Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 4.

Объектно-ориентированное моделирование. UML-диаграммы поведения.

Студент: Яшный Н. С.

ФИТ 4 курс 4 группа

Преподаватель: Якубенко К. Д.

Минск 2024

1. **Тема и цель работы**

Темой данной лабораторной работы является построение UML-диаграммы вариантов использования, которая выступает важным инструментом для графического представления взаимосвязей между ролями конечных пользователей и вариантами взаимодействия с системой. В данном случае речь идет о сервисе поиска недвижимости «HomeHub», который был представлен в первой лабораторной работе. Этот сервис предоставляет пользователям возможность находить, сравнивать и управлять объектами недвижимости, включая квартиры, дома и коммерческую недвижимость. В условиях современного рынка, где информационные технологии играют ключевую роль, такой сервис становится особенно актуальным и востребованным, обеспечивая удобство и доступность информации для пользователей.

Целью лабораторной работы является глубокое изучение методологии объектно-ориентированного моделирования с использованием средств UML. Этот процесс включает в себя не только знакомство с основными принципами объектно-ориентированного проектирования программного обеспечения, но и практическое применение этих принципов при создании диаграмм. Важно отметить, что UML предоставляет стандартизированный язык для описания различных аспектов программной системы, что способствует более эффективному общению между членами команды разработки. В процессе работы студенты получат необходимые навыки для проектирования функциональности информационной системы, что является критически важным для успешного выполнения задач, связанных с разработкой программного обеспечения.

Изучение UML-диаграмм вариантов использования позволяет не только лучше понять, как конечные пользователи взаимодействуют с системой, но и выявить потребности различных групп пользователей. Это знание помогает определить, какие функции они могут выполнять и какие требования предъявляются к разработчикам для реализации этих функций. Кроме того, графическое представление взаимодействий создает общее представление о системе, что упрощает дальнейшее тестирование и поддержку. Это делает UML-диаграммы неотъемлемой частью современного подхода к разработке программного обеспечения, обеспечивая структурированный и организованный процесс проектирования.

Таким образом, данная лабораторная работа направлена на формирование у студентов практических и теоретических навыков, необходимых для качественного и эффективного проектирования информационных систем. Она также способствует углублению понимания объектно-ориентированного подхода в программировании и важности системного мышления при разработке сложных программных решений. Студенты смогут перенести полученные знания на реальные проекты, что значительно повысит их профессиональную подготовку в сфере разработки программного обеспечения.

1. **Описание функциональных требований**

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

Функционал для пользователя:

* обеспечивать возможность регистрации, авторизации и аутентификации;
* просмотр каталога недвижимости;
* поиск и фильтрация недвижимости;
* просмотр подробной информации о недвижимости;
* просмотр информации о продавце;
* добавление объявлений в избранное;
* публикация собственных объявлений;
* удаление собственных объявлений;
* просмотр новостей;
* возможность оставить комментарий;
* просмотр геолокации недвижимости на карте.

Функционал для администратора:

* просмотр списка всех пользователей;
* публикация новостей;
* возможность изменить роль пользователя;
* возможность заблокировать/разблокировать пользователя;
* удаление любого объявления;
* просмотр геолокации недвижимости на карте.

Функционал для гостя:

* регистрация;
* просмотр каталога недвижимости;
* поиск и фильтрация недвижимости;
* просмотр подробной информации о недвижимости;
* просмотр информации о продавце;
* просмотр новостей;
* просмотр геолокации недвижимости на карте.

1. **Описание программных средств**

Для построения UML-диаграмм вариантов использования использовался веб-ресурс Draw.io, разработанный компанией JGraph Ltd. Этот инструмент предназначен для создания различных типов диаграмм и визуализаций, что делает его незаменимым помощником для разработчиков, проектировщиков и менеджеров. Адрес веб-ресурса – <https://www.drawio.com>.

Draw.io доступен на всех платформах, которые имеют веб-браузер и доступ в Интернет, что обеспечивает его универсальность и удобство использования. Пользователи могут работать с ним на различных устройствах, включая настольные компьютеры, ноутбуки и даже мобильные устройства, что делает процесс создания диаграмм еще более гибким и доступным. Также стоит отметить, что Draw.io не требует регистрации или создания учетной записи, что упрощает доступ к инструменту и позволяет быстро начать работу.

Функционал веб-ресурса Draw.io весьма разнообразен и включает в себя возможность построения не только UML-диаграмм, но и других графических элементов, таких как графики, смысловые карты, диаграммы Венна, Agile и Kanban доски, а также графики мозговых штурмов и диаграммы архитектур технических систем. Это позволяет пользователям создавать визуализации, которые соответствуют самым различным потребностям и задачам, будь то разработка программного обеспечения, проектирование бизнес-процессов или образовательные цели. Например, возможность создания диаграмм рабочего процесса может быть полезной для оптимизации бизнес-операций, а диаграммы архитектуры могут помочь в представлении сложных технических систем.

Кроме того, Draw.io предлагает интеграцию с популярными облачными хранилищами и платформами, такими как Google Drive, OneDrive, GitHub и многие другие. Эта функциональность позволяет пользователям легко сохранять свои работы и делиться ими с коллегами, что особенно важно для командной работы и совместного проектирования. Интеграция с такими платформами также обеспечивает возможность совместного редактирования в реальном времени, что значительно улучшает взаимодействие между членами команды.

Таким образом, использование Draw.io для создания UML-диаграмм вариантов использования не только упрощает процесс визуализации, но и значительно увеличивает эффективность работы команды, позволяя сосредоточиться на ключевых аспектах проектирования и разработки. Это делает инструмент идеальным выбором для организаций, стремящихся к повышению производительности и качеству своей работы в области разработки программного обеспечения и управления проектами.

1. **Описание практического задания**

В ходе выполнения практического задания необходимо построить диаграмму вариантов использования.

Основными составляющими диаграммы являются актёры, преценденты и связи. Актеры представляют некоторую роль, которую выполняют конечные пользователи в системе. Преценденты отражают варианты действий, которые могут выполнить в системе актеры.

Связи подразделяются на использование (между актером и прецендентом), обобщение (преценденты, свойственные некоторым актерам), включение (копирование варианта использования, который появляется в нескольких вариантах) и расширение (более подробное дополнение варианта использования).

Диаграмма вариантов использования представлена на рисунке 4.1.

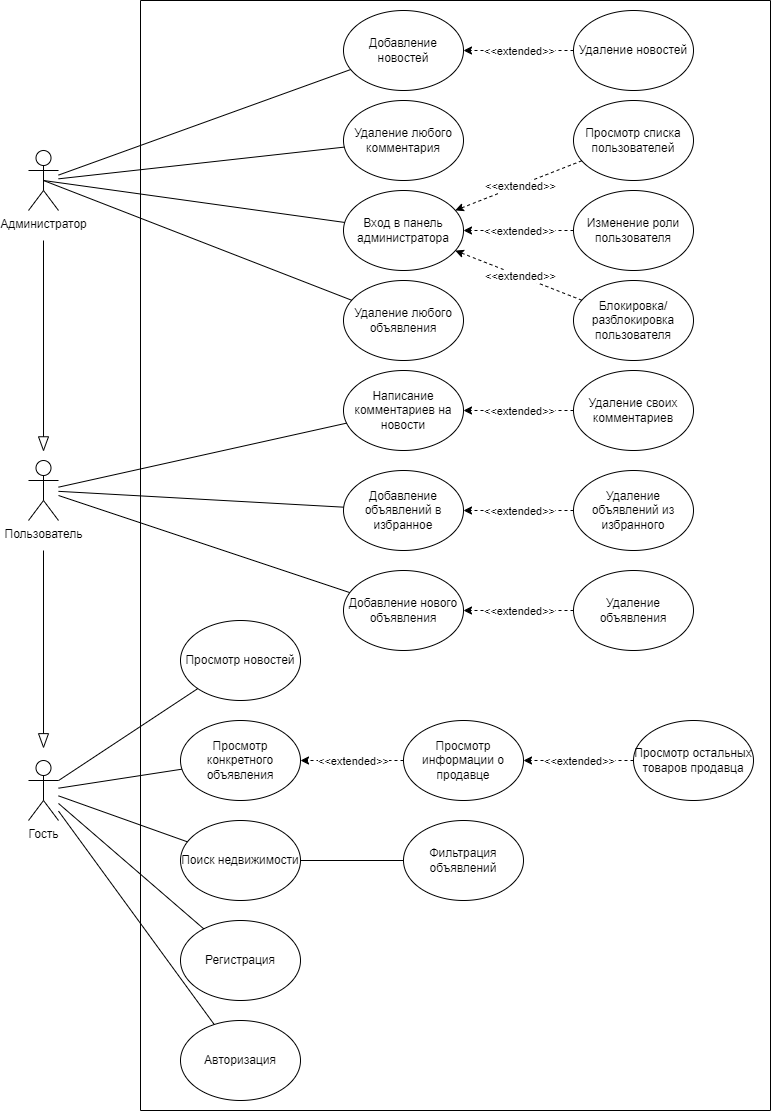


Рисунок 4.1 – Диаграмма вариантов использования приложения «HomeHub»

При построении диаграммы вариантов использования для программного средства «HomeHub» используются 3 актёра:

* Гость. Это конечный пользователь сервиса, не прошедший регистрацию и/или авторизацию.
* Пользователь. Это конечный пользователь сервиса, прошедший авторизацию и имеющий возможность публиковать и удалять собственные объявления, оставлять комментарии на новости и добавлять объявления в избранное.
* Администратор. Это пользователь с наивысшей привилегией, имеющий возможность удалять любые объявления, публиковать и удалять новости, изменять роли пользователей, блокировать пользователей и просматривать полную информацию о пользователях в панели администратора.

Актер Пользователь связан с актером Гость отношением обобщения. Таким образом, у Пользователя есть такой же функционал, как у Гостя. Актер Администратор связан с актером Пользователь отношением обобщения. Таким образом, у Администратора есть такой же функционал, как у Пользователя.

Рассмотрим подробнее преценденты, выделенные в данной системе:

* регистрация (связано ассоциацией с Гостем и Пользователем);
* авторизация (связано ассоциацией с всеми актерами);
* просмотр новостей (связано ассоциацией со всеми актерами);
* просмотр конкретного объявления (связано ассоциацией со всеми актерами);
* просмотр информации о продавце (связано расширением с просмотром конкретного объявления);
* просмотр остальных товаров продавца (связано расширением с просмотром информации о продавце);
* поиск недвижимости (связано ассоциацией со всеми актерами);
* фильтрация объявлений (связано ассоциацией с поиском недвижимости);
* добавление нового объявления (связано ассоциацией с авторизацией);
* удаление объявления (связано расширением с добавлением объявления);
* добавление объявлений в избранное (связано ассоциацией с авторизацией);
* удаление объявлений из избранного (связано расширением с добавлением объявлений в избранное);
* написание комментариев на новости (связано ассоциацией с авторизацией);
* удаление своих комментариев (связано расширением с написанием комментариев на новости);
* добавление новостей (связано ассоциацией с авторизацией у администратора);
* удаление новостей (связано расширением с добавлением новостей);
* удаление любого комментария (связано ассоциацией с авторизацией у администратора);
* удаление любого объявления (связано ассоциацией с авторизацией у администратора);
* вход в панель администратора (связано ассоциацией с авторизацией у администратора);
* просмотр всех пользователей (связано расширением со входом в панель администратора);
* изменение роли пользователя (связано расширением со входом в панель администратора);
* блокировка/разблокировка пользователя (связано расширением со входом в панель администратора).

1. **Ответы на вопросы**

Следующие ответы на вопросы:

* Дайте описание понятиям Unified process (UP) и UML: Unified Process методология разработки программного обеспечения, которая описывает процесс создания системы в терминах фаз, итераций и артефактов. Она основана на принципах итеративной и инкрементальной разработки, что позволяет гибко реагировать на изменения и эффективно управлять рисками. UML (Unified Modeling Language) графический язык, используемый для визуализации, спецификации, проектирования и документирования программных систем. UML служит средством для создания различных диаграмм, которые помогают разработчикам и аналитикам лучше понять архитектуру и функциональность системы.
* Перечислите основные диаграммы UML 2.0: диаграммы классов, диаграммы объектов, диаграммы последовательностей, диаграммы коммуникаций, диаграммы состояний, диаграммы активностей, диаграммы развёртывания.
* Назовите CASE-средства, поддерживающие создание UML диаграмм: к числу CASE-средств, поддерживающих создание UML диаграмм, относятся StarUML, Enterprise Architect, Visual Paradigm, Lucidchart, Draw.io, Microsoft Visio и Creately.
* Укажите назначение диаграммы вариантов использования: диаграмма вариантов использования предназначена для визуализации взаимодействия между пользователями (актерами) и системой через прецеденты. Она иллюстрирует функциональное назначение системы, описывая основные сценарии, в которых система будет использоваться. Диаграмма помогает определить требования пользователей и является основой для дальнейшего проектирования системы.
* Опишите нотации, которые используются для построения Use-Case диаграммы: нотации, используемые для построения Use Case диаграммы, включают несколько основных элементов. Актер (Actor) изображается как стилизованный человечек, представляющий типового пользователя или внешнюю систему, взаимодействующую с системой. Вариант использования (Use Case) представляется в виде овала с названием прецедента, описывающего функциональность системы. Связи (Associations) отображаются линиями, соединяющими актеров с вариантами использования, обозначая их взаимодействие. Отношения Include и Extend обозначаются стрелками с соответствующими стереотипами, указывая на зависимость между вариантами использования. Отношение Generalization представляется пустой стрелкой, показывающей обобщение между актерами. Примечания (Comments) добавляются для пояснения элементов диаграммы и изображаются в виде прямоугольников с заголовком «Comment», соединяющихся с элементами диаграммы линией.