Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа по предмету «Проектирование информационных систем» на тему «Объектно-ориентированное модулирование. UML–диаграммы поведения»

Студент: Трубач Д. С.

ФИТ 4 курс 5 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# 1 Постановка задач

Темой данной лабораторной работы является построение UML-диаграммы вариантов использования, необходимой для графического представления взаимосвязи ролей конечных пользователей и вариантов взаимодействий с системой, представленной в первой лабораторной работе – веб-приложения для раздельного сбора вторсырья «EcoSort».

Web-приложение должно быть выполнено с использованием асинхронного программирования, взаимодействовать с базой данных, реализовано под разными платформами. Web-приложение должно представлять собой web-приложение с асинхронным UI с использованием фреймворка React. Отображение, бизнес-логика и хранилище данных должны быть максимально независимы друг от друга для возможности расширения. Диаграмму вариантов использования разработать на основе UML, также необходимо разработать логическую схему базы данных и структурную схему приложения. Язык разработки проекта JavaScript, платформа «NodeJS». Развертывание конечного приложения для последующего использования должно осуществляться с использованием Docker.

Функционально web-приложение должно:

* поддерживать роли гостя, пользователя, администратора;
* обеспечить возможность регистрации, авторизации;
* обеспечивать возможностью просмотра пунктов вторсырья;
* обеспечивать возможностью обмена своих накопленных баллов на скидки в различных сервисах;
* обеспечивать возможностями создания, изменения и удаления своих статей;
* обеспечивать возможностью оценки статей;
* обеспечивать возможностью получения информации сколько новой продукции будет сделано из сданных отходов.
* обеспечивать возможностью добавления новых пунктов приема вторсырья, изменение и удаление существующих;
* обеспечивать возможностью изменения и удаления видов вторсырья;
* обеспечивать возможностью добавления скидок, которые пользователи могут использовать за накопленные баллы;
* обеспечивать возможностью изменения и удаление статей о раздельном сборе отходов.

Основные задачи приложения:

* разработать удобный интерфейс, который позволит пользователям легко ориентироваться и находить информацию о сортировке отходов;
* предоставить функционал для ознакомления с пошаговыми инструкциями по раздельному сбору мусора;

создать систему накопления баллов за сортировку, с возможностью их обмена на товары, произведенные из переработанных материалов.

# 2 Описание программных средств

Для построения моделей было использовано программное средство Draw.io (также известное как diagrams.net). Draw.io — это многофункциональный инструмент, предназначенный для создания разнообразных графических схем, таких как диаграммы классов, диаграммы баз данных, блок-схемы, диаграммы деятельности, диаграммы процессов и многие другие.

На момент использования проекта была задействована актуальная веб-версия программного обеспечения. Для доступа и использования данного инструмента можно посетить официальный сайт: https://app.diagrams.net.

Приложение предоставило интуитивно понятный графический интерфейс для создания структурных схем, которые помогли визуализировать ключевые компоненты системы, а также их взаимосвязи и процессы взаимодействия. Использование draw.io стало важным шагом для проектирования системы, так как оно упростило процесс создания диаграмм, описывающих функциональные блоки приложения и их взаимодействие. Приложение также поддерживает интеграцию с различными облачными сервисами, что облегчило совместную работу над схемами и предоставило гибкость в управлении проектами.

Инструмент draw.io был выбран за его следующие особенности:

* Доступность: кроссплатформенная, поддержка работы в браузере, а также на Windows, macOS и Linux.
* Поддержка стандартов: возможность создания UML–диаграмм, блок–схем, диаграмм ER, IDEF0 и IDEF3, что позволило эффективно визуализировать бизнес–процессы и архитектуру системы.
* Легкость в использовании: интуитивно понятный интерфейс, который позволяет создавать диаграммы без необходимости владения сложными графическими навыками.
* Интеграция с облачными хранилищами: поддержка Google Drive, OneDrive, GitHub, что упрощает доступ к схемам и их совместное редактирование.

Основное назначение draw.io в этом проекте заключалось в создании визуальных моделей бизнес–процессов и архитектурных решений, которые легли в основу разработки системы. Эти диаграммы помогли в структурировании процессов разработки и визуализации потоков данных, что обеспечило более четкое понимание системы в целом.

Данные о технологии draw.io:

* Разработчик: JGraph Ltd.
* Адрес загрузки: [app.diagrams.net](https://app.diagrams.net).
* Использовался для: создания диаграмм и визуальных моделей бизнес–процессов и архитектурных решений системы.
* Доступность: кроссплатформенная, поддержка веб–версии и настольных приложений.

# 3 Описание практического задания

# 3.1 Список пользователей и их роли (актеры)

Роль гостя ограничена базовыми функциями взаимодействия с приложением. Функции: регистрация нового аккаунта, авторизация, просмотр статей.

Пользователь – основной актер, использующий основной функционал приложения.

Функции:

* Просмотр пунктов вторсырья.
* Обмен накопленных баллов на скидки.
* Создание, изменение и удаление статей.
* Оценка статей (лайк/комментарий).
* Получение справочной информации (сколько новой продукции будет сделано из старых отходов.

Администратор имеет доступ к функциям управления системой и пользователями.

Функции:

* Добавление новых пунктов приема вторсырья.
* Управление видами вторсырья (добавление/изменение/удаление).
* Управление скидками (добавление/изменение/удаление).
* Управление статьями (изменение/удаление).

# 3.2 Иерархия актеров

* Администратор (высший уровень, доступ ко всем функциям).
* Пользователь (доступ ко всем основным функциям системы).
* Гость (ограниченный доступ, только регистрация, авторизация, просмотр статей).

# 3.3 Прецеденты подсистемы и схема взаимодействия

Прецеденты – это конкретные случаи использования системы, описывающие взаимодействие актеров с приложением:

Гость: регистрация нового аккаунта, авторизация, просмотр статей.

Пользователь: просмотр пунктов приема вторсырья, обмен накопленных баллов на скидки, создание/изменение/удаление статей, оценка статей, получение справочной информации.

Администратор: добавление новых пунктов приема вторсырья, добавление/изменение/удаление видов вторсырья, добавление/изменение/удаление скидок, изменение/удаление статей.

# 4 Взаимодействие актеров с «публичными» прецедентами

Гость взаимодействует с публичными прецедентами регистрации и авторизации, чтобы получить доступ к расширенному функционалу.

Пользователь после авторизации пользователь получает доступ ко всем основным публичным прецедентам системы, таким как создание статей, просмотром пунктов приема сдачи вторсырья.

Администратор имеет доступ к публичным и административным прецедентам, таким как управление статьями, пунктами приема вторсырья и видов вторсырья.

На рисунке 4.1 представлена диаграмма использования.

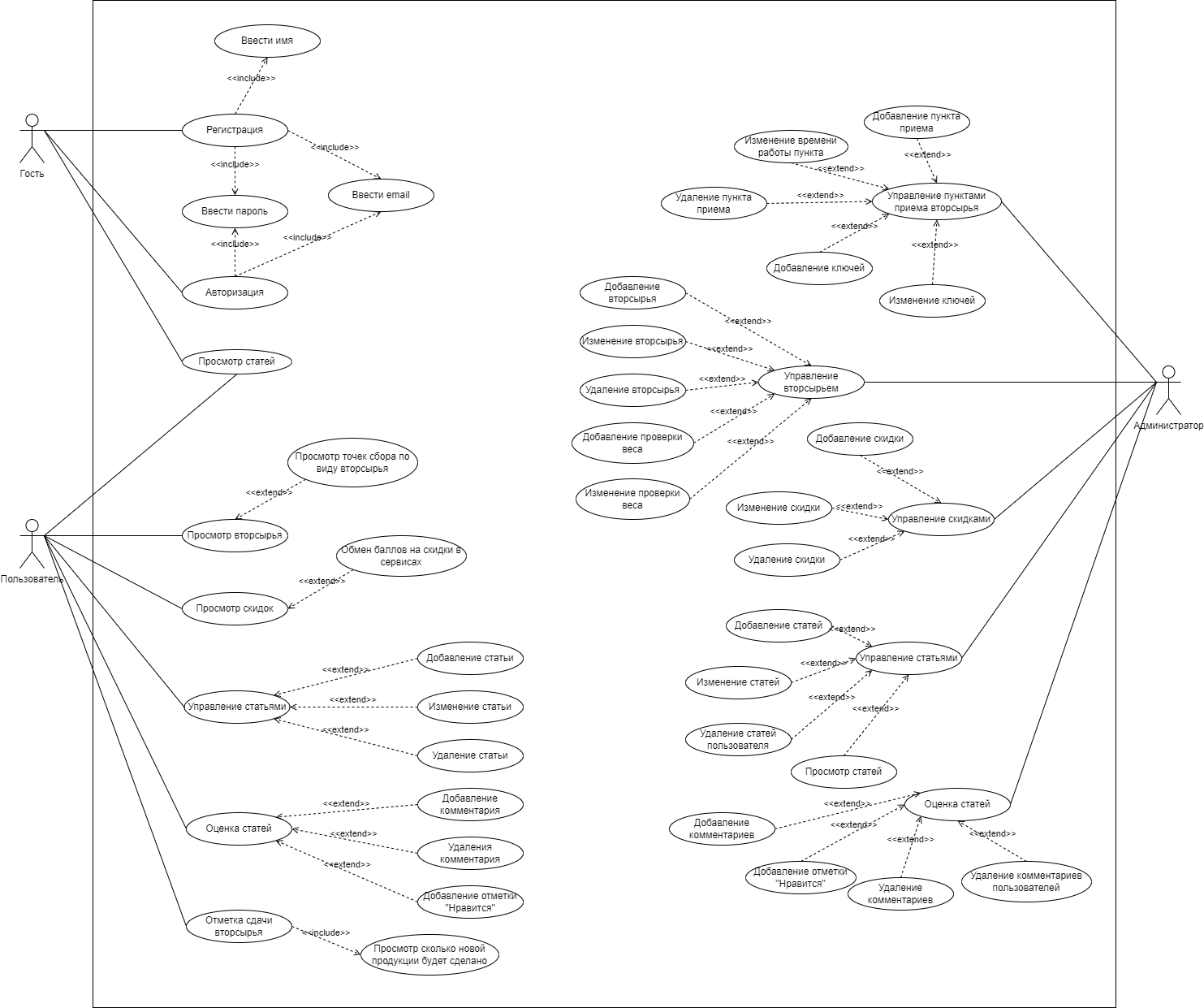


Рисунок 4.1 – Диаграмма использования

На диаграмме видно, какие прецеденты относятся к определенным ролям.

# 5 Ответы на вопросы

1. Дайте описание понятиям Unified process (UP) и UML.

Unified Process (UP, англ. унифицированный процесс) — процесс разработки ПО, который обеспечивает упорядоченный подход к распределению задач и обязанностей в организации – разработчике.

UML (Unified Modeling Language) — это язык визуального моделирования, который используется для описания структуры и поведения программных систем. UML позволяет разработчикам, аналитикам и менеджерам общаться на одном языке, обеспечивая понимание проекта с разных точек зрения. UML используется для создания различных диаграмм, которые помогают в проектировании архитектуры системы и документации кода.

2. Перечислите основные диаграммы UML 2.0.

Основные диаграммы UML 2.0 включают несколько категорий. Это диаграммы для моделирования структуры системы (например, диаграмма классов и диаграмма объектов), диаграммы для моделирования поведения (диаграмма последовательностей, диаграмма состояний), а также диаграммы для моделирования взаимодействия (например, диаграмма коммуникаций).

3. Назовите CASE-средства, поддерживающие создание UML диаграмм.

Существует множество CASE-средств, поддерживающих создание UML-диаграмм, среди которых популярны такие программы, как Rational Rose, Enterprise Architect, Visual Paradigm, Lucidchart и StarUML. Эти инструменты помогают разработчикам создавать модели программных систем на основе стандартов UML и упрощают процесс проектирования и документации.

4. Укажите назначение диаграммы вариантов использования.

Иллюстрирует возможные сценарии внешнего взаимодействия пользователей (Actors) с прецедентами (Use cases) системы, т.е. описывает функциональное назначение системы. Диаграмма отображает различные сценарии использования системы и описывает, как пользователи взаимодействуют с ней для достижения своих целей.

5. Опишите нотации, которые используются для построения Use-Case диаграммы.

Для построения диаграммы вариантов использования (Use Case) используются следующие нотации: акторы изображаются в виде человечков, а сами прецеденты – в виде овалов, которые описывают функциональность. Линии, связывающие акторов и прецеденты, показывают взаимодействие между пользователем и системой.