**Содержание**

[Введение 4](#_Toc194053554)

[1 Постановка задачи и аналитический обзор аналогичных решений 5](#_Toc194053555)

[1.1 Постановка задачи 5](#_Toc194053556)

[1.2 Аналитический обзор аналогов 5](#_Toc194053557)

[1.2.1 Интернет-ресурс «Зеленая карта» 5](#_Toc194053558)

[1.2.2 Интернет-ресурс rsbor.ru 6](#_Toc194053559)

[1.2.3 Интернет-ресурс target99.by 7](#_Toc194053560)

[1.3 Выводы по разделу 8](#_Toc194053561)

[2 Проектирование веб-приложения 9](#_Toc194053562)

[2.1 Диаграмма вариантов использования 9](#_Toc194053563)

[2.2 Логическая схема базы данных MySQL 10](#_Toc194053564)

[2.3 Архитектура веб-приложения 16](#_Toc194053565)

[2.4 Выводы по разделу 17](#_Toc194053566)

[Заключение 18](#_Toc194053567)

[Список используемых источников 19](#_Toc194053568)

[ПРИЛОЖЕНИЕ А 20](#_Toc194053569)

[ПРИЛОЖЕНИЕ Б 21](#_Toc194053570)

[ПРИЛОЖЕНИЕ В 22](#_Toc194053571)

# Введение

ООО «Софтомнител» – белорусская компания, основанная в 2022 году, занимается доработкой и развитием сервиса IP-телефонии и микросервисной платформы «Аксател», являющейся сертифицированной омниканальной средой на серверной платформе. Во время моей практики в этой компании я приобрел ценный опыт в разработке программного обеспечения, работая над реальными проектами и взаимодействуя с профессиональной командой.

Веб-приложение для организации раздельного сбора бытовых отходов представляет собой платформу, предназначенную для повышения осведомленности пользователей и упрощения процесса сортировки и утилизации мусора. Оно обеспечивает доступ к информации о видах отходов, их правильной переработке, ближайших пунктах приема, а также включает дополнительные функции, такие как уведомления, рекомендации по экологичным решениям и отслеживание вклада пользователей в защиту окружающей среды.

Проект посвящен разработке веб-приложения «*WasteWise*» – современной платформы для организации раздельного сбора бытовых отходов. Основной акцент сделан на создание удобного, информативного и функционального сервиса, который соответствует современным требованиям экологической осознанности и цифровых технологий.

Целью данного проекта является создание веб-сервиса, который предоставит пользователям инструменты для правильного разделения отходов и их дальнейшей переработки. Веб-приложение должно поддерживать функционал поиска информации о видах отходов, интерактивной карты пунктов приема, создания и управления личными экологическими целями, а также системы поощрений за ответственное обращение с отходами. В ходе выполнения проекта были поставлены следующие задачи:

* проанализировать аналогичные решения для определения оптимального функционала (Рассмотрено в разделе 1.2);
* спроектировать архитектуру приложения, обеспечивающую поддержку всех заявленных функций (Рассмотрено в разделе 2.3);

Целевая аудитория веб-приложения включает широкий круг пользователей, заинтересованных в экологически ответственном образе жизни. Это могут быть как индивидуальные пользователи, стремящиеся к правильному обращению с отходами, так и организации, внедряющие раздельный сбор мусора. Специальные функции для администраторов делают веб-приложение полезным для муниципальных служб и экологических организаций, контролирующих процессы сбора и переработки отходов.

Серверная часть создана на основе *Node.js 20*, клиентская часть – на *React 18.3.1* с использованием *RTK Query 2.5.0* для управления состоянием. Используется *ORM Sequelize 6.37.5* для работы с *MySQL 3.11.5*. Для работы c ИИ используется *TensorFlow.js 4.22.0*. Разработка ведется с учетом требований к производительности, безопасности и удобству эксплуатации, что подробно раскрыто в последующих главах.

# 1 Постановка задачи и аналитический обзор аналогичных решений

## 1.1 Постановка задачи

Задача заключается в разработке веб-приложения для раздельного сбора бытовых отходов, которое предоставит пользователям удобный и функциональный интерфейс для взаимодействия с системой утилизации мусора.

Для пользователя приложение должно реализовывать функции поиска и отображения ближайших пунктов сбора отходов, получения информации о сортировке мусора. Администратор должен иметь возможность управлять базой данных мест сбора, добавлять и удалять пункты, следить за системой выдачей промокодов.

## 1.**2 Аналитический обзор аналогов**

Были проанализированы цели и задачи, поставленные в данном проекте, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

### 1.2.1 Интернет-ресурс «Зеленая карта»

Для анализа был изучен один из популярных экологических ресурсов, белорусский сайт «Зеленая карта» — *greenmap.by* [1].

На рисунке 1.1 показана страница данного сайта.

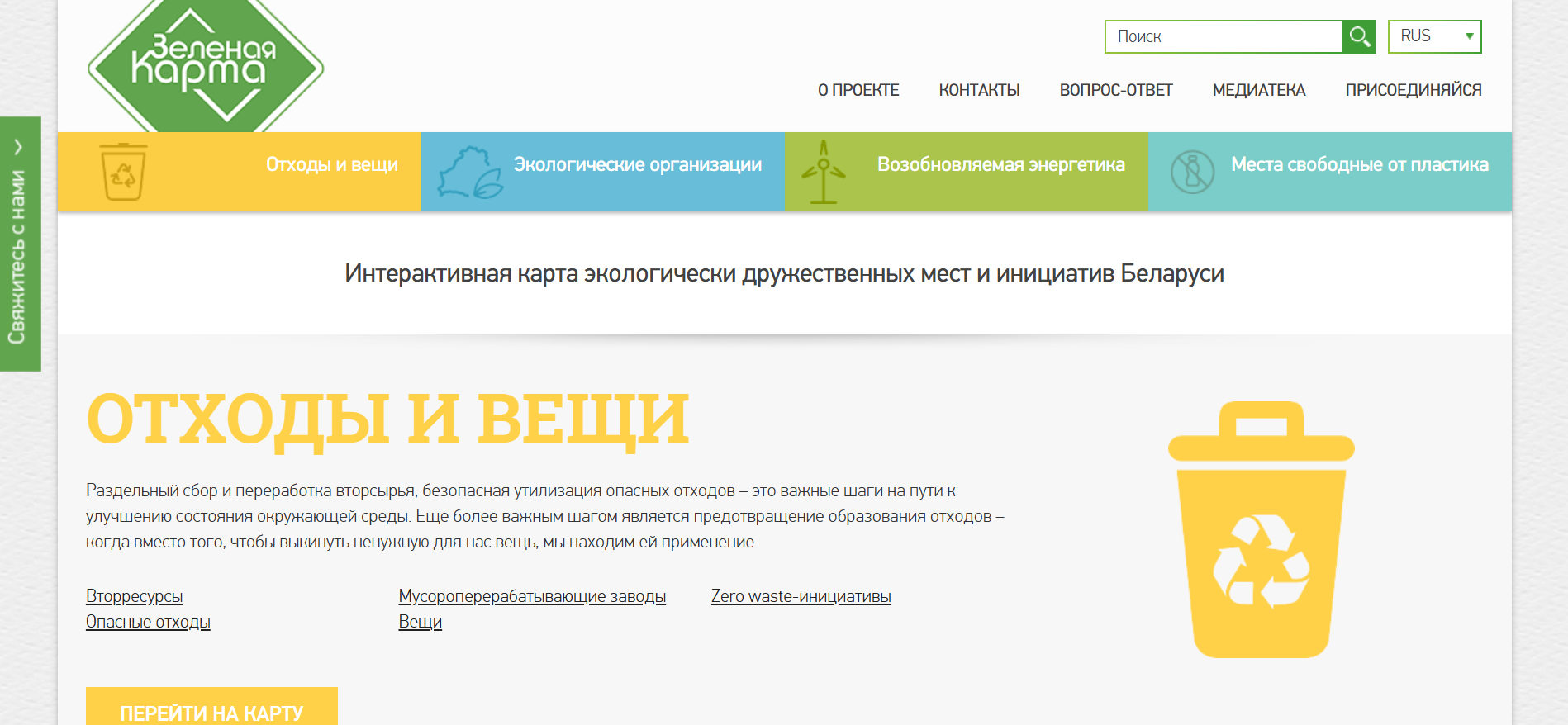


Рисунок 1.1 – Сайт «Зеленая карта»

Кроме информации о раздельном сборе отходов, на сайте также можно найти информацию о различных экологических организациях, включая государственные организации, научные лаборатории и учебные заведения. Кроме того, на сайте есть отдельный пункт меню, который посвящен возобновляемой энергетике, и на котором можно найти информацию о солнечных панелях, солнечных водонагревателях, ветрогенераторах, биогазовых установках и гидроэлектростанциях. На сайте также есть пункт меню, где представлены места, свободные от пластика, такие как школы, церковные приходы, кафе и офисы.

В данном проекте не было найдено никаких недостатков, ресурс полностью соответствует своему названию *«Зеленая карта»*, и все возможные места, связанные с экологическими объектами, расположены на карте в понятном и доступном виде. Этот ресурс является отличным источником информации для тех, кто заботится об окружающей среде и хочет внести свой вклад в экологическую инициативу. Благодаря *«Зеленой карте»* люди могут быть в курсе всех экологических объектов и ресурсов в своем городе, что помогает им принимать осознанные решения и принимать активное участие в раздельном сборе отходов, защите природы и пропаганде экологического образа жизни.

### 1.2.2 Интернет-ресурс rsbor.ru

Следующим результатом поиска аналогов стал сайт *rsbor.ru* [2], главная страница которого представлена на рисунке 1.2.

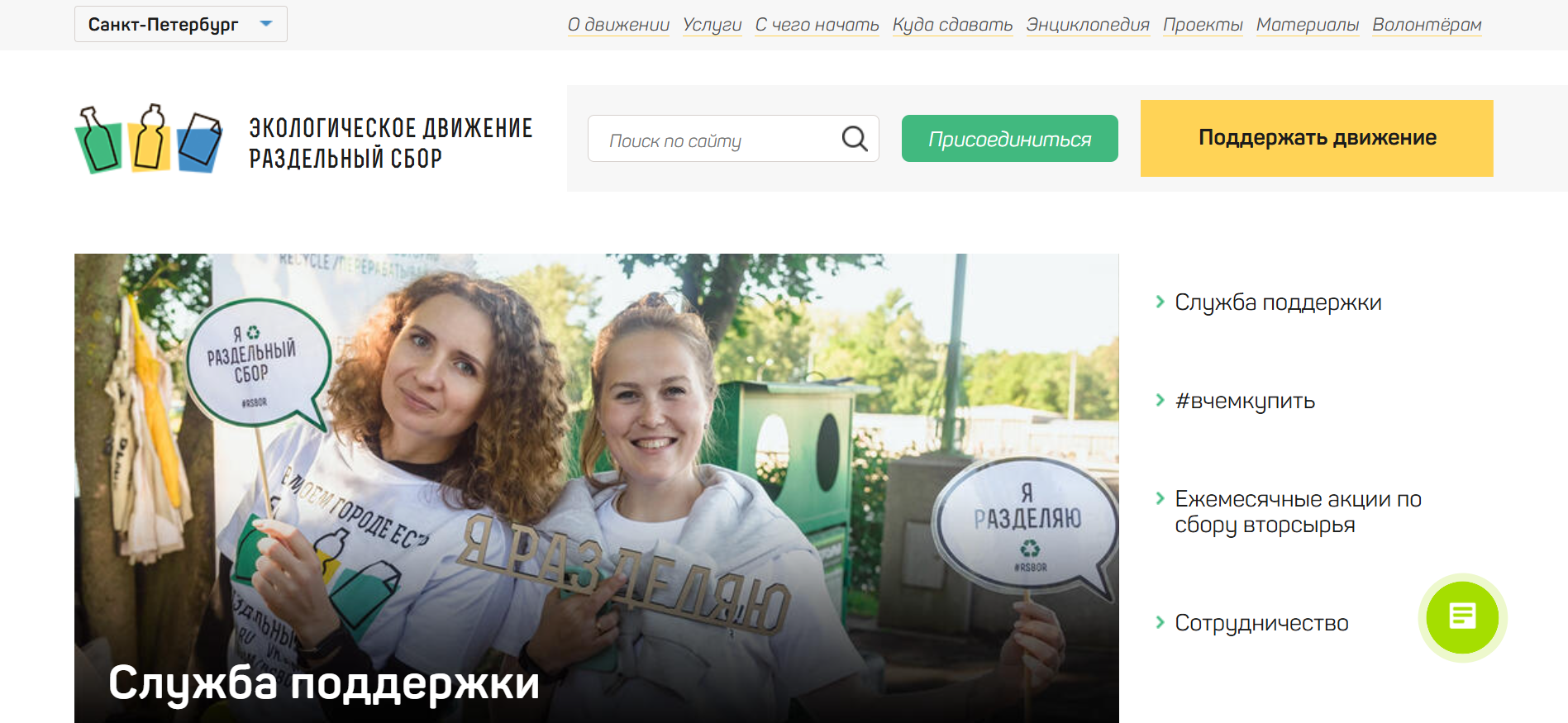


Рисунок 1.2 – Сайт *rsbor.ru*

Сайт *rsbor.ru* представляет собой информационный ресурс, основной упор которого сделан на новостную составляющую. На главной странице пользователи могут ознакомиться с последними новостями, связанными со сбором, разделением и переработкой отходов и вторсырья в Санкт-Петербурге или выбранном городе из выпадающего списка. Кроме того, на других страницах представлен зарубежный опыт в области переработки отходов и ресурсосбережения, позволяя пользователям расширить свои знания в этой сфере.

Одним из важных разделов сайта является раздел, посвященный проектам, проводимым Экологическим Движением «Раздельный Сбор». Здесь пользователи могут ознакомиться с текущими и предстоящими проектами, которые организует движение. Также на сайте предусмотрена возможность сделать пожертвования для поддержки и развития этого движения, позволяя желающим внести свой вклад.

Одной из удобных функций сайта является просмотр пунктов приема, где отходы могут быть сданы по видам вторсырья. Эта информация предоставляет удобный способ для пользователей найти ближайшие пункты приема и правильно сортировать свои отходы.

### 1.2.3 Интернет-ресурс target99.by

После первых двух приложений обратилось внимание к белорусскому ресурсу государственного учреждения «Оператор вторичных материальных ресурсов» [3], уполномоченного министерством жилищно-коммунального хозяйства Республики Беларусь, для большего понимания какое развитие имеет ситуация с вторсырьем в нашей стране. Данный веб-ресурс представлен на рисунке 1.3.

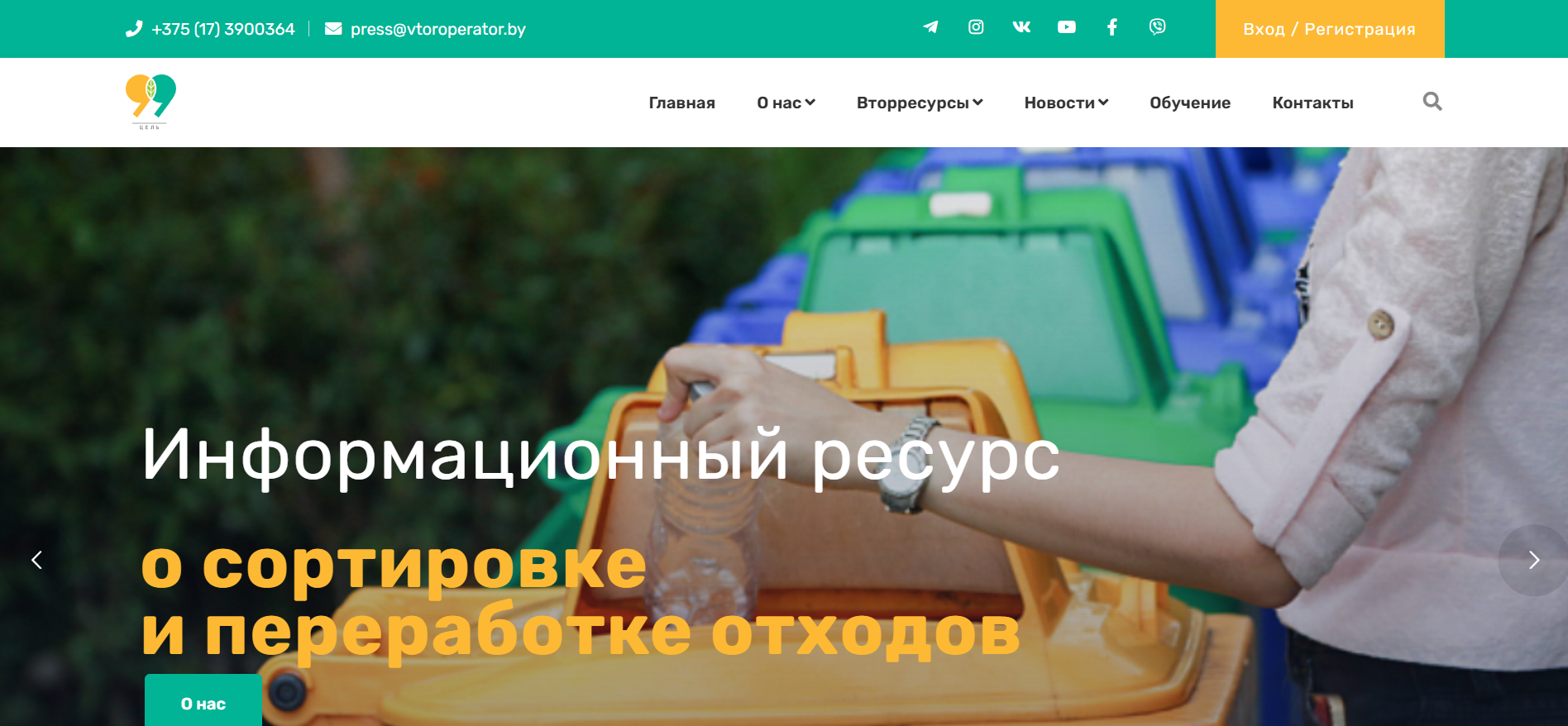


Рисунок 1.3 – Сайт target99.by

Основная информация, представленная на главной странице сайта, сосредоточена на правилах сортировки вторичных ресурсов. Чтобы облегчить поиск нужной информации, правила разделены по категориям, что позволяет пользователям быстро найти необходимую информацию в соответствии с их потребностями и видами отходов.

Одним из важных аспектов развития раздельного сбора мусора является подробная карта пунктов приема вторичного сырья. В Беларуси сегодня насчитывается более 1700 пунктов приема, и их число постоянно увеличивается. Эта карта предоставляет пользователю удобный инструмент для поиска ближайших пунктов приема и утилизации отходов, позволяя эффективно управлять своими ресурсами.

В целом, после рассмотрения всех указанных ресурсов на данном сайте не было выявлено никаких недостатков. Это говорит о том, что ресурс предоставляет полезную информацию и инструменты, необходимые для активного участия в раздельном сборе и переработке отходов, а также для продвижения экологичного образа жизни.

## 1.3 Выводы по разделу

Анализ аналогичных веб-ресурсов, таких как «Зеленая карта», *rsbor.ru* и сайт государственного учреждения «Оператор вторичных материальных ресурсов», позволил выделить ключевые принципы, которые следует учитывать при разработке веб-приложения для управления отходами и экологии.

Во-первых, важна доступность информации и удобство навигации. «Зеленая карта» демонстрирует, как интерактивная карта с геолокацией точек сбора отходов повышает вовлеченность пользователей. Это позволяет быстро находить экологические объекты и точки раздельного сбора.

Во-вторых, новостная и образовательная составляющая играет значительную роль. Опыт *rsbor.ru* показывает, что информирование пользователей о последних тенденциях, зарубежных практиках и локальных инициативах в сфере переработки отходов способствует осознанному подходу к экологии.

В-третьих, важна интеграция с государственными программами и официальными инициативами. Белорусский ресурс государственного учреждения «Оператор вторичных материальных ресурсов» демонстрирует, что предоставление актуальных правил сортировки отходов и возможности поиска пунктов приема делает сервис максимально полезным.

Таким образом, разрабатываемое веб-приложение должно включать удобный интерфейс для поиска пунктов приема отходов, актуальную новостную ленту о переработке отходов, а также интеграцию с государственными и частными экологическими инициативами. Это позволит создать полезный и востребованный ресурс для пользователей, заинтересованных в раздельном сборе и экологическом образе жизни.

# 2 Проектирование веб-приложения

## 2.1 Диаграмма вариантов использования

Для описания основных сценариев взаимодействия пользователей с веб-приложением была разработана диаграмма вариантов использования. Диаграмма демонстрирует ключевые роли в системе и действия, доступные для каждой из них.

На диаграмме отражены все действующие в приложении роли и функции, которые доступны этим ролям. Описание ролей представлено в таблице 2.1.

Таблица 2.1 – Описание ролей

|  |  |
| --- | --- |
| Роль | Описание |
| Гость | Может регистрироваться в системе и авторизоваться. Имеет доступ к просмотру статей. Не имеет доступа к основному функционалу. |
| Пользователь | Может просматривать пункты приема вторсырья, обменивать накопленные баллы на скидки в различных сервисах, создавать, изменять и удалять свои статьи, оценивать статьи, а также получать информацию о том, сколько новой продукции будет произведено из сданных отходов. |

Описание основных функций представлено в таблице 2.2.

Таблица 2.2 – Описание функций

|  |  |  |
| --- | --- | --- |
| Функция | Роль | Описание |
| Просмотр статей | Пользователь | Обеспечивает возможность читать информационные материалы. |
| Просмотр пунктов приема | Пользователь | Предоставляет информацию о пунктах приема вторсырья. |
| Просмотр скидок | Пользователь | Отображает список доступных скидок за сдачу вторсырья. |
| Обмен баллов на скидки в сервисах | Пользователь | Позволяет пользователям получить скидку за сдачу вторсырья. |
| Добавление статьи | Пользователь | Позволяет пользователям добавлять новые статьи для публикации. |
| Изменение статьи | Пользователь | Позволяет пользователям редактировать существующие статьи. |
| Удаление статьи | Пользователь | Позволяет пользователям удалять статьи. |
| Добавление комментария | Пользователь | Позволяет комментировать статьи для обсуждения материалов. |
| Удаление комментария | Пользователь | Удаляет комментарии из статьи. |
| Добавление отметки “Нравится” | Пользователь | Позволяет отметить понравившиеся статьи. |

Окончание таблицы 2.2

|  |  |  |
| --- | --- | --- |
| Отметка сдачи вторсырья | Пользователь | Фиксирует факт сдачи вторсырья. |
| Добавление пункта приема | Администратор | Позволяет администратору добавлять новые пункты приема вторсырья. |
| Изменение времени работы | Администратор | Позволяет администратору изменять время работы существующих пунктов приема. |
| Удаление пункта приема | Администратор | Позволяет администратору удалять пункты приема вторсырья. |
| Добавление вторсырья | Администратор | Вносит новые виды вторсырья в классификацию. |
| Изменение вторсырья | Администратор | Корректирует описание существующих видов вторсырья. |
| Удаление вторсырья | Администратор | Удаляет ненужные категории или виды из базы данных. |

В приложении А представлена полная диаграмма вариантов использования.

## 2.2 Логическая схема базы данных MySQL

Диаграмма базы данных таблиц (*Database Table Diagram*) – это визуальное представление структуры базы данных и отношений между таблицами, которые хранятся в этой базе данных.

Для веб-приложения была спроектирована база данных, включающая **12 таблиц**, обеспечивающих хранение информации о пользователях, статьях, пунктах приема и других объектах системы. Ее логическая схема показана в приложении Б.

Таблица *users* содержит данные о пользователях приложения, включая их уникальные идентификаторы, учетные записи и другую важную информацию.

Таблица *article* хранит опубликованные статьи, предоставляя пользователям доступ к актуальному контенту.

Таблица *ratings* содержит комментарии к статьям, позволяя пользователям оставлять обратную связь и участвовать в обсуждениях.

Таблица *likes* фиксирует информацию о лайках, которые пользователи поставили статьям. При повторном вызове функции лайка запись удаляется, что позволяет реализовать механику отмены лайка.

Таблица *points* содержит данные о пунктах приема вторсырья, помогая пользователям находить ближайшие места для сдачи отходов.

Таблица *marks* хранит сведения о видах вторсырья, что позволяет классифицировать принимаемые материалы.

Таблица *points\_marks* реализует связь «многие-ко-многим» между пунктами приема вторсырья (*points*) и видами вторсырья (*marks*), обеспечивая гибкость в управлении информацией.

Таблица *check\_weight* содержит коды для проверки веса сданного вторсырья и его типа, что позволяет автоматизировать процесс учета.

Таблица *s\_keys* хранит секретные ключи для пунктов приема вторсырья, обеспечивая защиту данных и контроль доступа.

Таблица *discounts* содержит информацию о доступных скидках, предлагаемых пользователям.

Таблица *promo\_codes* хранит промокоды, которые пользователи могут активировать для получения бонусов и специальных предложений.

Таблица *receptions* является статистической таблицей, фиксирующей данные о том, какой пользователь сдал какое вторсырье, что позволяет анализировать динамику и эффективность переработки.

В таблице 2.3 показано описание полей таблицы «*users*».

Таблица 2.3 – Описание структуры таблицы «*users*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор пользователя (первичный ключ) |
| *username* | *VARCHAR(20)* | *NOT NULL* | Имя пользователя |
| *email* | *VARCHAR(100)* | *NOT NULL, UNIQUE* | Почта |
| *password\_hash* | *VARCHAR(200)* | *NULLABLE, UNIQUE* | Захешированный пароль |
| *points* | *INT* | *DEFAULT 0, NULL* | Количество накопленных баллов |
| *role* | *VARCHAR(5)* | *DEFAULT ‘USER’, NOT NULL, CHECK (role IN ('admin', 'user')* | Роль пользователя |
| *is\_activated* | *TINYINT(1)* | *DEFAULT 0, NULL* | Активирована почта или нет |
| *activation\_link* | *VARCHAR(150)* | *NULL* | Ссылка активации, которая была отправлена на почту |

В таблице 2.4 показано описание полей таблицы «*articles*».

Таблица 2.4 – Описание структуры таблицы «*articles*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | INT | *PRIMARY KEY* | Идентификатор статьи (первичный ключ) |
| *title* | VARCHAR(100) | *NOT NULL, UNIQUE* | Название статьи |
| *text* | TEXT | *NOT NULL* | Текст статьи |
| *date\_of\_pub* | DATE | *NOT NULL* | Дата публикации |
| *image\_url* | VARCHAR(150) | *NOT NULL* | Ссылка на изображение |
| *author* | INT | *NOT NULL, FOREIGN KEY* | Автор публикации (внешний ключ) |

Окончание таблицы 2.4

|  |  |  |  |
| --- | --- | --- | --- |
| *likes* | INT | *DEFAULT 0, NULL* | Количество лайков у данной статьи |

В таблице 2.5 показано описание полей таблиц «*ratings*».

Таблица 2.5 – Описание структуры таблицы «*ratings*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | INT | *PRIMARY KEY* | Идентификатор комментария (первичный ключ) |
| *article\_id* | INT | *NOT NULL, FOREIGN KEY* | Идентификатор статьи (внешний ключ) |
| *commentator* | INT | *NOT NULL, FOREIGN KEY* | Идентификатор пользователя (внешний ключ) |
| *comment* | VARCHAR(500) | *NOT NULL* | Текст комментария |
| *date\_of\_add* | DATETIME | *DEFAULT CURRENT\_TIMESTAMP* | Дата добавления комментария |

В таблице 2.6 показано описание полей таблицы «*likes*».

Таблица 2.6 – Описание структуры таблицы «*likes*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор лайка (первичный ключ) |
| *user\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор пользователя (внешний ключ) |
| *article\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор статьи (внешний ключ) |
| *date\_of\_add* | *DATETIME* | *DEFAULT CURRENT\_TIMESTAMP* | Время добавления отметки |

В таблице 2.7 представлена структура таблицы «*s\_keys*».

Таблица 2.7 – Описание структуры таблицы «*s\_keys*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор секретного ключа (первичный ключ) |
| *secret\_key* | *VARCHAR(100)* | *NOT NULL* | Секретный ключ в хешированном виде |
| *is\_used* | *INT* | *DEFAULT 0, NOT NULL, CHECK (is\_used IN (1, 0))* | Содержит значения 1 – если ключ использован и 0 – не использован |

В таблице 2.8 представлена структура таблицы «*points*».

Таблицы 2.8 – Описание структуры таблицы «*points*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор пункта приема (первичный ключ) |
| *address* | *VARCHAR(100)* | *NOT NULL, UNIQUE* | Адрес пункта приема |
| *time\_of\_work* | *VARCHAR(100)* | *NOT NULL* | Время работы пункта приема |
| *key\_id* | *INT* | *NOT NULL, UNIQUE, FOREIGN KEY* | Идентификатор на секретный ключ (внешний ключ) |
| *admin\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор администратора, который добавил этот пункт (внешний ключ) |
| *link\_to\_map* | *TEXT* | *NOT NULL* | Ссылка на Яндекс карту |
| *point\_name* | *VARCHAR(100)* | *NOT NULL, UNIQUE* | Имя пункта приема |

В таблице 2.9 представлена структура таблицы «*marks*».

Таблица 2.9 – Описание структуры таблицы «*marks*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор вида вторсырья (первичный ключ) |
| *rubbish* | *VARCHAR(50)* | *NOT NULL* | Вид вторсырья |
| *points\_per\_kg* | *INT* | *NOT NULL* | Начисляемые баллы за один килограмм |
| *new\_from\_kg* | *FLOAT* | *NOT NULL* | Сколько новой продукции будет произведено из 1 кг сданного вторсырья |
| *image\_link* | *VARCHAR(255)* | *NULL* | Ссылка на картинку |

В таблице 2.10 представлено описание структура таблицы «*points\_marks*».

Таблица 2.10 – Описание структуры таблицы «*points\_marks*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор (первичный ключ) |
| *points\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор пункта приема (внешний ключ) |
| *marks\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор вторсырья (внешний ключ) |

В таблице 2.11 представлено описание структуры таблицы «*check\_weigth*».

Таблица 2.11 – Описание структуры таблицы «*check\_weigth*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор (первичный ключ) |
| *rubbish\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор на вид вторсырья (внешний ключ) |
| *weight* | *INT* | *NOT NULL* | Вес |
| *key\_of\_weight* | *VARCHAR(100)* | *NOT NULL, UNIQUE* | Ключ для проверки в захешированном виде |
| *is\_used* | *INT* | *DEFAULT 0, NOT NULL, CHECK (is\_used IN (0, 1))* | Содержит значения 1 – если ключ использован и 0 – не использован |
| *oridinal\_key* | *VARCHAR(100)* | *NOT NULL* | Оригинальный ключ |

В таблице 2.12 представлено описание структуры таблицы «*receptions*».

Таблица 2.12 – Описание структуры таблицы «*receptions*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор приема (первичный ключ) |
| *user\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор пользователя (внешний ключ) |
| *weight* | *FLOAT* | *NOT NULL* | Вес сколько было сдано вторсырья |
| *accrued* | *INT* | *NULL* | Начисленные баллы |
| *new\_kg* | *INT* | *NULL* | Вес новой продукции |
| *type\_waste* | *INT* | *NOT NULL, FOREIGN KEY* | Вид вторсырья |
| *station\_key* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор секретного ключа (внешний ключ) |
| *weight\_key* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор ключа для проверки веса (внешний ключ) |

В таблице 2.13 представлено описание структуры таблицы «*discounts*».

Таблица 2.13 – Описание структуры таблицы «*discounts*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор скидки (первичный ключ) |
| *discounts* | *VARCHAR(50)* | *NOT NULL, UNIQUE* | Название скидки |
| *promo\_code* | *VARCHAR(50)* | *NOT NULL, UNIQUE* | Промокод для применения скидки |

Окончание таблицы 2.13

|  |  |  |  |
| --- | --- | --- | --- |
| *count\_for\_dnt* | *INT* | *NOT NULL* | Стоимость скидки в баллах |

В таблице 2.14 представлено описание структуры таблицы «*promo\_codes*».

Таблица 2.14 – Описание структуры таблицы «*promo\_codes*»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Ограничения | Описание |
| *id* | *INT* | *PRIMARY KEY* | Идентификатор промокода (первичный ключ) |
| *promo\_code* | *VARCHAR(50)* | *NOT NULL* | Текст промокода |
| *user\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор пользователя, который открыл скидку (внешний ключ) |
| *discount\_id* | *INT* | *NOT NULL, FOREIGN KEY* | Идентификатор скидки (внешний ключ) |
| *date\_of\_add* | *DATE* | *NOT NULL, DEFAULT ‘2025-03-27’* | Дата, когда был открыт промокод |

Один пользователь может быть автором множества статей, что определяется связью между таблицами *users* и *articles*. В свою очередь, одна статья может иметь множество комментариев, что устанавливает связь между *articles* и *ratings*.

Один пользователь может оставить множество лайков, что фиксируется связью между таблицами *users* и *likes*. Аналогично, одна статья может иметь множество лайков, что связывает *articles* и *likes*.

Один пользователь может быть администратором множества пунктов приема, что выражается в связи между *users* и *points*. Каждый пункт приема может принимать множество видов вторсырья, что фиксируется связью между *points* и *points\_marks*.

Один пользователь может сдать множество сдач во множестве пунктов приема, что устанавливает связь между *users* и *receptions*, а также между *points* и *receptions*.

Один вид вторсырья может быть связан с множеством пунктов приема, что устанавливает связь между *marks* и *points\_marks*.

Один вид вторсырья может иметь множество записей о проверке веса, что фиксируется связью между *marks* и *check\_weight*.

Один вид вторсырья может быть сдан множество раз, что выражается в связи между *marks* и *receptions*.

Одна запись о проверке веса может быть связана с множеством сдач, что фиксируется связью между *check\_weight* и *receptions*.

Одна скидка может быть связана с множеством промокодов, что устанавливает связь между *discounts* и *promo\_codes*.

## 2.3 Архитектура веб-приложения

Архитектура приложения представляет собой распределённую систему, развернутую с использованием современных технологий, обеспечивающих производительность, масштабируемость и удобство управления. Основные компоненты системы включают серверную часть, клиентскую часть, развёрнутые в контейнерах *Docker* 27.3.1 [4]. Диаграмма развертывания представлена в приложении В. Описание компонент веб-приложения содержится в таблице 2.15.

Таблица 2.15 – Описание компонент веб-приложения

|  |  |
| --- | --- |
| Компонента | Описание компоненты |
| *Frontend Server* | Предоставляет доступ к статическим ресурсам клиентской части веб-приложения |
| *Database Server* | Используется для хранения и предоставления доступа к данным, которые необходимы для работы веб-приложения. |
| *Backend Server* | Обрабатывает запросы пользователя, запрашивает данные из базы данных с помощью *ORM Sequelize* [5]. |
| *Browser* | Отображает клиентскую часть веб-приложения, отправляет запросы пользователя, отображает ответы сервера. |
| *Cloudinary* [6] | Облачный сервис, предназначенный для хранения изображений. |
| *SMTP Yandex* [7] | Используется для отправки электронных писем, например, для уведомлений пользователей или восстановления пароля. |
| *Yandex Maps API* [8] | Предоставляет функциональность для работы с картами, геолокацией и маршрутами. |
| *Goodle API* [9] | Включает сервисы, такие как *Google Maps*, *OAuth*, *Drive*, и другие, используемые в веб-приложении. |

Описание протоколов, используемых при работе веб-приложений, представлено в таблице 2.16.

Таблица 2.16 – Описание используемых протоколов

|  |  |
| --- | --- |
| Протокол | Назначение |
| *HTTP* [10] | Обмен данными между *Client Browser* и *Frontend Server*, *Frontend Server* и *Backend Server.* |
| *HTTPS* [11] | Обмен данными между *Backend Server* и *Cloudinary*. Обеспечивает безопасную передачу данных путём использования криптографического протокола *TLS* |
| *SMTP* [12] | Протокол для отправки электронных писем, используемый для передачи сообщений между почтовыми серверами и клиентами. |
| *TCP* [13] | Обмен данными между *Database Server* и *Web API Server*. |

Таким образом были рассмотрены все ключевые элементы архитектуры веб-приложения.

## 2.4 Выводы по разделу

В процессе проектирования веб-приложения «*WasteWise*» определены основные роли пользователей, функции, архитектура, а также схемы базы данных и связи между сущностями. Приложение разработано с учётом современных технологий и стандартов, что обеспечивает его функциональность, масштабируемость и удобство использования.

Основные выводы представлены в следующем списке:

1. Поддержка трех ролей с четко разграниченными правами доступа и функциональными возможностями: гость, клиент, администратор. Гость может зарегистрироваться и авторизоваться, Пользователь имеет доступ к основным функциям приложения, а Администратор управляет системой.
2. Количество функций: 29. Функционал приложения охватывает регистрацию и авторизацию, просмотр пунктов приема, вторсырья, создание, редактирование и удаление статей.
3. Количество таблиц в базе данных: 12. Таблицы охватывают основные сущности приложения, такие как пользователи, статьи, вторсырье, пункты приема, скидки и промокоды.
4. Веб-приложение имеет монолитную архитектуру с применением *Nginx* в качестве веб-сервера, *MySQL* для хранения данных, *Node.js* для серверной части и *Docker Compose* для запуска многоконтейнерных *Docker*-приложений.

Таким образом, веб-приложение «*WasteWise*» спроектировано с учётом всех ключевых аспектов, включая современные технологии и практики проектирования, что делает его эффективным инструментом для решения задач пользователей.

# Заключение

В рамках прохождения практики в компании **ООО «Софтомнител»,** занимающейся развитием сервиса IP-телефонии и микросервисной платформы «Аксател», была выполнена работа по разработке веб-приложения для обмена сообщениями. Компания предоставила необходимые ресурсы, поддержку опытных специалистов и доступ к современным технологиям, что позволило успешно реализовать поставленные задачи.

В ходе выполнения проекта было реализовано веб-приложение, которое соответствует всем заявленным целям и требованиям.

* 1. В системе предусмотрено три ключевые роли – пользователь, администратор и гость.
  2. Реализовано 29 функций, включая регистрацию, авторизацию, просмотр статей, просмотр пунктов приема вторсырья, обмен баллов на скидки, создание, изменение и удаление статей, оценку статей, а также управление системой со стороны администратора.
  3. В базе данных для хранения было создано 12 таблиц: пользователи, статьи, комментарии, лайки, пункты приема, виды вторсырья, секретные ключи пункты приема, скидки, промокоды, таблица для статистики и таблица для связи «многие-ко-многим» между таблицами с пунктами приема и видами вторсырья.
  4. Архитектура приложения построена на принципах клиент-серверного взаимодействия. Серверная часть разработана с использованием платформа *Node.js*, а клиентская – на *React.js*.
  5. В проекте написано 9500 строк кода, включая серверную и клиентскую части.
  6. Разработаны и проведены 29 успешных функциональных тестов, а покрытие тестами составило 100%

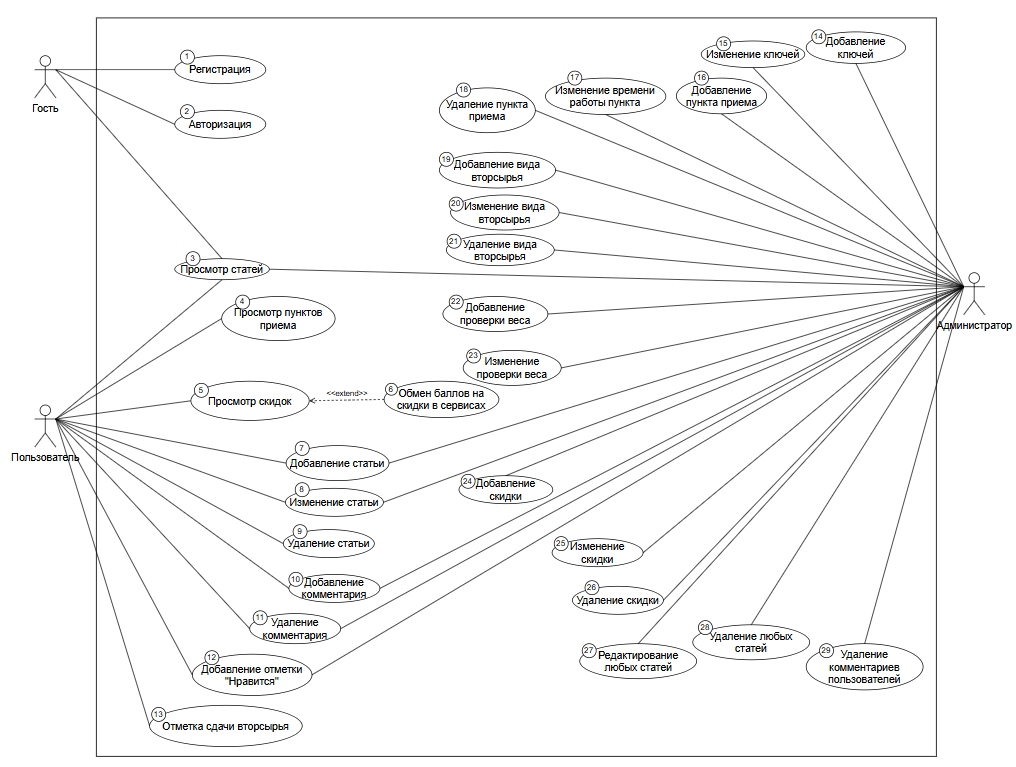
В соответствии с полученным результатом работы можно сделать вывод, что цель достигнута, а требования технического задания полностью выполнены.

# Список используемых источников

1. Сайт по раздельному сбору вторсырья Зеленая карта [Электронный ресурс]. – Режим доступа: http://greenmap.by/ – Дата доступа: 05.10.2024.
2. Сайт по раздельному сбору вторсырья Rsbor [Электронный ресурс]. – Режим доступа: https://rsbor.ru/ – Дата доступа: 05.10.2024.
3. Сайт по раздельному сбору вторсырья Target99 [Электронный ресурс]. – Режим доступа: https://target99.by/– Дата доступа: 09.10.2024.
4. Docker Official Documentation. [Электронный ресурс] – Режим доступа: https://docs.docker.com. – Дата обращения: 04.12.2024.
5. Sequelize Documentation. [Электронный ресурс] – Режим доступа: https://sequelize.org/docs/v6/. – Дата обращения: 02.11.2024.
6. Cloudinary [Электронный ресурс]. – Режим доступа: https://cloudinary.com/ – Дата доступа: 26.10.2024.
7. SMTP Yandex [Электронный ресурс]. – Режим доступа: https://yandex.ru/support/yandex-360/customers/mail/ru/mail-clients/others– Дата доступа: 03.11.2024.
8. Yandex Maps API — Документация [Электронный ресурс]. – Режим доступа: https://yandex.ru/dev/maps/ – Дата обращения: 03.11.2024.
9. Google OAuth API — Документация [Электронный ресурс]. – Режим доступа: https://developers.google.com/identity/protocols/oauth2 – Дата обращения: 03.11.2024.
10. RFC 2616 HTTP/1.1 [Электронный ресурс]. – Режим доступа: https://www.rfc-editor.org/rfc/rfc2616 – Дата доступа: 01.10.2024.
11. HTTPS — Википедия [Электронный ресурс] / Режим доступа: https://ru.wikipedia.org/wiki/HTTPS – Дата доступа: 21.10.2024.
12. SMTP [Электронный ресурс]. – Режим доступа: https://www.ietf.org/rfc/rfc5321.txt – Дата доступа: 03.11.2024.
13. RFC 2818 HTTP Over TLS [Электронный ресурс]. – Режим доступа: https://www.rfc-editor.org/rfc/rfc2818 – Дата доступа: 26.10.2024.

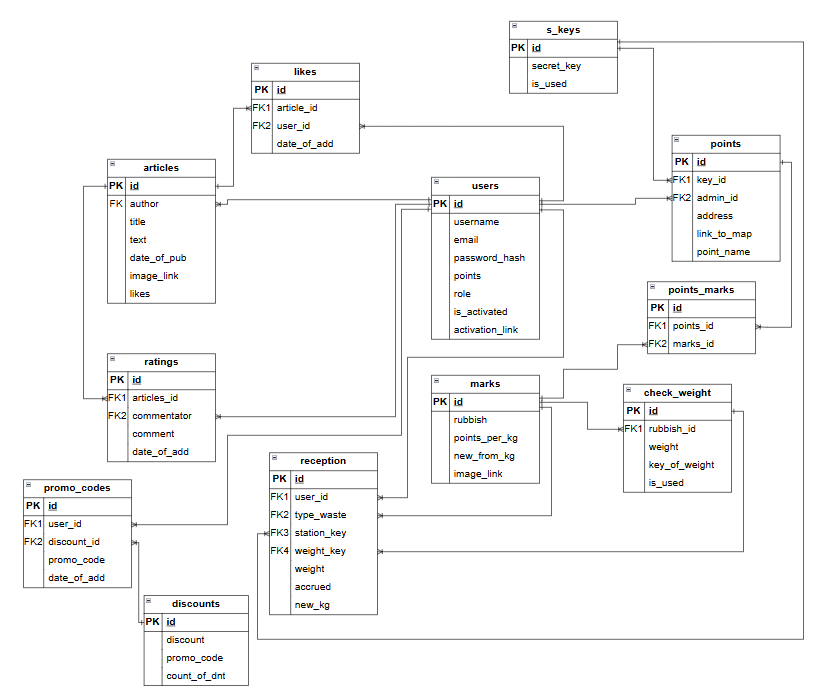
# ПРИЛОЖЕНИЕ А

**Диаграмма вариантов использования**



# ПРИЛОЖЕНИЕ Б

**Логическая схема базы данных**



# ПРИЛОЖЕНИЕ В

**Диаграмма развертывания**

