Техническое задание на разработку Веб-приложения «Навигатор учебных заведений Уфы»

1. Введение
   1. Цель

Создание веб-приложения, которое обеспечит удобный поиск и навигацию по учебным заведениям Уфы.

* 1. Актуальность и назначение проекта

Современным жителям города требуется простой и доступный способ подбора учебного заведения. Навигатор позволит экономить время, предоставляя актуальную информацию в одном месте.

* 1. Основные пользователи системы

Основными пользователями системы будут:

* Администратор
* Зарегистрированный пользователь
* Незарегистрированный пользователь

1. Цели и задачи проекта
   1. Основные цели

Основными целями проекта являются:

* Обеспечение пользователей полной и актуальной информацией о учебных заведениях.
* Создание платформы для выбора заведений.
  1. Конкретные задачи
* Разработка базы данных информационной системы.
* Реализация функционала поиска и фильтрации.
* Интеграция с картографическим сервисом для отображения местоположения.
* Создание интерфейса для добавления отзывов и оценок пользователями.

1. Требования к системе
   1. Функциональные требования
2. Разработка функциональности для администратора:

* модерация отзывов;
* управление базой данных информационной системы;
* управление пользователями;
* настройка системы.

1. Разработка функциональности для пользователя:

* фильтрация заведений по типу, местоположению, рейтингу;
* интерактивная карта с метками учебных заведений;
* полнотекстовый поиск по названию и описанию;
* создание отзывов и рейтингов.
* регистрация и аутентификация.
  1. Нефункциональные требования
* Время отклика системы - не более 5 секунд;
* Система должна быть расчитана на более 1000 пользователей с одновременным доступом;
* Интерфейс должен быть интуитивно понятным для обычного пользователя.

1. Требования к пользовательскому интерфейсу
   1. Основные экраны

* Главная страница: список учебных заведений с возможностью фильтрации.
* Страница заведения: информация о заведении, его рейтинге и отзывах.
* Карта: отображение всех заведений в выбранном районе.
* Личный кабинет: информация о пользователе, просмотр оставленных отзывов, просмотр избранных заведений.
  1. Навигация
* система должна включать меню навигации для пользователя с разделами: "Главная", "Карта", "Личный кабинет", "Вход", "Выход";
* система должна включать меню навигации для администратора с разделами: "Учебные заведения", "Отзывы", "Пользователи", "Вход", "Выход";
  1. Юзабилити
* Простой и понятный интерфейс для быстрого доступа к учебным заведениями и карте.

1. Требования к технической реализации
   1. Языки и технологии

* Backend: Node.js;
* Frontend: React.js;
* База данных: MySQL;
* Карты: Google Maps или Яндекс.Карты API(переделать).
  1. Архитектура системы

Хранение данных в реляционной базе данных.

* 1. Интеграции

Внешние уведомления: интеграция с сервисом email-уведомлений, интеграция географических карт.

1. Требования к безопасности
   1. Аутентификация и авторизация

* Авторизация: разграничение прав доступа — администраторы, пользователи.
  1. Шифрование данных
* Хранение паролей: пароли должны храниться в БД в зашифрованном виде.
  1. Политики доступа
* Защита от SQL-инъекций и XSS.

1. Ограничения и допущения
   1. Технические ограничения
   2. Финансовые ограничения

* Бюджет проекта: до 100 ₽, включая тестирование и документацию.
  1. Сроки выполнения
* Полная реализация проекта — 1,5 месяца с начала разработки.

1. Требования к тестированию и приемке
   1. Типы тестирования

* функциональное тестирование: проверка всех реализованных функций;
* тестирование безопасности: проверка на уязвимости.
  1. Критерии приемки
* полное выполнение всех функциональных и нефункциональных требований;
* прохождение не менее 75% тестов по результатам приемочных испытаний.

1. Требования к документации
   1. Пользовательская документация

* руководство пользователя с инструкциями по навигации сайта, записи на занятия, просмотра и редактирования персональных данных.
  1. Техническая документация
* описание архитектуры, структура базы данных.

1. План реализации
   1. Этапы разработки
2. анализ и проектирование — 2 недели;
3. разработка прототипа — 1 неделя;
4. полноценная разработка и тестирование — 5 недели;
5. внедрение и обучение — 2 недели.
   1. Сроки выполнения этапов
6. дата начала: 28 октября 2024 года;
7. дата завершения: 20 декабря 2024 года.

10.3 **Ответственные лица**

Менеджер проекта: Имангулов В.У.

Ведущий разработчик: Имангулов В.У.

Диаграмма UseCase

1. UML диаграммы

UML (с английского аббревиатура расшифровывается как UnifiedModelingLanguage — унифицированный язык моделирования) — это способ наглядно описать архитектуру, проектирование и реализацию комплексных программных систем.

Диаграмма прецедентов (диаграмма вариантов использования) -диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма прецедентов представлена на рисунке 1.

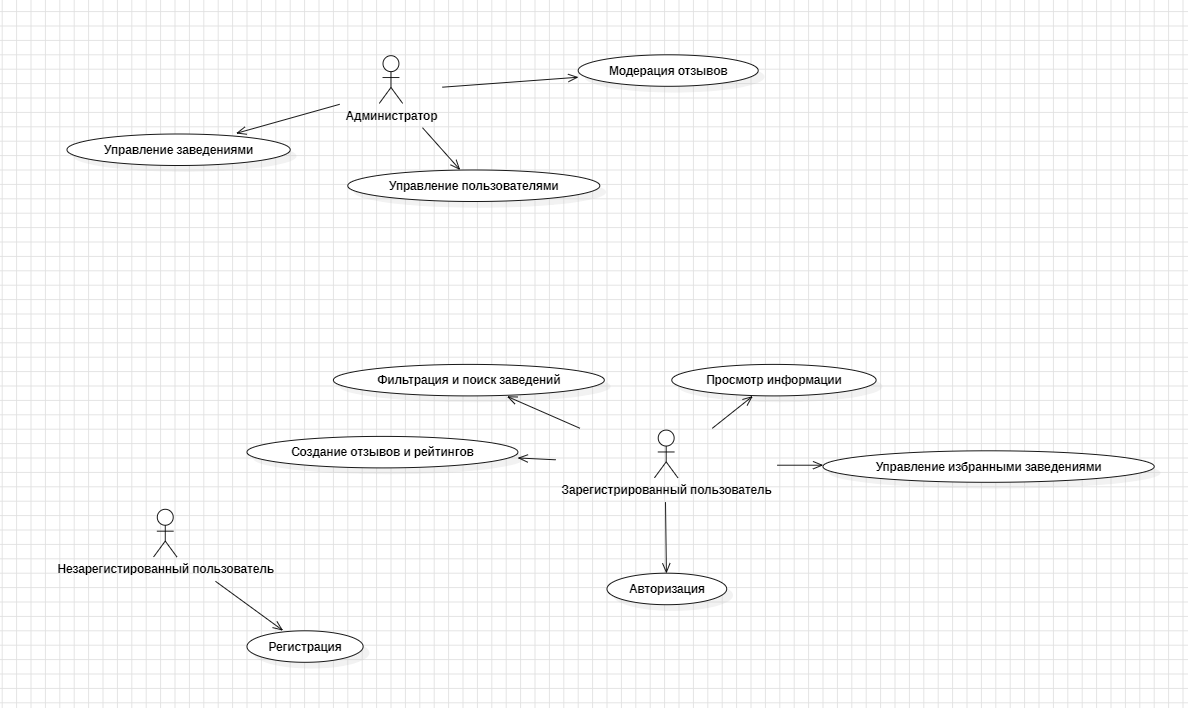
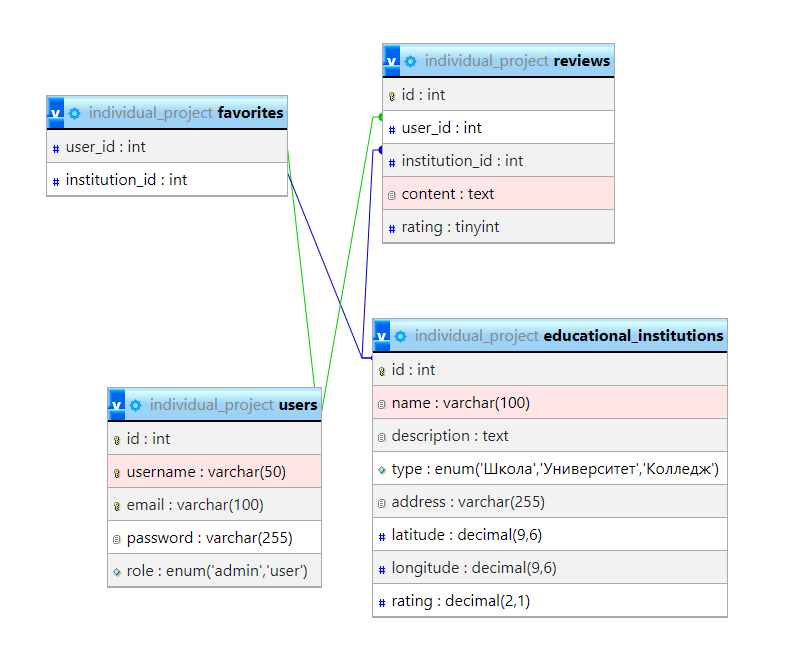


Рисунок 1 — Диаграмма прецедентов

1. Описание структуры базы данных

Схема отношений базы данных представлена на рисунке 2.

  
Рисунок 2 – Схема отношений базы данных

Структура таблицы users представлена в таблице 2.1.

Таблица 2.1 – Users(Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | Id | INTEGER | Первичный ключ, уникальное значение |
| Имя пользователя | username | VARCHAR(50) | Обязательное поле, уникальное значение |
| Почта пользователя | email | VARCHAR(100) | Обязательное поле |
| Пароль | password | VARCHAR(255) | Обязательное поле |
| Роль | role | ENUM(admin, user) | Обязательное поле |

Структура таблицы educational\_institutions представлена в таблице 2.2.

Таблица 2.2 – educational\_institutions (Заведение)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код заведения | Id | INTEGER | Первичный ключ,  уникальное значение |
| Название | name | VARCHAR(100) | Обязательное поле |
| Описание | description | TEXT | Обязательное поле |
| Тип | type | ENUM(Школа, Коллледж, Университет) | Обязательное поле |
| Адрес | address | VARCHAR(255) | Обязательное поле |
| Широта | latitude | DECIMAL(9,6) | Обязательное поле |
| Долгота | longitude | DECIMAL(9,6) | Обязательное поле |
| Рейтинг | rating | DECIMAL(2,1) | Обязательное поле |

Структура таблицы reviews представлена в таблице 2.3.

Таблица 2.3 – reviews (Отзывы)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код записи | id | INTEGER | первичный ключ, уникальное значение |
| Код пользователя | user\_id | INTEGER | Внешний ключ (к users) |
| Код заведения | Institution\_id | INTEGER | Внешний ключ (к educational\_institutions) |
| Контент | content | TEXT | Обязательное поле |
| Рейтинг | rating | TINYINT | Обязательное поле |

Структура таблицы favorites представлена в таблице 2.4.

Таблица 2.4 – favorites (Избранное)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | user\_id | INTEGER | Внешний ключ (к users) |
| Код заведения | Institution\_id | INTEGER | Внешний ключ (к educational\_institutions) |