Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 4.

«Объектно-ориентированное моделирование. UML – диаграммы поведения»

Студент: Точило О. В.

ФИТ 4 курс 4 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# Постановка задачи

Темой данной лабораторной работы является построение UML-диаграммы вариантов использования, необходимое для графического представления взаимосвязи ролей конечных пользователей и вариантов взаимодействий с системой, представленной в первой лабораторной работе – web-приложением переводчика статей «GPTranslate».

Целью лабораторной работы является изучение методологии объектно-ориентированного моделирования средствами UML, ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения и получение навыков проектирования функциональности информационной системы с применением UML. Создание диаграмм вариантов использования позволит визуализировать сценарии взаимодействия пользователей с системой, что особенно важно для выявления потенциальных проблем и улучшения пользовательского опыта. Это поможет более глубоко понять потребности конечных пользователей и сформировать четкие требования к функционалу системы.

Создание диаграммы вариантов использования предназначено для:

* идентификации всех значимых ролей пользователей (актеров): К примеру, обычный пользователь, зарегистрированный пользователь, администратор системы. Диаграмма позволит формализовать доступную каждой роли функциональность.
* определения всех вариантов использования;
* анализа взаимосвязей между вариантами использования: Например, "Регистрация пользователя" предшествует "Переводу текста", а "Сохранение истории переводов" связано с "Просмотром истории переводов". Данный анализ выявляет зависимости и потенциальные конфликты;
* уточнения требований к системе;
* определения приоритетов разработки;
* минимизации рисков и затрат;
* повышения качества пользовательского опыта;

Кроме того, данный подход способствует более качественной разработке системы, позволяя команде разработчиков заранее предугадать, какие функции будут наиболее важны для пользователей, и сосредоточиться на их реализации. Процесс построения диаграмм вариантов использования также помогает выявить возможные сценарии, которые могли быть упущены на ранних этапах, тем самым минимизируя риски во время разработки.

Также стоит отметить, что использование UML в проектировании позволяет создать общую визуальную основу для всех участников проекта, включая разработчиков, тестировщиков и бизнес-аналитиков. Это облегчает коммуникацию между ними, так как все могут опираться на одну и ту же модель системы. В результате успешное выполнение данной лабораторной работы не только улучшит навыки работы с UML, но и поможет в дальнейшем развитии системы, обеспечивая её гибкость и адаптивность к изменяющимся условиям.

# Описание функциональных требований

Функциональные требования к системе можно разделить на требования к функционалу для различных ролей приложения: пользователя, гостя, модератора и администратора.

Функционал для пользователя:

* загрузка исходной статьи: возможность вставки текста напрямую или загрузки файла с помощью встроенного редактора или файлового менеджера;
* выполнение перевода статьи: автоматический запрос системы на выполнение перевода, с возможностью выбора языка оригинала и целевого языка;
* оценка перевода статьи: шкала оценки с возможностью оставлять комментарии;
* создание жалобы на перевод статьи: простой и удобный интерфейс для описания проблем с переводом, которые должны быть рассмотрены модератором;
* редактирование личной информации: поле для ввода email адреса, кнопка для смены пароля, возможность изменения отображаемого имени.

Функционал для модератора:

* просмотр открытых жалоб: список актуальных проблем с переводами, отсортированный по времени поступления;
* удовлетворение и отклонение жалоб: простая кнопочная система для принятия решения, с возможностью добавления комментариев к жалобе.

Функционал для администратора:

* просмотр и редактирование списка всех пользователей: таблица с информацией о пользователях, возможность фильтрации по ролям и сортировки по различным параметрам;
* просмотр и редактирование списка моделей перевода: панель управления языками, где можно добавлять новые языки, изменять названия или удалять ненужные языки;
* просмотр и редактирование запросов перевода: список всех текущих запросов, с возможностью фильтрации по статусу и дате создания.

Функционал для гостя:

* регистрация: простая форма регистрации с обязательным указанием email адреса и пароля;
* аутентификация: получение доступа к возможностям зарегистрированной ранее учётной записи по адресу email и паролю.

# Описание программных средств

Draw.io (ранее известный как diagrams.net) - это мощный и многофункциональный онлайн-инструмент для создания диаграмм и схем. Разработанный компанией JGraph Ltd., этот веб-ресурс предоставляет широкий спектр инструментов для визуального проектирования и представления концепций.

Основные характеристики и возможности Draw.io:

* гибкость и универсальность: Draw.io предлагает возможности работы с многими типами диаграмм, включая функциональные диаграммы (IDEF0, UML, BPMN), диаграммы потока, смысловые карты, диаграммы Венна, архитектурные диаграммы, мозговые штурмы, канбан-доски, схемы баз данных, элементы управления проектами;
* интерфейс и удобство использования: простой и интуитивно понятный интерфейс, возможность быстрого создания диаграмм с помощью готовых шаблонов, поддержка многоканального редактирования (включая мобильные устройства);
* совместимость и интеграция: поддержка импорта и экспорта различных форматов файлов (.vsdx, .gliffy, .lucidchart), возможность интеграции с популярными инструментами для совместной работы (Google Drive, Dropbox, OneDrive);
* расширенные функции: комментарии и заметки к диаграммам, возможность создания и управления библиотеками элементов;
* безопасность и конфиденциальность: защита проектов за счет шифрования, возможность настройки прав доступа к диаграммам;
* обучение и ресурсы: встроенная справка и учебные материалы, база знаний с примерами и инструкциями;
* поддержка и развитие: регулярные обновления с новыми функциями и улучшениями, активное сообщество пользователей и форумы для обсуждения.

Применение Draw.io в проекте GPTranslate:

* создание подробных функциональных диаграмм для каждого модуля системы;
* разработка архитектуры системы, показывающей взаимосвязь между различными компонентами;
* создание схемы потока данных, демонстрирующей, как информация передается через систему;
* создание диаграммы пользовательского интерфейса, показывающей все доступные функции и их взаимодействие;
* разработка диаграммы безопасности, отображающей меры защиты данных и пользователей.

# Описание практического задания

В ходе выполнения практического задания необходимо построить диаграмму вариантов использования.

Основными составляющими диаграммы являются следующие элементы:

* актёры представляют собой внешние по отношению к системе сущности, взаимодействующие с ней. Важно отметить, что актеры представляют роли, а не конкретных людей или системы. Один человек может выполнять несколько ролей, а одна роль может выполняться несколькими людьми. На диаграмме актеры обычно изображаются человечками (или другими условными обозначениями);
* прецеденты описывают конкретный функциональный сценарий взаимодействия актёра с системой. Это последовательность действий, приводящих к достижению определенной цели. Каждый прецедент должен иметь четко определенное название, отражающее его цель. На диаграмме прецеденты обычно изображаются овалами;
* связи показывают взаимодействие между актёрами и прецедентами, а также взаимосвязи между самими прецедентами. Существует несколько типов связей. Связи подразделяются на использование (отображает взаимодействие между актёром и прецедентом; она показывает, что данный актер может инициировать и участвовать в выполнении данного прецедента; на диаграмме изображается сплошной линией), обобщение (показывает, что один актёр или прецедент является специализацией другого, наследуя его свойства и поведение, но добавляя при этом свои уникальные особенности; на диаграмме изображается сплошной линией со стрелкой, указывающей от специализированного элемента к обобщенному), включение (копирование варианта использования, который появляется в нескольких вариантах) и расширение (более подробное дополнение варианта использования). Применение связей обобщения позволяет наследовать базовые функции между актерами, демонстрируя схожий функционал. Связи включения дают возможность повторно использовать элементы прецедентов в разных сценариях. Благодаря связи расширения можно детализировать процесс взаимодействия, добавляя специфические действия к основным сценариям, что способствует более детализированной проработке поведения системы.

При построении диаграммы вариантов использования для web-приложения переводчика статей «GPTranslate» используются 5 актёров:

1. Гость. Это конечный пользователь системы, который не прошел регистрацию или аутентификацию. Гость может зарегистрироваться и аутентифицироваться.
2. Пользователь. Это зарегистрированный и аутентифицированный пользователь, который может изменять список своих исходных и переведённых статей, изменять список жалоб на переводы своих статей, изменять список комментариев к своим жалобам и изменять список своих настроек переводчика.
3. Модератор. Это зарегистрированный и аутентифицированный пользователь с ролью модератора, который может просматривать, удовлетворять и отклонять открытые жалобы, а также оставлять комментарии к ним.
4. Администратор. Это зарегистрированный и аутентифицированный пользователь с ролью администратора, который может проматривать статистику жалоб на переводы статей, изменять список стилей перевода, моделей, использующихся для перевода, а также изменять список пользователей.
5. Любой аутентифицированный пользователь. Это пользователь, прошедший аутентификацию, который может просматривать список своих открытых сессий и завершать их, а также изменять свою информацию (отображаемое имя).

Пользователь, Модератор и Администратор связаны с актером Любой аутентифицированный пользователь отношением обобщения. Это означает, что у всех этих актёров есть возможность просмотра и закрытия своих сессий, а также изменения персональной информации. Однако каждый из этих актеров имеет свои дополнительные возможности.

Диаграмма вариантов использования представлена на рисунке 4.1.

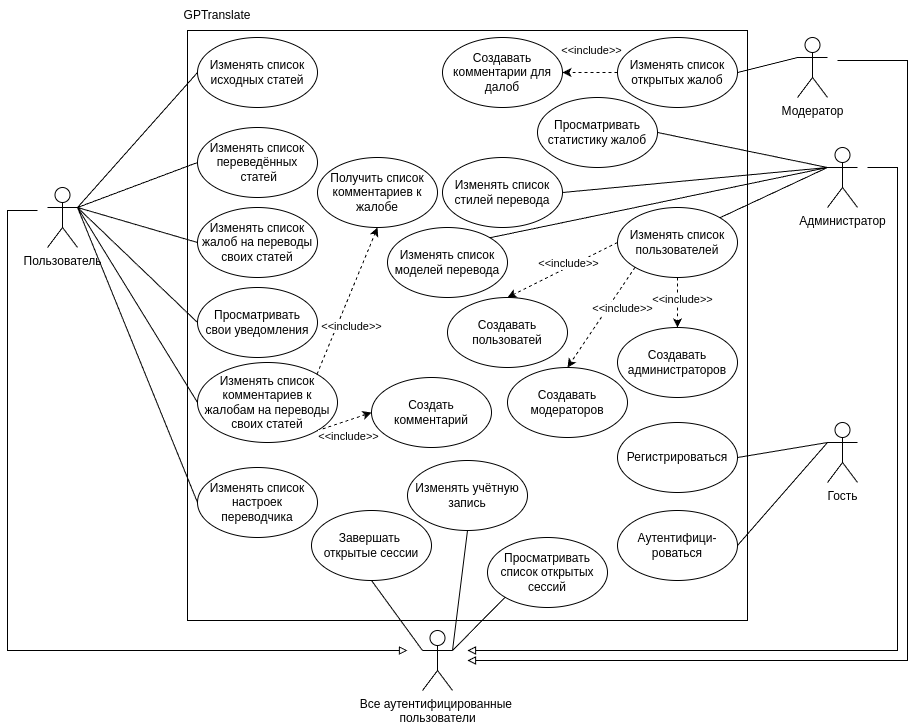


Рисунок 4.1 – Диаграмма вариантов использования

Рассмотрим подробнее прецеденты, выделенные в данной системе:

* регистрация (связана ассоциацией с Гостем);
* аутентификация (связана ассоциацией с Гостем);
* завершение открытых сессий (связано ассоциацией с актёром Все аутентифицированные пользователи)
* просмотр списка открытых сессий (связан ассоциацией с актёром Все аутентифицированные пользователи)
* изменение учётной записи (связано ассоциацией с актёром Все аутентифицированные пользователи)
* просмотр статистики жалоб (связан ассоциацией с Администратором)
* изменение списка стилей перевода (связано ассоциацией с Администратором)
* изменение списка моделей перевода (связано ассоциацией с Администратором)
* изменение списка пользователей (связано ассоциацией с Администратором)
* создание Пользователей (связан включением)
* создание Модераторов (связан включением)
* создание Администраторов (связан включением)
* изменение списка открытых жалоб (связано ассоциацией с Модератором)
* создание комментариев для жалоб (связано включением)
* изменение списка исходных статей (связано ассоциацией с Пользователем)
* изменение списка переведённых статей (связано ассоциацией с Пользователем)
* изменение списка жалоб на переводы своих статей (связано ассоциацией с Пользователем)
* изменение списка комментариев к жалобам на переводы своих статей (связано ассоциацией с Пользователем)
* получение списка комментариев к жалобе (связано включением)
* создание комментария (связано включением)

Данный список показывает, какие прецеденты могут включать другие или быть расширены, а также их связи с актерами в системе.

Отношения Include между вариантами использования обозначаются пунктирной линией с меткой <<include>>. Например, процесс изменения списка пользователей включает создание Модератора. Создание Администратора также включено в изменение списка пользователей.

Таким образом, отношение обобщения между актерами отражает иерархию доступа и прав в системе: базовый функционал актёра Все аутентифицированные пользователи расширяется для Пользователя, Модератора и Администратора, а Гость имеет ограниченный набор возможностей, не связанных с остальными актёрами.

# Ответы на теоретические вопросы

# Перечислите структурные диаграммы, которые входят в UML 2.0.

Структурные диаграммы в UML 2.0: В UML 2.0 существует несколько типов структурных диаграмм, которые описывают статические аспекты системы, такие как классы, объекты, компоненты и их взаимосвязи. Эти диаграммы включают:

* диаграмма классов (Class Diagram);
* диаграмма объектов (Object Diagram);
* диаграмма компонентов (Component Diagram);
* диаграмма развертывания (Deployment Diagram);
* диаграмма пакетов (Package Diagram);
* диаграмма композитной структуры (Composite Structure Diagram);
* диаграмма профилей (Profile Diagram).

# Укажите назначение структурных диаграммы

Структурные диаграммы UML предназначены для отображения статической структуры системы на различных уровнях детализации. Их цель — показать классы, компоненты, пакеты и другие элементы системы, а также их взаимосвязи. Основное назначение каждой диаграммы:

* диаграммы классов — отображают структуры классов и их отношения (ассоциации, обобщения и композиции);
* диаграммы объектов — представляют конкретные экземпляры классов в определенном состоянии;
* диаграммы компонентов — помогают визуализировать, как модули системы взаимодействуют друг с другом;
* диаграммы развертывания — описывают физическое развертывание системы на оборудовании;
* диаграммы пакетов — группируют элементы системы, обеспечивая модульную структуру;
* диаграммы композитной структуры — отображают внутренние части классов и их взаимодействия;
* диаграммы профилей — позволяют добавлять специализированные элементы для конкретных областей приложения.

# Опишите нотации, которые используются для построения Classes диаграмм

В диаграммах классов используется несколько ключевых нотаций:

* класс — прямоугольник, разделенный на три части: верхняя часть содержит имя класса, средняя — атрибуты (с переменными или данными), нижняя — методы или операции;
* ассоциация — линия, соединяющая классы, которая показывает, как они взаимодействуют. Могут быть добавлены кратности (например, 1..\*, 0..1), чтобы указать количество объектов;
* обобщение (наследование) — пустая треугольная стрелка, указывающая на суперкласс;
* реализация (интерфейс) — пунктирная линия с треугольной стрелкой, указывающей на интерфейс, который класс реализует;
* агрегация — линия с пустым ромбом на стороне класса-владельца, показывающая отношение «часть-целое»;
* композиция — линия с закрашенным ромбом, указывающая, что объект полностью владеет своими частями;
* зависимость — пунктирная линия со стрелкой, показывающая, что один элемент зависит от другого;
* модификаторы доступа — используются для обозначения уровня видимости атрибутов и методов: + (public), - (private), # (protected), ~ (package).

# Для чего применяются расширения диаграмм UML?

Расширения в UML используются для добавления специфических возможностей и элементов, которые не предусмотрены стандартными диаграммами. Они позволяют адаптировать UML для различных доменов и бизнес-сценариев.

Расширения включают:

* стереотипы — для добавления новых видов элементов в диаграммы, которые могут иметь специальные свойства;
* тегированные значения — для хранения дополнительных данных или атрибутов, не включенных в стандартные элементы;
* ограничения — для выражения условий или правил, которые должны быть выполнены (например, на атрибуты или ассоциации).

# Что означают понятия «стереотип» и «тегированное значение» в контексте расширенных диаграмм?

Стереотип — это механизм расширения UML, который позволяет создавать новые типы элементов на основе существующих. Стереотипы визуально обозначаются угловыми скобками (например, <<controller>>, <<entity>>). Они могут использоваться для спецификации, например, ролей элементов, таких как «Контроллер» или «Сущность» в архитектурных моделях.

Тегированное значение (tagged value) — это дополнительный атрибут или метка, которую можно прикрепить к элементу модели (например, классу или ассоциации) для хранения дополнительных данных. Эти значения могут использоваться для хранения информации, важной для домена, но не предусмотренной в стандартных элементах UML (например, «версия», «автор»).

Стереотипы и тегированные значения позволяют расширять функциональность UML и адаптировать его под специфические нужды проекта.

# Заключение

В рамках данной лабораторной работы была проведена детальная разработка диаграммы вариантов использования для web-приложения переводчика статей «GPTranslate». Основной целью работы являлось получение практических навыков построения диаграмм с использованием унифицированного языка моделирования (UML) для проектирования функциональной структуры системы. Эти навыки являются важными для успешного проектирования и разработки информационных систем, так как UML предоставляет стандартизированный подход к моделированию процессов и сценариев взаимодействия пользователей с системой.

На этапе проектирования было смоделировано основные сценарии взаимодействия пользователей с системой, в которой участвуют четыре основных актёра: Гость, Пользователь, Модератор и Администратор. Каждый актёр выполняет определенные действия в рамках своих полномочий. Например, Гость может зарегистрироваться, но для выполнения перевода статьи ему необходимо аутентифицироваться в системе, становясь Пользователем.

Моделирование различных вариантов использования позволило лучше понять взаимодействие пользователей с системой. Например, процесс изменения списка комментариев к жалобе на перевод своей статьи включает получение списка комментариев и создание комментария. Система поддерживает возможность удаления ненужных переводов, учитывая изменяющиеся планы клиентов. При моделировании выявлено использование отношений обобщения и включения между прецедентами. Например, акторы Пользователь, Модератор и Администратор отеля связаны с актёром Все аутентифицированные пользователи через обобщение, что означает, что они наследуют все его возможности, дополняя их дополнительными функциями. Это помогает снизить количество повторяющихся действий.

Диаграмма вариантов использования структурирует функциональность системы и помогает организовать бизнес-процессы. Она предоставляет визуальное представление о взаимодействиях между пользователями и системой, что помогает разработчикам и менеджерам проектов понять, как должна функционировать система. Это позволяет заранее выявить возможные проблемы и оптимизировать процесс разработки. Такой подход позволяет предусмотреть все сценарии использования системы, что упрощает процесс дальнейшей разработки. Структурированное моделирование помогает всем участникам разработки понимать ключевые процессы и функции системы, что снижает вероятность ошибок и недочетов на этапе разработки и тестирования.

Таким образом, построение диаграммы вариантов использования для web-приложения переводчика статей «GPTranslate» является ключевым этапом в проектировании и разработке, позволяющим учесть все сценарии работы пользователей с системой и обеспечить высокое качество конечного продукта.