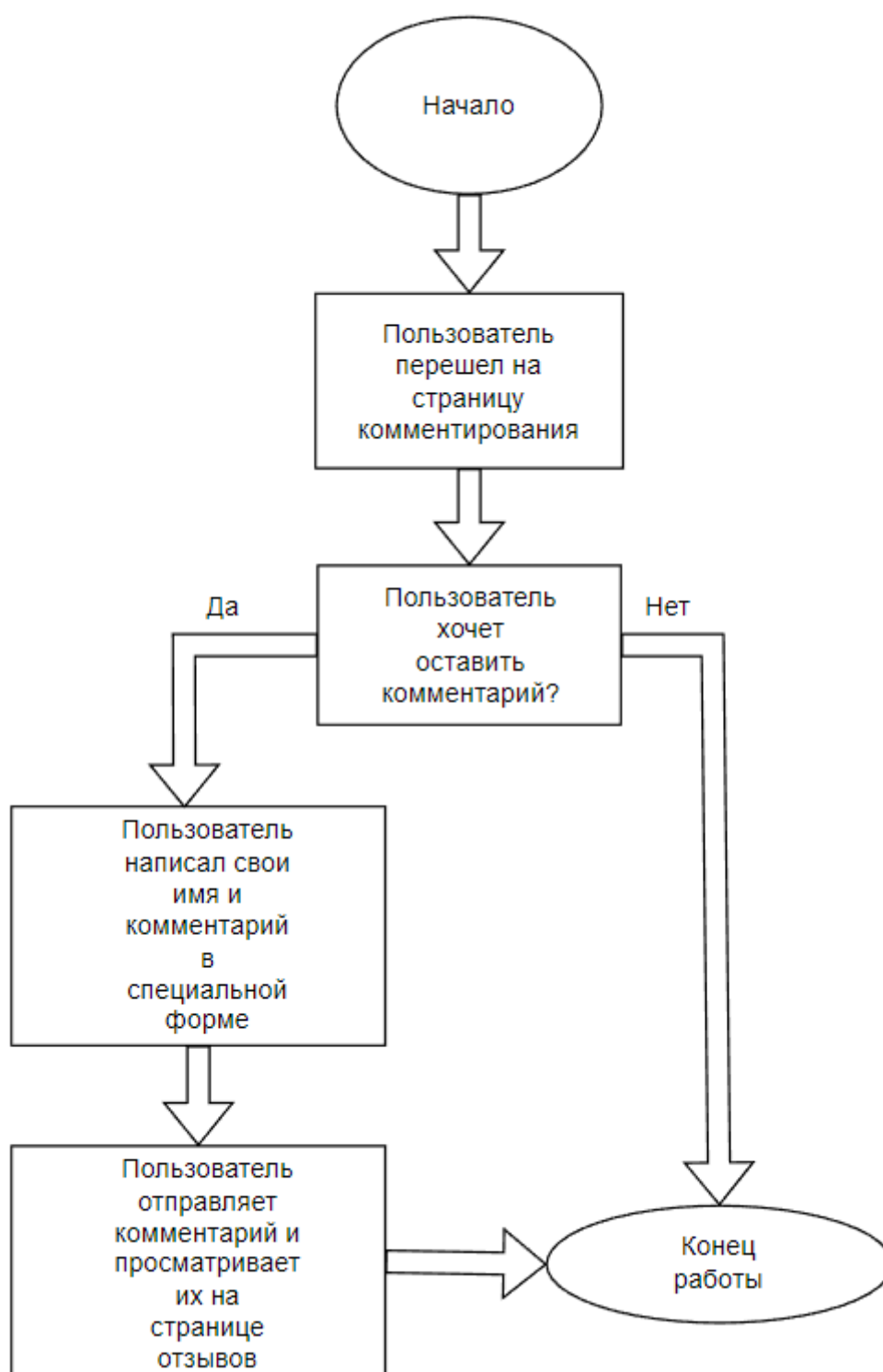


Рис. 9 Комментирование пользователем

После того как пользователь открыл сайт или закончил работать с системой он может перейти в раздел комментариев через кнопку «Перейти на

страницу с комментариями». Авторизация для этого не требуется. Там он может посмотреть последние шесть комментариев других пользователей или же перейти на страницу написания комментария через кнопку «Оставить комментарий». Там в специальной форме он заполняет имя пользователя и написать свой комментарий. Нажав на кнопку «Отправить», он переходит обратно на страницу с отзывами и может посмотреть его.

2.5. Требования к программному обеспечению системы

Для выполнения системой заявленных функций требуется наличие дополнительного ПО.

База данных для информационной системы реализована в СУБД (системе управления базами данных) SQLite DB, а в качестве сервера фреймворк Flask.

Требования к ПО клиентской части системы:

1. Операционная система:

- Windows 7 или более поздняя версия;
- Linux: Ubuntu 10.04 или более поздняя версия;
- Mac OS X 10.6 или более поздняя версия;

2. Браузер с поддержкой JavaScript:

- Google Chrome;
- Mozilla Firefox;
- Яндекс браузер;
- Microsoft Edge.

Требования к ПО, установленному на серверном аппаратном обеспечении:

3. Операционная система:

- Windows 7 или более поздняя версия;
- Linux: Ubuntu 10.04 или более поздняя версия;
- Mac OS X 10.6 или более поздняя версия.

4. Веб-сервер:

- Flask версии 2.3.2 и выше;
- SQLite DB.

Для создания клиентской и серверной части системы используются:

- Фреймворк Flask для программирования серверной части системы, использующий язык Python.
- Visual Studio Code - текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

2.5. Требования к техническому обеспечению системы

Для выполнения своих поставленных функций, система предъявляет следующие требования к ПО пользователя:

1. Операционная система: Windows 7 или выше, Linux, MacOS.
2. Процессор: с поддержкой архитектуры x86 или x64, с тактовой частотой не менее 1 ГГц.
3. Оперативная память: не менее 2 Гб.
4. Свободное место на жестком диске: не менее 500 мб.
5. Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge или Safari.
6. Наличие подключения к интернету с достаточной скоростью для работы с API. Также для работы с конкретным API могут потребоваться дополнительные библиотеки или пакеты программного обеспечения, которые необходимо установить в системе.
7. Клавиатура. Механическое устройство ввода текстовой информации.
8. Мышь. Механическое устройство управления курсором и отдачи различных команд компьютеру, путем взаимодействия с различными графическими элементами интерфейса: кнопки, ползунки, контекстные меню и т.д.

9. Монитор. Устройство оперативной визуальной связи с пользователем и интерпретации результатов работы системы, поддерживает возможность вывода цветного изображения

2.6. Требования к методическому обеспечению

При разработке данной информационной системы и создании документации на неё, необходимо руководствоваться следующими нормативными документами:

- ГОСТ 34.602-2020. Техническое задание на создание автоматизированной системы; [8]
- ГОСТ Р 59793–2021. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания; [9]
- ГОСТ 34.201-2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем. [10]

2.7. Требования к математическому обеспечению

В разрабатываемой ИС используется следующая формула:

$$F = k \cdot w_1 + p \cdot w_2 + o \cdot w_3 + a \cdot w_4 + 50\%.$$

Данная формула является предварительной и нуждается в экспертной оценке.

Обозначения используемых данных в формуле:

w_1, w_2, w_3, w_4 - весовые коэффициенты, k - конкуренты, p - количество людей, o - организации и постройки, a - аренда.

Обоснование используемой формулы:

Для системы была выбрана линейная формула, с использованием весовых коэффициентов, так как все получаемые данные имеют разное влияние на конечный результат.

Распределение весовых коэффициентов:

W1 для k. Берется значение (-1), так как каждый конкурент либо переманивает клиентов, либо передает их, если не может удовлетворить их потребности.

W2 для p. Берется значение (0,015) так как от количества людей зависит то, будут ли у СЕ посетители. Значение дробное, так как плотность населения в Сургуте составляет больше 1000 человек на квадратный километр. Коэффициент не равен (0,01), так как каждый проходящий человек в теории может стать клиентом заведения, а шанс такого события равен пятидесяти процентам, поэтому добавляется (0,005).

W3 для o. Берется значение (0,5) так как все организации привлекают людей, однако шанс того, что они станут клиентами равен пятидесяти процентам.

W4 для a. Берется значение (-0,0001) так как средняя цена аренды помещения под кафе или другое заведение составляет примерно от 1000 до 1500 рублей за квадратный метр. А площадь помещения необходимая даже для среднего размера заведения составляет примерно 100 квадратных метров, то выходит, что средняя цена аренды варьируется от 100000 рублей. Поэтому необходимо, чтобы конечный результат обработки этого показателя был адекватным в соответствии с другими данными, нужно уменьшить его до показаний других видов используемых данных.

Добавление 50% обуславливается тем, что любое новое заведение имеет 50% шанс на успех, без предварительного анализа. То есть шанс того, что постройка будет пользоваться популярностью равен шансу того, что заведение придется закрыть спустя некоторое время.

ГЛАВА 3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

3.1. Таблица SWOT-анализа

SWOT-анализ – это метод комплексной оценки факторов, напрямую и косвенно влияющих на бизнес.

SWOT расшифровывается так:

Strengths – сильные стороны компании,

Weakness – слабые стороны компании, Opportunities – возможности внешней среды,

Threats – угрозы внешней среды.

В таблице 2 представлен SWOT-анализ разрабатываемой ИС.

Табл. 2. SWOT-таблица

	Возможности	Угрозы
Внешние	Использование API у известной компании, может повысить надежность продукта в лице потребителей	Использование API у большинства информационно-географических серверов является платным, а бесплатные версии ограничены, что чревато задержками в работе.
Внутренние	Появление нового вида анализа геоданных.	Сложности в сборе данных, если они недоступны в общем доступе.

Из таблицы 2 видно, что внешние возможности «CE-helper» для продвижения – это то, что, используя API от Яндекс Карты, которые является одной из самых известных информационно-картографической службой в России [19] будет доверие пользователей, так как информация актуальна. А внутренние

возможности — это то, что такой вид анализа геоданных пока нигде не реализован. Внешней угрозой является, то, что использование API на бесплатной основе ограничено количеством запроса и как только они превысят лимит, сервис перестанет работать. А внутренней угрозой будет недоступность информации, так как некоторые данные защищены от свободного доступа.

3.2. Концептуальная модель предметной области

Диаграмма предметной области занимает центральное место в проектировании системы, показывая ее структуру.

Диаграмма представляет собой абстракции предметной области и их связи (статические отношения), существующие между ними.

На концептуальной модели предметной области отображены основные сущности и связи между ними (рис.10)

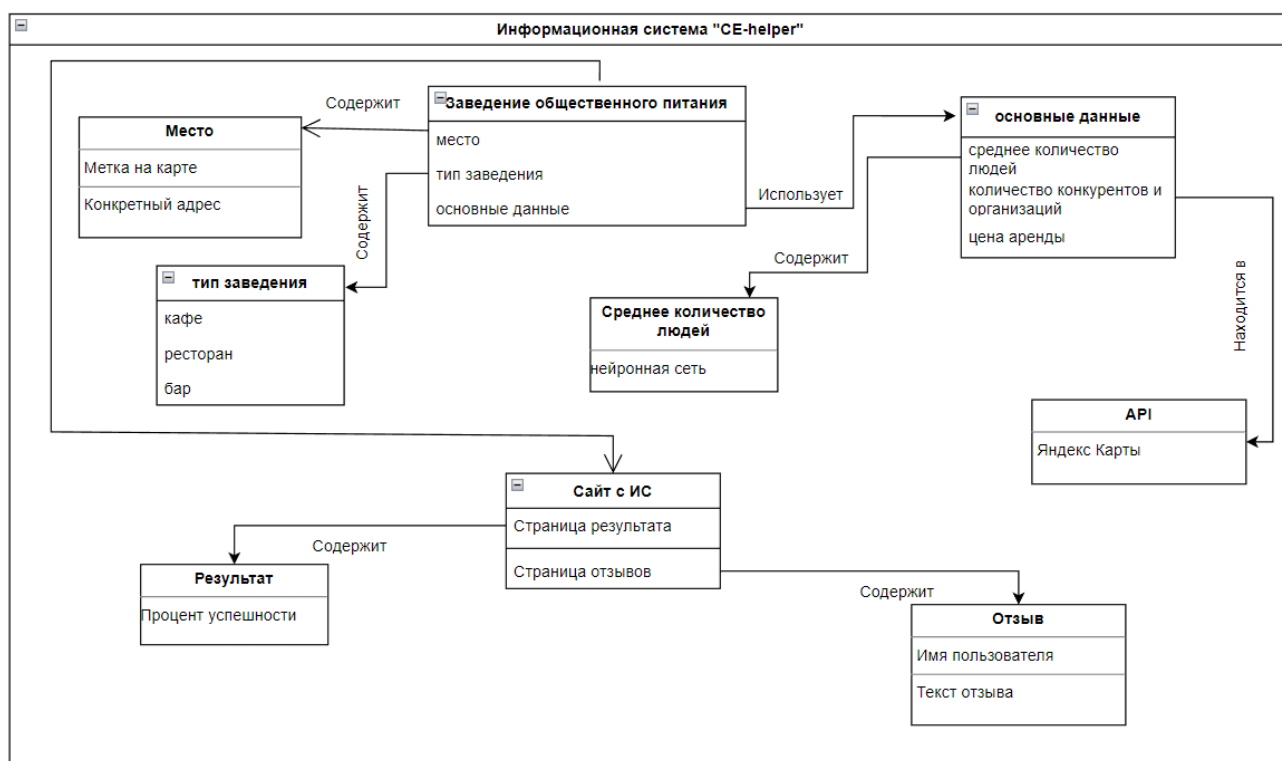


Рис. 10 Концептуальная модель предметной области

В концептуальной модели предметной области отображено взаимодействие главной сущности «Заведение общественного питания» и основных связанных с ним понятий. Первое из понятий это «Место». Оно содержит в себе метку на карте, которая содержит конкретный адрес. Следующее

— это «тип заведения», отражающий следующие атрибуты: кафе, ресторан, бар. «Основные данные» отражает собираемые данные (среднее количество людей, количество конкурентов и организаций, цена аренды). Каждый из них содержит в себе метод сбора:

- Для подсчета количества людей была использована нейросеть, считывающая людей по данным с видеокамер;
- Количество конкурентов и организаций получается при помощи запросов через систему API от Яндекс Карт;
- Цена аренды заносится вручную в базу данных, так как нет никакого способа получать данные напрямую.

Сайт с ИС содержит в себе страницу результата и страницу отзывов, на которых отображаются соответственно процент успешности и пользовательский комментарий.

3.3. Use case диаграмма

Use Case – это сценарная техника описания взаимодействия. С помощью Use Case может быть описано и пользовательское требование, и требование к взаимодействию систем, и описание взаимодействия людей и компаний в реальной жизни [10].

На рис. 11 отображена диаграмма вариантов использования системы пользователем

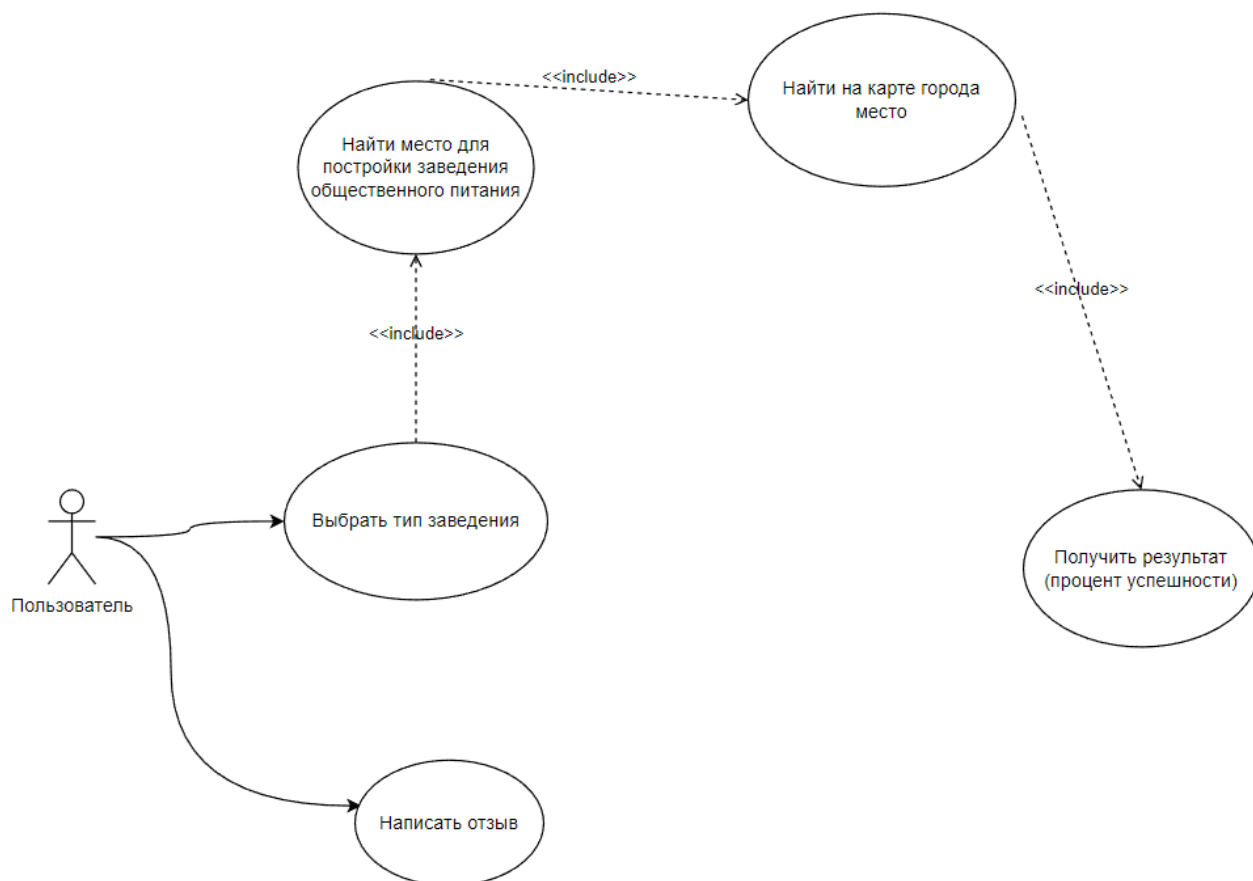


Рис. 11. Use case диаграмма

При помощи диаграммы use case можно увидеть возможности пользователя при работе с системой. Чтобы получить результат в виде процента успешности, необходимо выбрать тип заведения из соответствующего фильтра с выбором заведения. Далее он выбирает место путем «клика» по карте города и переходит на страницу с результатом. Также пользователь может написать отзыв.

3.4. Описание прецедентов

Название прецедента:

Поиск места для постройки заведения общественного питания

Основной исполнитель:

Пользователь

Заинтересованные лица и их требования:

1. Пользователь. Требуется показать процент успешности постройки заведения в выбранном им месте.

2. Владелец сервиса. Хочет получать достоверные сведения о собираемых данных от Яндекс. Карты.

Предусловия:

Пользователь выбрал место для постройки методом «клика» по карте.

*Иногда нужно место нужно вводить вручную

Постусловия:

Пользователь получил удовлетворивший его процент в результате.

Основной успешный сценарий:

1. Пользователь зашел на сайт.
2. Пользователь выбрал место постройки путем ввода определенного адреса или «клика» по карте.
3. Пользователь выбрал тип заведения.
4. Веб-сервис выдает результат в виде процента.
5. Пользователя устраивает результат.
6. Пользователь оставляет или не оставляет отзыв.
7. Пользователь покидает сайт.

Основной неуспешный сценарий:

1. Пользователь зашел на сайт.
2. Пользователь выбрал место постройки одним из двух методов.
3. Пользователь выбрал тип заведения.
4. Веб-сервис выдает результат в виде процента.

5. Пользователя не устраивает результат.
6. Он выбирает место путем ввода определенного адреса.
7. Веб-сервис выдает результат в виде процента.
8. Пользователя не устраивает результат.
9. Пользователь оставляет или не оставляет отзыв.
10. Пользователь покидает сайт.

Сценарий с не обновлённой информацией:

1. Пользователь зашел на сайт.
2. Пользователь выбрал место постройки одним из двух методов.
3. Пользователь выбрал тип заведения.
4. Веб-сервис выдает результат в виде процента.
5. Пользователя устраивает результат.
6. Пользователь покидает сайт.
7. Пользователь проверяет полученную информацию в реальности.
8. Данные не совпадают.
9. Пользователь не удовлетворен результатом.
10. Пользователь оставляет или не оставляет отзыв.

3.5. BPMN диаграмма

Диаграмма BPMN (англ. Business Process Model and Notation, нотация и модель бизнес-процессов) — система условных обозначений (нотация) и их описания для моделирования бизнес-процессов.

На рис. 12 отображена диаграмма BPMN для системы «CE-helper»

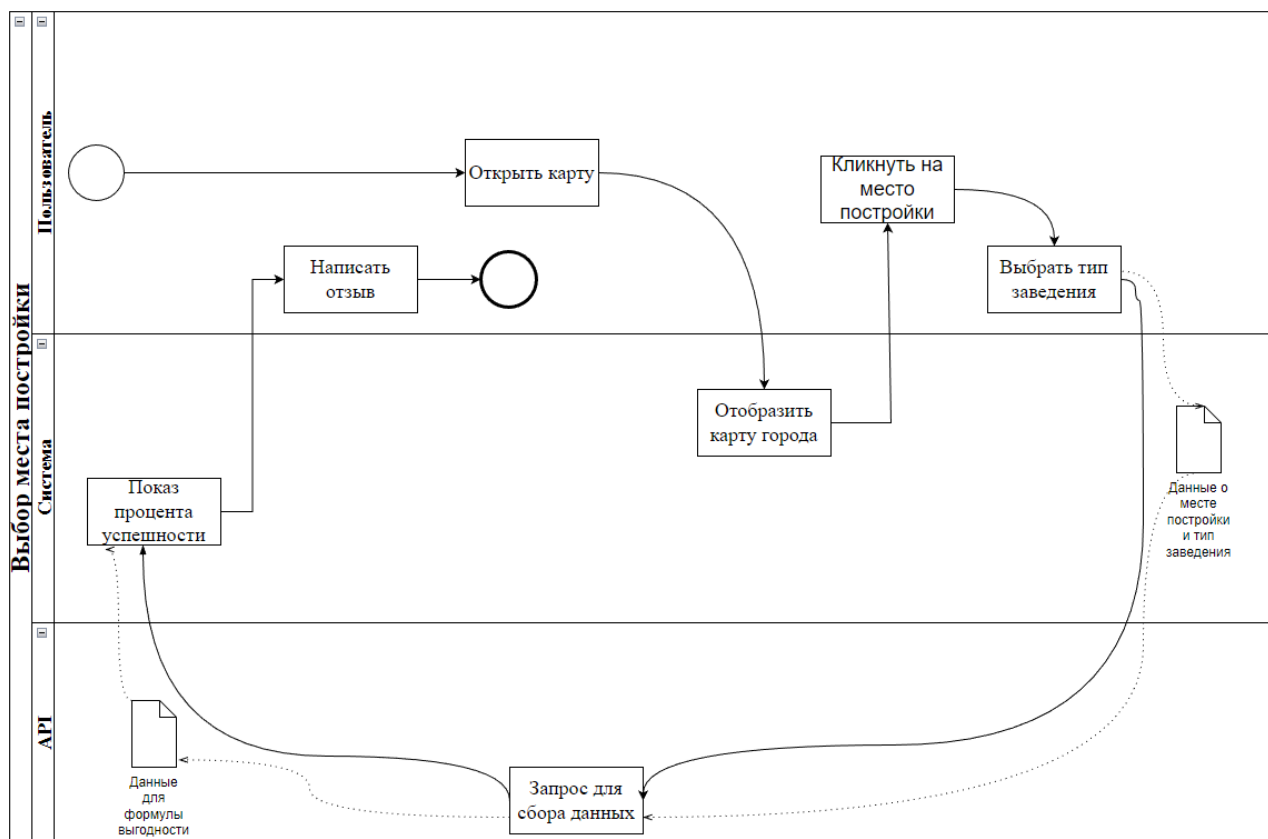


Рис. 12. BPMN диаграмма

Как видно из диаграммы пользователь начинает с действия открыть карту, после которого система отображает ему карту города. Далее он кликает на место постройки и выбирает тип заведения. Одновременно с этим отправляются данные о месте и типе заведения в систему API, откуда приходит json файл отображающий в системе процент успешности.

3.6. Диаграмма классов

Для того чтобы представить статическую структуру модели системы используется диаграмма классов. Данная диаграмма служит для отображения различных взаимосвязей между отдельными сущностями, такие как объекты и подсистемы, и для описания их внутренней структуры, а также типы отношений между ними.

На рис. 13 представлена диаграмма классов, сущности и типы связи между ними.

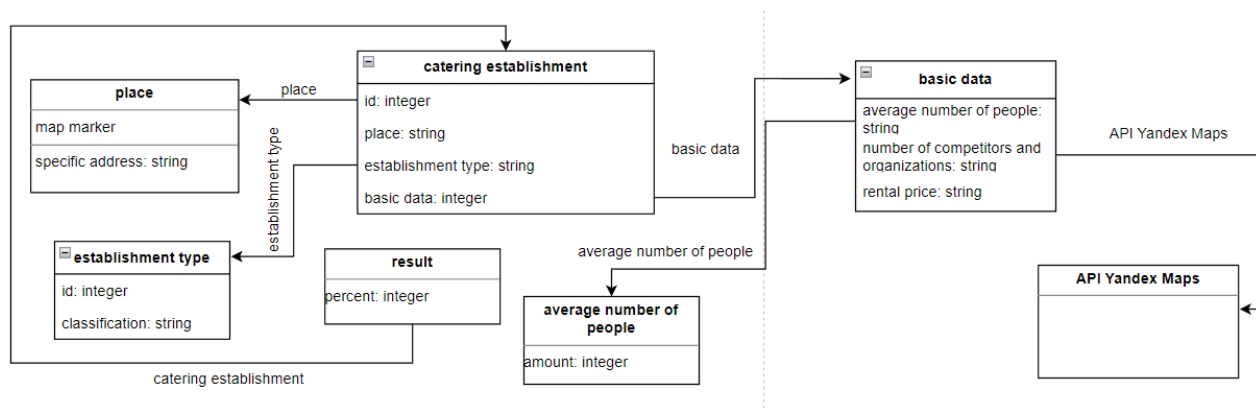


Рис. 13. Диаграмма классов «CE-helper»

На диаграмме отображены следующие сущности: catering establishment, place, establishment type, result, basic data, average number of people, API Yandex Maps. Их описание представлено в таблице (Таблица 3).

Табл. 3. Описание сущностей в ИС «CE-helper»

Сущность	Наименование
Catering establishment	Заведение общественного питания
Place	Место
Establishment type	Тип заведения
Basic data	Основные данные
Average number of people	Среднее количество людей

Сущность	Наименование
Result	Результат
API Yandex Maps	API Яндекс Карты

Сущности содержат атрибуты. Атрибуты сущностей описаны в таблице (Таблица 4).

Табл. 4. Описание атрибутов сущностей в ИС «CE-helper»

Сущность	Атрибут	Наименование	Тип	Обязательный	Мах длина	Ссылка на сущность
Catering establishment	id	id	ID	1		
	Establishment type	Тип заведения	STRING	1		Establishment type
	Basic data	Основные данные	INTEGER	1		Basic data
Place	Map marker	Метка на карте	-	1		
	Specific address	Конкретны й адрес	STRING	1		
Establishment type	id	id	INTEGER	1		

	Classification	Тип	STRING	1		
Сущность	Атрибут	Наименование	Тип	Обязательный	Мах длина	Ссылка на сущность
Basic data	Average number of people	Среднее количество людей	STRING	1		Average number of people
	Number of competitors and organizations	Количество конкурентов и организаций	STRING	1		API Yandex Maps
	Rental price	Цена аренды	STRING	1		API Avito

Result	Percent	Процент	INTEGER	1		Catering establishment
Average number of people	Amount	Количество	INTEGER	1		
API Avito	-	-	-	1		
Сущность	Атрибут	Наименование	Тип	Обязательный	Мах длина	Ссылка на сущность
API Yandex Maps	-	-	-	1		

3.7. Диаграмма компонентов

Диаграмма компонентов позволяет определить архитектуру проектируемой системы, установив зависимости между компонентами. Основные графические элементы данной диаграммы – компоненты, интерфейсы, и также зависимости между ними [20].

На рис. 13 отображена диаграмма компонентов, которая отображает работу основных элементов системы.

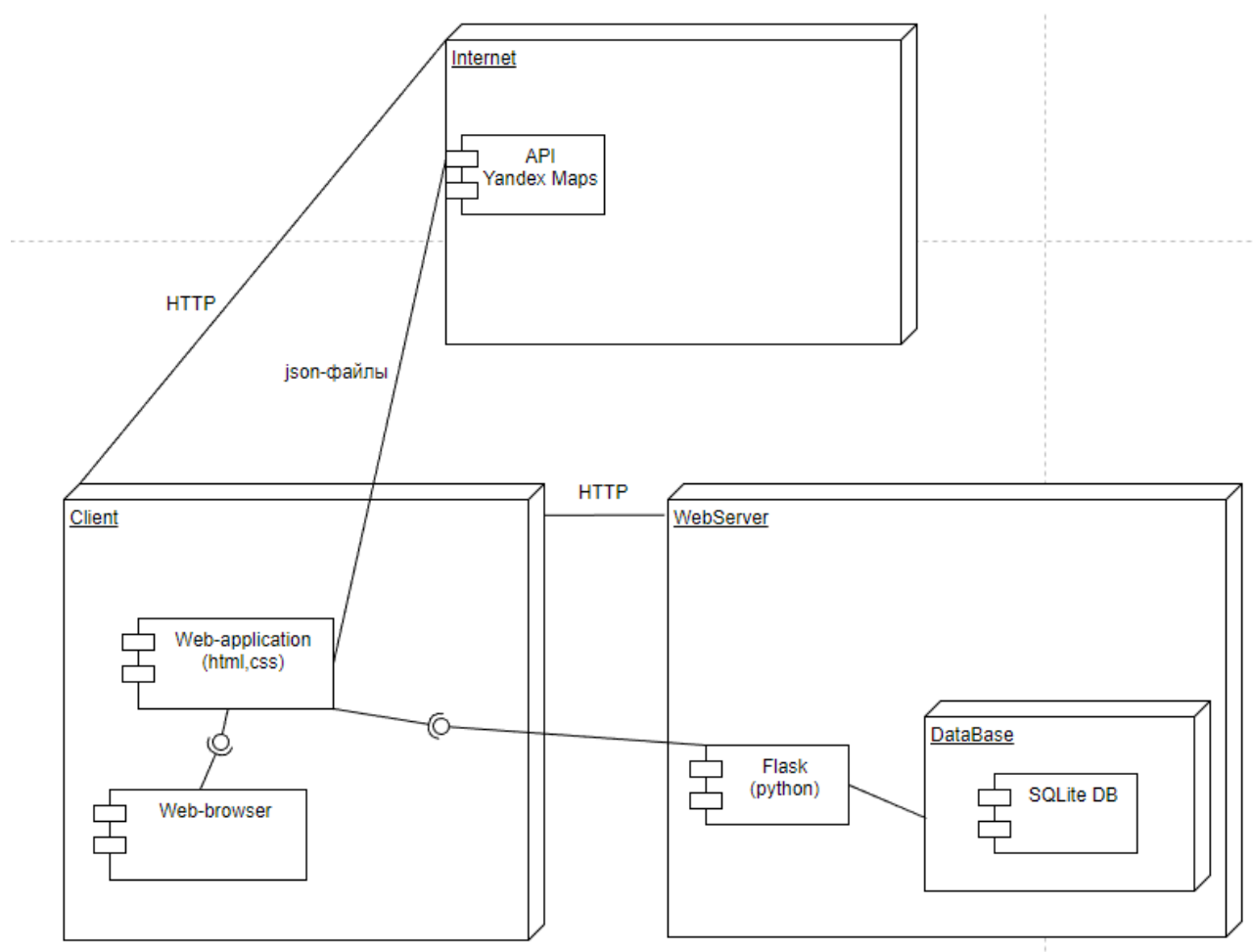


Рис. 13 Диаграмма компонентов

На диаграмме видно 3 больших блока:

- Client;
- WebServer;

- Internet.

В блоке Client атрибут Web-browser означает браузер пользователя, web-application – это веб-приложение написанное на языке разметки html и стилем каскадных таблиц css.

Блок Client соединяется при помощи http подключения с блоками WebServer и Internet. В блок Internet он отправляет запрос на API Yandex Maps и в ответ получает json файл. А в блоке WebServer основной компонент это фреймворк Flask работающий на языке программирования python, который берет данные из Database созданной при помощи СУБД SQLite DB.

ГЛАВА 4. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

В данной главе отображен пользовательский интерфейс ИС и описаны основные сценарии взаимодействия с ним.

4.1. Пользовательский интерфейс

В данном пункте отображены основные страницы сайта, с помощью которых пользователь взаимодействует с информационной системой.

На (рис. 14) показана главная страница сайта, на которой можно сделать два действия:

1. Сразу начать работу, нажав на кнопку, перед этим выбрав необходимый город (Пока реализована только работа с городом Сургут);
2. Перейти и посмотреть комментарии, перед тем как воспользоваться веб-приложением.

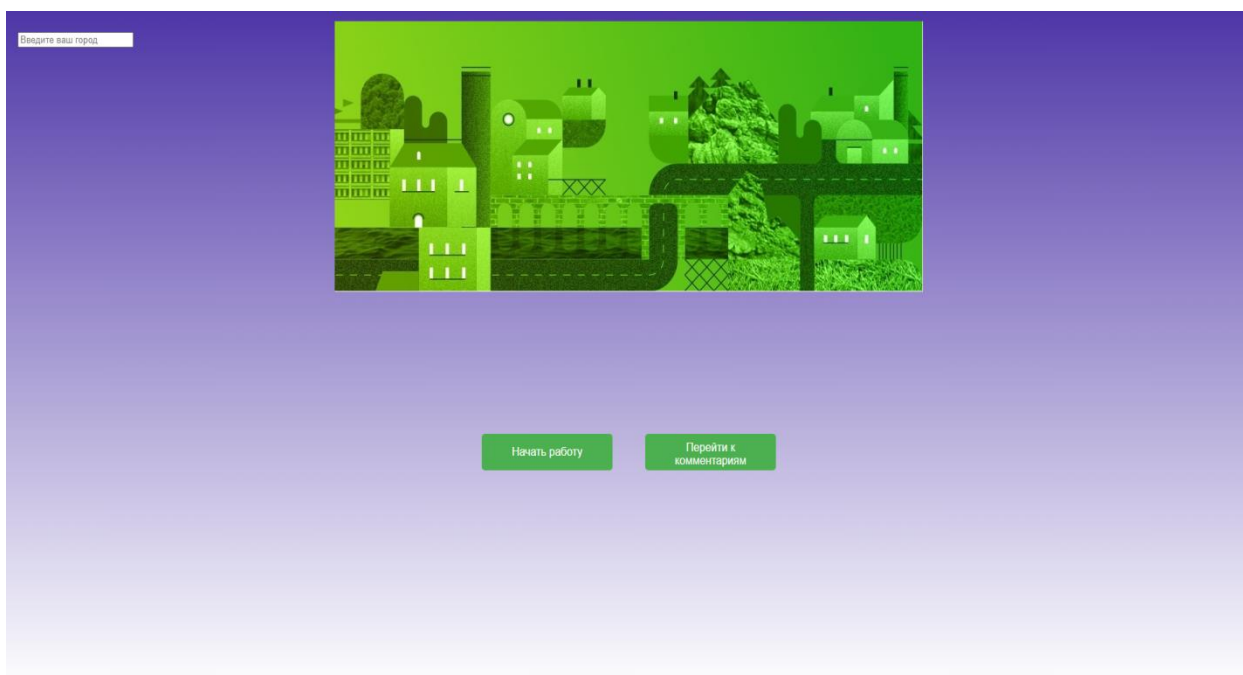


Рис. 14 Главная страница

На (рис.15) представлена рабочая страница сайта подключается карты от Яндекс Карты через систему API. Есть фильтр по заведениям в виде выпадающего меню (рис. 16).

Основной алгоритм работы на рабочей странице представляет собой следующие шаги:

1. Выбрать в фильтр необходимый тип заведения (кафе, ресторан или бар).
2. «Кликнуть» на любое место в городе, тем самым отправив запросы через API в Яндекс Карты и в базу данных.
3. Нажать на кнопку «Получить результат».

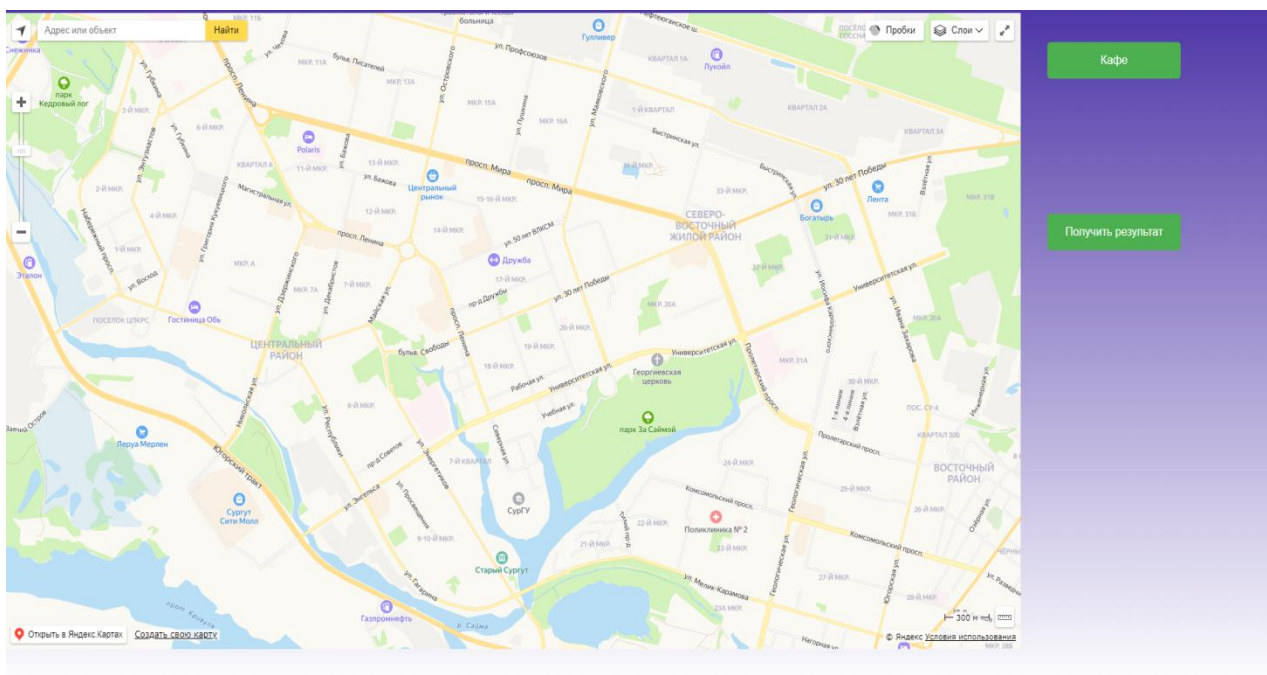


Рис. 15 Рабочая страница

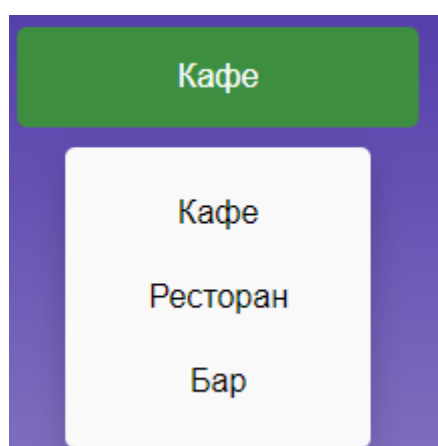


Рис. 16 Выпадающее меню

На (рис. 17) представлен результат работы системы по следующему адресу: Ханты-Мансийский автономный округ, Сургут, улица Чехова, 10/2.

Всего идет пять основных форм, каждая из которых отображает значения выбранных показателей.

Порядок отображения идет слева направо:

Конкуренты. В данной форме показывается значение в виде количества конкурентов в радиусе одного километра. Радиус обуславливается тем, что человек в большинстве случаев в городской местности преодолевает данное расстояние за десять или пятнадцать минут. И если человеку необходимо поесть где-либо, он ищет ближайшее заведение от себя. Если же до заведения нужно добираться больше, чем двадцать минут, скорее всего он не выберет данное заведение, кроме особых случаев, когда он намеренно собирался в именно то заведение. Также значение уже сразу умножается на выбранный числовой коэффициент.

Организации. В данной форме показывается количество организаций, также расположенных в радиусе одного километра. Если внутри этих зданий отсутствует своя столовая, то рабочий человек во время обеда, будет искать заведение общественного питания. Поэтому строящееся заведение должно быть в радиусе одного километра, чтобы человек имел возможность выбрать именно данное заведение.

Супермаркеты. Также показываются в радиусе одного километра, потому что, данный вид строений, как и организации собирают людей рядом.

Количество людей. Данные по количеству людей были собраны через видеокамеры и занесены в базу данных, которая подключена к системе. Выбор значения количества людей из базы данных зависит от того, какая точка для постройки заведения была выбрана на карте. И ближайшее значение количества определяется наименьшей разницей в полученных и хранимых координатах.

Цена аренды. Также как количество людей, выбирается по наименьшей разнице в координатах.

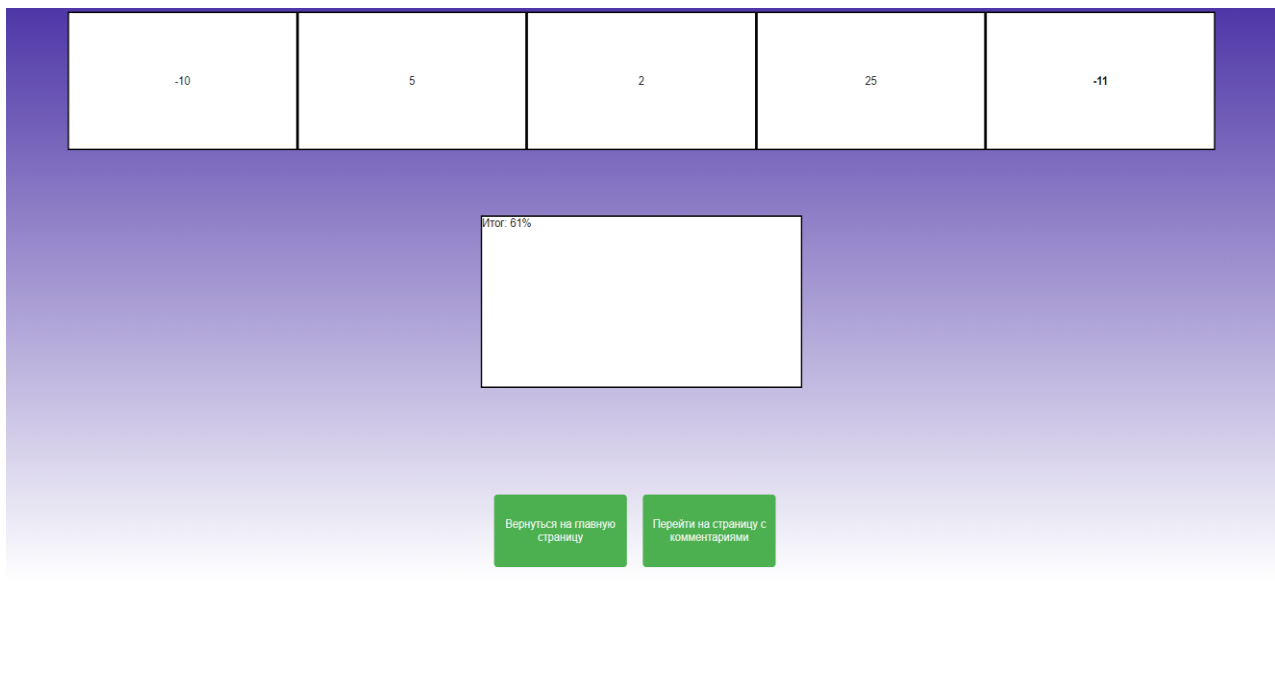


Рис. 17 Страница результата

На (рис. 13) представлены формы, в которых буду оставляться последние из комментариев пользователей. Чтобы написать комментарий регистрация не требуется.



Рис. 18 Страница с комментариями

4.2. Запрос через API и запрос к базе данных

На (рис. 19) представлен json файл, который представляет собой собранные данные через запросы к API Яндекс Карт, а также выводы из базы данных.

▶ {response: {...}}	Geocoder.html:27
▶ (2) [61.253988833384945, 73.40496177233874]	event reverse_geocode.js:15
▶ (2) [{...}, '2,595']	event reverse_geocode.js:70
гипермаркет ▶ {type: 'FeatureCollection', properties: {...}, features: Array(3)}	event reverse_geocode.js:57
организации ▶ {type: 'FeatureCollection', properties: {...}, features: Array(10)}	event reverse_geocode.js:65
▶ Promise {<pending>}	event reverse_geocode.js:45
кафе ▶ {type: 'FeatureCollection', properties: {...}, features: Array(10)}	event reverse_geocode.js:49
>	

Рис. 19 JSON файл

На (рис. 14) представлен json файл, который представляет собой собранные данные через запросы к API Яндекс Карт, а также выводы из базы данных.

Первая строка после {response: {...}} представляет собой массив из двух чисел, а именно координаты выбранной точки на карте.

Далее идет количество людей ближайшее к выбранной точке, путем вывода значения из базы данных.

Затем идет три коллекции с названиями: гипермаркет, организации и кафе.

Это и есть запросы к API. Каждый из них возвращает коллекцию данных, из которой для работы системы извлекается только количество вернутых объектов.

ЗАКЛЮЧЕНИЕ

В результате выпускной квалификационной работы были выполнены все поставленные задачи.

Изучена предметная область заведений общественного питания, произведен анализ аналогов ИС, выделены их преимущества и недостатки. Были описаны все необходимые требования к разрабатываемой системе (функциональное, информационное, лингвистическое, программное, техническое, методическое, математическое). Проведен SWOT- анализ и построены все необходимые модели и диаграммы такие как: модель предметной области, BPMN диаграмма, use-case диаграмма, спецификация прецедентов, диаграмма классов, диаграмма компонентов. Также был разработан пользовательский интерфейс.

В ходе разработки информационной системы были выполнены следующие задачи:

- Был создан сайт под ИС;
- Определены все необходимые модули программы (модуль с запросами к сервису API от Яндекс Карт, модуль с сервером сайта и модуль с подключением сервиса к базам данных)
- Был разработан код, который реализовывал работу всех модулей на сайте.

В результате была спроектирована и разработана информационная система для определения выгодности постройки заведения общественного питания.

СПИСОК ЛИТЕРАТУРЫ

1. За 2017-2021 гг численность предприятий общественного питания в России выросла на 2%: с 186,9 тыс до 190,4 тыс. // Busines Stat URL: <https://marketing.rbc.ru/articles/13430/> (дата обращения: 11.06.2023).
2. Google Maps URL: <https://www.google.com/maps> (дата обращения: 11.06.2023).
3. Яндекс Карты URL: <https://yandex.ru/maps> (дата обращения: 11.06.2023).
4. 2 GIS URL: <https://2gis.ru/> (дата обращения: 11.06.2023).
5. Ларман К. Применение UML 2.0 и шаблонов проектирования. - 3-е изд. - Диалектика, 2019. - 736 с.
6. Эрик Э. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем.: Пер. с англ. - М.: 000 "И.Д. Вильямс", 2011. - 448 с. : ил. - Парал. тит. англ.
7. Элти Д., Кумбс М. Экспертные системы: концепции и примеры / Д. Элти, М. Кумбс. - М.: Финансы и статистика, 2006. – 236 с.
8. ГОСТ 34.602-2020 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс]: Учебно-методические материалы – URL: <https://docs.cntd.ru/document/1200181804> (дата обращения: 10.06.2023).
9. ГОСТ Р 59793–2021 Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. [Электронный ресурс]: Учебно-методические материалы – URL: <https://protect.gost.ru/v.aspx?control=8&baseC=-1&page=0&month=-1&year=-1&search=&RegNum=1&DocOnPageCount=15&id=231287> (дата обращения: 08.04.2023).
10. ГОСТ 34.201-2020 Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение

- документов при создании автоматизированных систем. [Электронный ресурс]: Учебно-методические материалы – URL: <https://docs.cntd.ru/document/1200181803> (дата обращения: 08.04.2023).
11. Проектирование информационных систем. Лекции. Модель вариантов использования [Электронный ресурс]: Учебно-методические материалы – Электрон. текстовые дан. – URL: <https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12> (дата обращения: 08.04.2023).
12. Шаблоны корпоративных приложений / Мартин Фаулер, Дейвид Райс, Мэттью Фоммел, Эдвард Хайет, Роберт Ми, Рэнди Стаффорд, - 2-е изд. – М.: Вильямс, 2016. – 544 с.
13. Русакова, О. В. Функции и особенности системы общественного питания / О. В. Русакова. — Текст : непосредственный // Молодой ученый. — 2012. — № 7 (42). — С. 121-123. — URL: <https://moluch.ru/archive/42/5116/> (дата обращения: 12.06.2023).
14. Анализ рынка общественного питания в России в 2017-2021 гг, прогноз на 2022-2026 гг. Перспективы в условиях санкций // РБК URL: <https://marketing.rbc.ru/research/47269/> (дата обращения: 07.06.2023).
15. Число ресторанов и баров в России превысило доковидный уровень // РИА Новости URL: <https://ria.ru/20220928/restoran-1819971484.html> (дата обращения: 07.06.2023).
16. Обзор онлайн-картографических сервисов // 3DNews URL: <https://3dnews.ru/587095> (дата обращения: 07.06.2023).
17. Классификация предприятий общественного питания // КонсультантПлюс URL: https://www.consultant.ru/document/cons_doc_LAW_135474/53a2c9fb24fa23a594c4eb96dac214ef0d67e48f/ (дата обращения: 07.06.2023).
18. Продвижение ресторана в интернете // АЛЕКСАНДР БЕРНИКОВ URL: <https://abernik.ru/blog/prodvizhenie-restorana-v-internete/> (дата обращения: 07.06.2023).

- 19.Статистика Яндекса в 2023 году (обновлено) // ИНКЛИЕНТ URL: <https://inclient.ru/yandex-stats/> (дата обращения: 08.06.2023).
- 20.Маркелов В. В., Власов А. И., Камышная Э. Н. Визуальные методы системного анализа при управлении качеством изделий электронной техники // НиКа. 2014. №. URL: <https://cyberleninka.ru/article/n/vizualnye-metody-sistemnogo-analiza-pri-upravlenii-kachestvom-izdeliy-elektronnoy-tehniki> (дата обращения: 17.05.2023).
- 21.Welcome to Flask — Flask Documentation (2.3.x) // URL: <https://flask.palletsprojects.com/en/2.3.x/> (дата обращения: 20.05.2023).
- 22.SQLite Documentation // URL: <https://www.sqlite.org/docs.html> (дата обращения: 20.05.2023).
- 23.Страшко М.Т, Ситникова С.Ю. СОВРЕМЕННЫЕ ПОДХОДЫ К СОЗДАНИЮ ВЕБ-САЙТОВ // НАУЧНО-ТЕХНИЧЕСКОЕ И ЭКОНОМИЧЕСКОЕ СОТРУДНИЧЕСТВО СТРАН АТР В XXI ВЕКЕ. - 2022. - №2. - С. 296-300.
- 24.Сысолетин, Е. Г. Разработка интернет-приложений : учеб. пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под науч. ред. Л. Г. Доросинского. — М. : Издательство Юрайт, 2019. — 90 с
- 25.Голицына, О.Л. Информационные системы: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2016. - 352 с.
- 26.Федотова, Е.Л. Информационные технологии и системы: Уч.пос / Е.Л. Федотова. - М.: Форум, 2018. - 149 с.
- 27.Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем / Э. Р. Ипатова, Ю. В. Ипатов. – Москва : ФЛИНТА, 2021. – 256 с. – ISBN 9785893499780. – URL: http://biblioclub.ru/index.php?page=book_red&id=79551 (дата обращения: 03.03.2023). – Текст : электронный
- 28.Web-технологии ; Составитель: Малышева Е. Н.. – Кемерово : Кемеровский государственный университет культуры и искусств (КемГУКИ), 2014. – 104 с. – URL:

http://biblioclub.ru/index.php?page=book_red&id=275540 (дата обращения: 02.03.2023). – Текст : электронный.

29. Никулова, Г. А. Web-программирование / Г. А. Никулова, В. Р. Субботин. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2017. – 58 с. – ISBN 9785885268349. – URL: http://biblioclub.ru/index.php?page=book_red&id=577452 (дата обращения: 09.03.2023). – Текст : электронный.
30. Рыбальченко, М. В. Архитектура информационных систем : учебное пособие для вузов : [для студентов, обучающихся по направлению "Информационные системы и технологии"] / М. В. Рыбальченко ; Южный федеральный университет. – Москва : Юрайт, 2019. – 91 с. – (Университеты России). – Текст : непосредственный.
31. Стасышин, В. М. Проектирование информационных систем и баз данных : учебное пособие / В. М. Стасышин ; В. М. Стасышин ; Новосиб. гос. техн. ун-т. – Новосибирск : Изд-во НГТУ, 2012. – 97, [2] с. – URL: <http://library.kuzstu.ru/meto.php?n=178035&type=nstu:common> (дата обращения: 16.04.2023). – Текст : электронный
32. Жукова, Ю. А. Информационные технологии в сфере общественного питания / Ю. А. Жукова // В мире научных открытий : Материалы V Международной студенческой научной конференции, Ульяновск, 20–21 мая 2021 года. Том III. Часть 1. – Ульяновск: Ульяновский государственный аграрный университет им. П.А. Столыпина, 2021. – С. 169-172. – EDN CEEUTB.
33. Сергеева, И. И. Информационные системы и технологии в общественном питании / И. И. Сергеева, Е. А. Реденкова // Экосистема цифровой экономики: проблемы, реалии и перспективы : Сборник научных трудов национальной научно-практической конференции, Орел, 23–25 апреля 2018 года / Под редакцией Л.И. Малявкиной. – Орел: Орловский государственный университет экономики и торговли, 2018. – С. 172-178. – EDN YXKLAR.

34. Габараева, З. И. Информационные технологии в общественном питании / З. И. Габараева, Д. Г. Мустафаева // Современные тенденции развития информационных технологий в научных исследованиях и прикладных областях : Сборник докладов III Международной научно-практической конференции, Владикавказ, 28–29 апреля 2022 года. – Владикавказ: Северо-Кавказский горно-металлургический институт (государственный технологический университет), 2022. – С. 21-23. – EDN VGVJJKG.

ПРИЛОЖЕНИЕ 1. Модуль event_reverse_geocode.js

```
ymaps.ready(init);

function init() {
    var myPlacemark,
        myMap = new ymaps.Map('map', {
            center: [61.24942013169517, 73.40993995227039],
            zoom: 14
        }, {
            searchControlProvider: 'yandex#search'
        });

    // Слушаем клик на карте.
    myMap.events.add('click', function (e) {
        var coords = e.get('coords');
        console.log(coords);
        // Если метка уже создана – просто передвигаем ее.
        if (myPlacemark) {
            myPlacemark.geometry.setCoordinates(coords);
        }
        // Если нет – создаем.
        else {
            myPlacemark = createPlacemark(coords);
            myMap.geoObjects.add(myPlacemark);
            // Слушаем событие окончания перетаскивания на метке.
            myPlacemark.events.add('dragend', function () {
                getAddress(myPlacemark.geometry.getCoordinates());
            });
        }
    });
}
```

```

    getAddress(coords);

    // const coords = [55.7522, 37.6156]; // Широта и долгота центра
поисковой области

    const category = 'кафе'; // Категория организаций для поиска (в данном
случае - заведения питания)

    // fetch(https://search-
maps.yandex.ru/v1/?apikey=${apiKey}&text=${category}&ll=${coords[1]},${coord
s[0]}&spn=0.05,0.05&lang=ru&type=biz&results=500&radius=${radius})

    // .then(response => response.json())
    // .then(data => {
    //   const features = data.features;
    //   // Обработка полученных объектов
    // })
    // .catch(error => console.error(error));

    fetch(`https://search-maps.yandex.ru/v1/?apikey=b774c439-0479-418c-
b490-
5eceae361a0f&text=${category}&ll=${coords[1]},${coords[0]}&bbox=${coords[1]
- 0.0287},${coords[0] - 0.1772}~${coords[1] + 0.0287},${coords[0] +
0.1772}&rspn=1&lang=ru`)

    .then((response) => {
      const filterData = response.json()

      console.log(filterData)
      return filterData;
    })
    .then((data) => {
      console.log('кафе', data);
      localStorage.setItem("Cafe", data.features.length)

```

```

    });

    fetch(`https://search-maps.yandex.ru/v1/?apikey=b774c439-0479-418c-
b490-
5eceae361a0f&text=%D0%B3%D0%B8%D0%BF%D0%B5%D1%80%D0%BC%D
0%B0%D1%80%D0%BA%D0%B5%D1%82%20%D0%A1%D1%83%D1%80%D0
%B3%D1%83%D1%82&ll=${coords[0]},${coords[1]}&bbox=${coords[1]
-
0.0287},${coords[0] - 0.1772}~${coords[1] + 0.0287},${coords[0] +
0.1772}&rspn=1&lang=ru`)

    .then((response) => {
        return response.json();
    })

    .then((data) => {
        console.log('гипермаркет', data);
        localStorage.setItem("Giper", data.features.length)
    });

    fetch(`https://search-maps.yandex.ru/v1/?apikey=b774c439-0479-418c-
b490-
5eceae361a0f&text=%20%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B
8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%20%D0%A1%D1%83%D1%80
%D0%B3%D1%83%D1%82&ll=${coords[0]},${coords[1]}&bbox=${coords[1]
-
0.0287},${coords[0] - 0.1772}~${coords[1] + 0.0287},${coords[0] +
0.1772}&rspn=1&lang=ru`)

    .then((response) => {
        return response.json();
    })

    .then((data) => {
        console.log('организации', data);
        localStorage.setItem("Orga", data.features.length)
    });

    axios.post("http://127.0.0.1:5000/coord", { coords })

```

```

.then((response) => {
    console.log(response.data);
    localStorage.setItem("People", response.data[1]);
})

```

```

.catch(error => {
    console.log(error)
    alert("Ошибка обращения к BackEnd: " + error)
});

```

```

    // peoples.innerText = response.data[1];
    // Создание метки.
    function createPlacemark(coords) {
        return new ymaps.Placemark(coords, {
            iconCaption: 'поиск...'
        }, {
            preset: 'islands#violetDotIconWithCaption',
            draggable: true
        });
    }

```

```

    // Определяем адрес по координатам (обратное геокодирование).
    function getAddress(coords) {
        myPlacemark.properties.set('iconCaption', 'поиск...');
        ymaps.geocode(coords).then(function (res) {
            var firstGeoObject = res.geoObjects.get(0);

            myPlacemark.properties

```

```

        .set({
            // Формируем строку с данными об объекте.
            iconCaption: [
                // Название населенного пункта или вышестоящее
                административно-территориальное образование.
                firstGeoObject.getLocalities().length                ?
                firstGeoObject.getLocalities() : firstGeoObject.getAdministrativeAreas(),
                // Получаем путь до топонима, если метод вернул null,
                запрашиваем наименование здания.
                firstGeoObject.getThoroughfare()                      ||
                firstGeoObject.getPremise()
            ].filter(Boolean).join(', '),
            // В качестве контента балуна задаем строку с адресом
            объекта.
            balloonContent: firstGeoObject.getAddressLine()
        });
    });
}

// Если метка уже создана – просто передвигаем ее.
if (myPlacemark) {
    myPlacemark.geometry.setCoordinates(coords);
}
// Если нет – создаем.
else {
    myPlacemark = createPlacemark(coords);
    myMap.geoObjects.add(myPlacemark);
    // Слушаем событие окончания перетаскивания на метке.
    myPlacemark.events.add('dragend', function () {
        getAddress(myPlacemark.geometry.getCoordinates());
    });
}

```



```

    });
}
getAddress(coords);

```

// Создание метки.

```

function createPlacemark(coords) {
    return new ymaps.Placemark(coords, {
        iconCaption: 'поиск...'
    }, {
        preset: 'islands#violetDotIconWithCaption',
        draggable: true
    });
}

```

// Определяем адрес по координатам (обратное геокодирование).

```

function getAddress(coords) {
    myPlacemark.properties.set('iconCaption', 'поиск...');
    ymaps.geocode(coords).then(function (res) {
        var firstGeoObject = res.geoObjects.get(0);

```

```

        myPlacemark.properties

```

```

            .set({

```

```

                // Формируем строку с данными об объекте.

```

```

                iconCaption: [

```

```

                    // Название населенного пункта или вышестоящее
                    административно-территориальное образование.

```

```

                    firstGeoObject.getLocalities().length ?

```

```

                    firstGeoObject.getLocalities() : firstGeoObject.getAdministrativeAreas(),

```

// Получаем путь до топонима, если метод вернул null,
запрашиваем наименование здания.

```
firstGeoObject.getThoroughfare()           ||  
firstGeoObject.getPremise()
```

```
    ].filter(Boolean).join(', '),  
    // В качестве контента балуна задаем строку с адресом  
    объекта.
```

```
        balloonContent: firstGeoObject.getAddressLine()  
    });  
});  
}  
});
```

```
// var sravnenie=document.getElementById("comparison")  
// sravnenie.onclick(()=>axios.get("http://127.0.0.1:5000/coord/"+coords)  
// .then((response) => onOk(response))  
// .catch(error => {  
//   console.log(error)  
//   alert("Ошибка обращения к BackEnd: " + error)  
//   })))  
}
```

ПРИЛОЖЕНИЕ 2. Модуль `init.py`

```
from flask import Flask, redirect, url_for, request, jsonify
```

```
from routes.main import mainpg
```

```
from routes.zapros_people import coords_bp
```

```
# инициализация классов
```

```
app = Flask(__name__)
```

```
app.register_blueprint(coords_bp)
```

```
app.register_blueprint(mainpg)
```

```
# запуск приложения
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True)
```

ПРИЛОЖЕНИЕ 3. Модуль main.py

```
from flask import Blueprint, render_template
# from coord_repository import
```

```
mainpg = Blueprint('mainpg', __name__)
```

```
@mainpg.route("/")
```

```
def route_test():
```

```
    return render_template("./Mainpage.html")
```

```
@mainpg.route("/Geocoder.html")
```

```
def route_test2():
```

```
    return render_template("./Geocoder.html")
```

```
@mainpg.route("/results.html")
```

```
def route_test3():
```

```
    return render_template("./results.html")
```

```
@mainpg.route("/comments.html")
```

```
def route_test4():
```

```
    return render_template("./comments.html")
```

```
@mainpg.route("/Mainpage.html")
```

```
def route_test5():
```

```
    return render_template("./Mainpage.html")
```

ПРИЛОЖЕНИЕ 4. Модуль `zaprosh_people.py`

```
from flask import Blueprint, jsonify, request
from coord_repository import CoordsRepository

coordsRepository = CoordsRepository()

coords_bp = Blueprint('coords', __name__)

@coords_bp.route("/coord", methods=["POST"])
def route_coords():
    coordinates=request.get_json()

    nearest_number = coordsRepository.Coord(coordinates)
    print(nearest_number)
    print(coordinates)
    response = jsonify(coordinates,nearest_number)
    response.headers.add('Access-Control-Allow-Origin', '*')
    return response
```

ПРИЛОЖЕНИЕ 5. Модуль coord_repository.py

```
import sqlite3
import math

class CoordsRepository:

    def get_connect(self):
        return sqlite3.connect("./number_of_peoples.sqlite")

    def Coord(self, coordinates):
        try:
            connect = self.get_connect()
            cursor = connect.cursor()

            cursor.execute(
                # """SELECT * FROM `list` WHERE number = ?""", (coords,)
                """SELECT * FROM `list` """,coordinates )

            list = cursor.fetchall()
            print(f"coordinates {coordinates}")

            (ind, nearest_number) = find_nearest_point(list,coordinates)
            print(f"list {list[ind]}")

            # print(f"list {list}")
            return list
        except Exception as error_list:
            print(f"error_list {error_list}")
        finally:
```

```

        connect.close()

    return nearest_number

def find_nearest_point(list, coordinates):
    min_dist = math.inf
    number = -1
    ind = 0
    for elem in list:
        el_coords=elem[2][:-1]
        el_coords=el_coords[1:]
        list_el_coords = el_coords.split(' ')
        el_coords = []
        for a in list_el_coords:
            el_coords.append(float(a))
        cur_dist = haversine_distance(el_coords[1],el_coords[0],
                                      coordinates['coords'][1],coordinates['coords'][0]
                                      )
        if cur_dist<min_dist:
            min_dist = cur_dist
            number = elem[1]
        ind = ind +1
    return ind, number

def haversine_distance(lat1, lon1, lat2, lon2):
    # Константа радиуса Земли в километрах
    R = 6371
    # Преобразование градусов в радианы
    lat1 = math.radians(lat1)
    lon1 = math.radians(lon1)
    lat2 = math.radians(lat2)

```

```

lon2 = math.radians(lon2)
# Вычисление разности широт и долгот
dlat = lat2 - lat1
dlon = lon2 - lon1
# Вычисление гаверсинуса центрального угла
a = math.sin(dlat/2)**2 + math.cos(lat1) * math.cos(lat2) *
math.sin(dlon/2)**2
c = 2 * math.asin(math.sqrt(a))
# Вычисление расстояния по дуге сферы
d = R * c
return d

```


ПРИЛОЖЕНИЕ 6. Модуль menu.js

```
// Получаем ссылки на элементы списка и кнопку
var dropdown = document.querySelector(".dropdown");
var dropdownContent = document.querySelector(".dropdown-content");
var dropbtn = document.querySelector(".dropbtn");

// Добавляем обработчик события для клика на кнопке
dropbtn.addEventListener("click", function() {
    dropdownContent.classList.toggle("show");
});

// Добавляем обработчики событий для клика на элементах списка
var dropdownItems = dropdownContent.querySelectorAll("a");
for (var i = 0; i < dropdownItems.length; i++) {
    dropdownItems[i].addEventListener("click", function() {
        // Запоминаем выбранное значение в localStorage
        localStorage.setItem("selectedValue", this.textContent);

        // Обновляем текст на кнопке
        dropbtn.textContent = this.textContent;

        // Скрываем выпадающее меню
        dropdownContent.classList.remove("show");
    });
}

// При загрузке страницы проверяем, есть ли сохраненное значение в
localStorage
if (localStorage.getItem("selectedValue")) {
    dropbtn.textContent = localStorage.getItem("selectedValue");
    window.addEventListener('beforeunload', function() {
        localStorage.removeItem('dropdown');
    });
}
```