Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

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

Студент: Коршун Н. И.

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 каталог автомобилей «AUTOSTOP».

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 NextJS. 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:

* обеспечивать возможность просмотра информации о различных моделях автомобилей (характеристики, фотографии);
* предоставлять функционал просмотра видеообзоров на автомобили;
* позволять пользователю сравнивать характеристики нескольких автомобилей;
* обеспечивать регистрацию и авторизацию пользователей с возможностью восстановления пароля через электронную почту
* поддерживать роли гостя, пользователя, администратора;
* для гостей доступен просмотр автомобилей и видеообзоров;
* для зарегистрированных пользователей доступен функционал сохранения избранных автомобилей и добавление автомобилей в список для сравнения;
* поддерживать функционал торговой площадки для продажи и покупки автомобилей с возможностью размещения объявлений, загрузки фотографий и контактов;
* администратор может добавлять и редактировать информацию о автомобилях, видеообзорах и управлять пользователями;
* поддерживать фильтрацию и поиск по различным параметрам автомобилей (марка, год выпуска, тип двигателя и т.д.);

Основные задачи приложения:

* разработать удобный интерфейс для поиска информации о автомобилях;
* предоставить функционал для просмотра видеообзоров на автомобили;
* создать систему сравнения характеристик автомобилей для более детального анализа.

# 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.

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: 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 информации автомобилей.

Пользователь – основной актер, использующий основной функционал приложения.

Функции:

* Просмотр информации о автомобилях.
* Сравнение автомобилей.
* Сохранение автомобилей в избранное.
* Просмотр видеообзоров.
* Добавление объявлений на торговую площадку.

Администратор имеет доступ к функциям управления системой и пользователями.

Функции:

* Добавление новых моделей автомобилей.
* Редактирование и удаление информации о моделях автомобилей.
* Управление видеообзорами на автомобили (добавление/удаление).
* Управление пользователями (изменение/удаление).
* Модерация добавляемых объявлений на торговую площадку.

# 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 информации о автомобилях).

# 3.3 Прецеденты подсистемы и схема взаимодействия

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

Гость: регистрация нового аккаунта, авторизация, просмотр информации о автомобилях.

Пользователь: просмотр информации о автомобилях, сравнение автомобилей, сохранение автомобилей в избранное, просмотр видеообзоров, добавление объявлений на торговую площадку.

Администратор: добавление/изменение/удаление информации о моделях автомобилей, добавление/удаление видеообзоров на автомобили, изменение/удаление пользователей, модерация добавляемых объявлений на торговую площадку.

# 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 объявлений на торговой площадке, просмотром информации о моделях авто.

Администратор имеет доступ к публичным и административным прецедентам, таким как управление автомобилями, видеообзорами и пользователями.

На рисунке 4.1 представлена диаграмма использования.

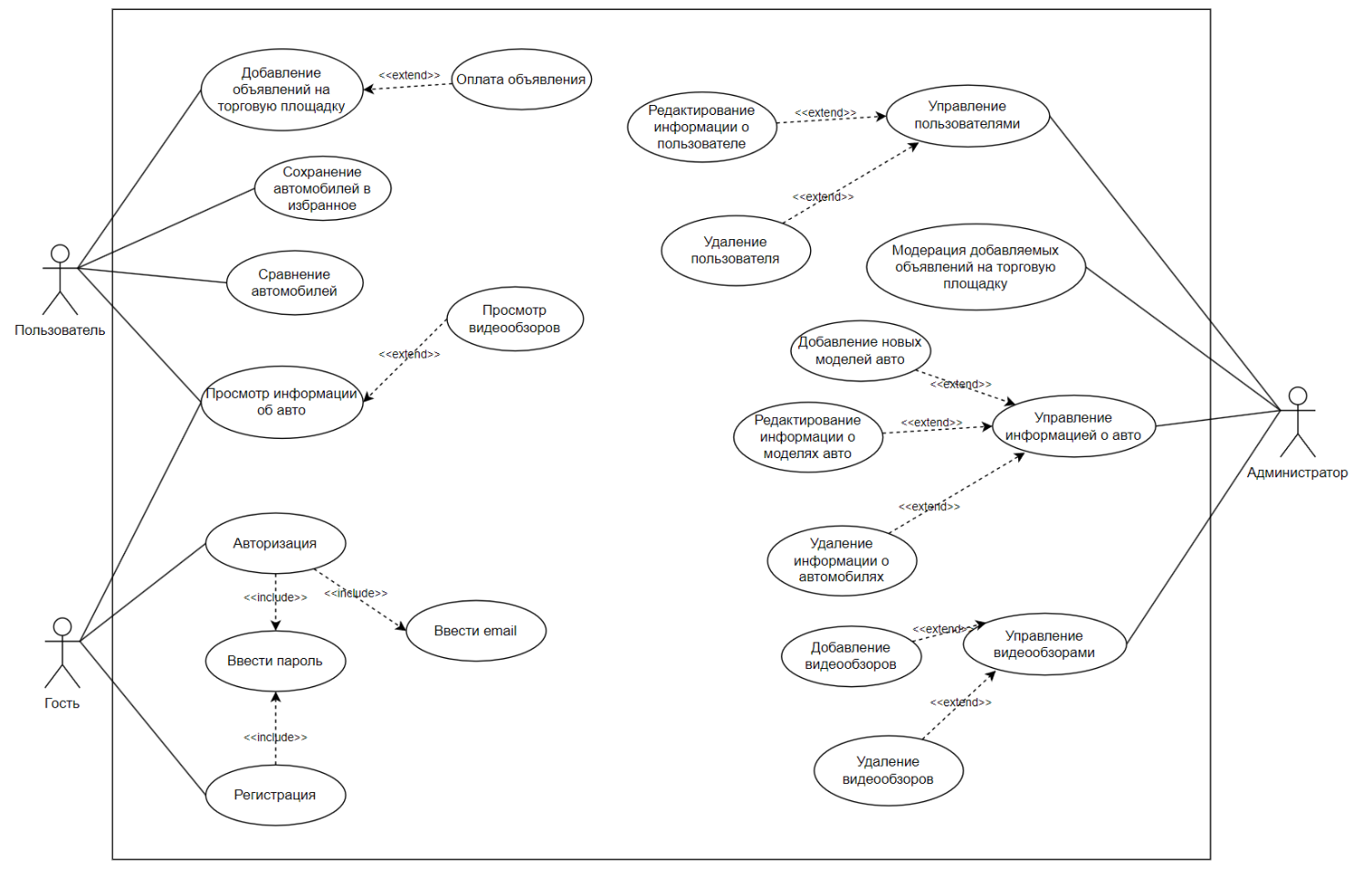


Рисунок 4.1 – Диаграмма использования

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, англ. унифицированный язык моделирования) — 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.