Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 4.

«Объектно-ориентированное моделирование. UML – диаграммы поведения»

Студент: Ковкель Н. В.

ФИТ 4 курс 4 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# Теоретический материал

Дайте описание понятиям Unified process (UP) и UML.

Унифицированный процесс (Unified Process, UP) – это методология моделирования программных систем. Она указывает на исполнителей, действия и артефакты, которые необходимо использовать, осуществить или создать для моделирования программной системы.

UML (Unified Modeling Language) - это универсальный язык визуального моделирования систем.

Перечислите основные диаграммы UML 2.0.

Существует два основных типа диаграмм UML: структурные диаграммы и поведенческие диаграммы (а внутри этих категорий имеется много других). Эти варианты существуют для представления многочисленных типов сценариев и диаграмм, которые используют разные типы людей.

Назовите CASE-средства, поддерживающие создание UML диаграмм.  
CASE-средства, представляющие собой набор инструментов, предназначенный для автоматизации визуального моделирования, проектирования, документирования и генерации кода реализации на выбранном алгоритмическом языке. К таким средствам относятся CASE-средства визуального моделирования и проектирования  компании IBM Rational Software Corp — Rational Rose и [Rational Software Architect](https://brasmlibras.ru/rational-software-architect/" \t "_blank), продукт проектирования и интеграции компании Borland – Together и другие.

Укажите назначение диаграммы вариантов использования.

Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

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

В качестве иллюстративного примера рассмотрим систему онлайн-оплаты учебного курса. Пользователем этой системы является клиент. В терминологии UML он будет называться *актёр* – сущность за пределами системы, которая взаимодействует с ней. На UML-диаграмме Use Case он изображается в виде человечка. Актёру «Клиент» доступен основной вариант использования – «Оплатить договор» (на проведение обучающего курса по бизнес-анализу).  Расширением этого варианта использования является «Оплатить со скидкой по промокоду», который уменьшает сумму платежа. Этот вариант использования является опциональным и расширяет основной, поэтому он будет связан с основным через связь *extend,* которая выглядит как пунктирная стрелочка с соответствующей надписью.

На диаграмме использования изображаются:

* актёры — группы лиц или систем, взаимодействующих с нашей системой;
* варианты использования (прецеденты) — сервисы, которые наша система предоставляет актёрам;
* комментарии;
* отношения между элементами диаграммы.

# Постановка задачи

# Тема и цель работы

Темой данной лабораторной работы является построение UML-диаграммы вариантов использования, которая необходима для графического представления взаимосвязи ролей конечных пользователей и различных вариантов взаимодействий с системой, представленной в первой лабораторной работе – мобильное приложение для знакомств под названием «EventFlow». Это приложение помогает пользователям находить потенциальных партнёров через систему свайпов, получать уведомления о взаимных симпатиях и общаться в чате.

Целью данной лабораторной работы является не только изучение методологии объектно-ориентированного моделирования с использованием средств UML, но и углубленное ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения. Лабораторная работа нацелена на получение практических навыков проектирования функциональности информационных систем с применением UML. Создание диаграмм вариантов использования играет важную роль, поскольку эти диаграммы позволяют визуализировать сценарии взаимодействия пользователей с системой и тем самым выявить потенциальные проблемы и области для улучшения пользовательского опыта. Это особенно важно в контексте приложений для знакомств, где удобство и интуитивность интерфейса оказывают значительное влияние на уровень вовлечённости и удовлетворённости пользователей.

Построение UML-диаграмм способствует более качественному процессу разработки системы. Это позволяет команде разработчиков заранее предсказать, какие функции окажутся наиболее значимыми для пользователей приложения, например, такие функции, как регистрация и авторизация, система лайков, активация чатов при взаимных симпатиях, а также возможность отправки сообщений или мультимедийных файлов. Процесс построения диаграмм вариантов использования помогает заранее выявить возможные сценарии взаимодействия, которые могли бы быть упущены на ранних стадиях проектирования, что минимизирует риски на этапе разработки.

Кроме того, использование UML в процессе проектирования системы предоставляет всем участникам проекта – разработчикам, тестировщикам и бизнес-аналитикам – общую визуальную основу для обсуждения и улучшения системы. Это значительно упрощает коммуникацию между членами команды, так как все участники могут опираться на единую модель системы. Более того, такая модель позволяет наглядно демонстрировать основные бизнес-процессы и возможности, обеспечивая согласованное понимание структуры приложения среди всех заинтересованных сторон.

# Описание функциональных требований

Функциональные требования к системе веб-приложения «EventFlow» можно разделить на требования для различных ролей пользователей – гостя и пользователя.

Функционально web-приложение должно:

* поддерживать роли гостя, пользователя;

Обеспечивать гостям возможности:

* зарегистрироваться;
* аутентифицироваться;
* авторизоваться.

Обеспечивать пользователям возможности:

* взаимодействовать через свайп-систему с потоком людей;
* ставить лайки;
* пропускать анкеты;
* редактировать профиль (логин, пароль, аватар, предпочтения);
* получать уведомления о новых сообщениях и взаимных симпатиях;
* общаться с другими пользователями через чат.

Основные страницы веб-приложения:

* страница регистрации;
* страница авторизации;
* главная страница;
* личный кабинет пользователя;
* страница уведомлений;
* чат;
* настройки;
* страница помощи и поддержки.

Основные страницы веб-приложения обеспечивают полное покрытие всех функций, необходимых для эффективного взаимодействия с приложением. Каждая из этих страниц играет ключевую роль в создании удобного и эффективного интерфейса, обеспечивая пользователей всем необходимым для успешного взаимодействия с приложением. Эти страницы разработаны с учетом принципов удобства использования и интуитивно понятного интерфейса, что способствует повышению удовлетворенности пользователей и эффективности работы с приложением.

# 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.

Разработчиком Draw.io является компания JGraph Ltd. 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](https://app.diagrams.net" \t "_new).

Draw.io поддерживает интеграцию с облачными сервисами, такими как Google Drive, OneDrive и GitHub, что позволяет легко сохранять и управлять проектами. Кроме того, программное обеспечение поддерживает работу с локальными файлами и экспорт моделей в популярные форматы (PNG, PDF, SVG, XML), что делает его универсальным и удобным в использовании.

Инструмент поддерживает интеграцию с такими популярными сервисами, как Google Workspace, Microsoft Office 365, Confluence и Jira. Это позволяет пользователям бесшовно внедрять диаграммы и схемы в документы.   
Draw.io доступен в двух режимах:

1. Веб-версия — основной способ использования программного средства, работающий через любой современный браузер. Поддерживается на всех платформах, включая Windows, macOS, Linux, а также мобильные устройства под управлением Android и iOS.
2. Десктопная версия — доступна для скачивания и установки на операционные системы Windows, macOS и Linux. Обе версии имеют идентичный функционал, однако настольная версия может быть предпочтительна для пользователей, работающих с чувствительными данными, так как она поддерживает полностью автономную работу без подключения к интернету.

Draw.io предоставляет все необходимые средства для работы с различными моделями и диаграммами, применяемыми в инженерии программного обеспечения, таких как:

* UML диаграммы классов, последовательностей, прецедентов;
* ERD диаграммы (сущность-связь) для моделирования баз данных;
* BPMN диаграммы бизнес-процессов;
* диаграммы архитектуры программных систем и сетевых решений;
* блок-схемы для описания алгоритмов и процессов;
* диаграммы потоков данных и управления;
* схемы организации структуры данных.

Использование данного программного средства значительно упростило процесс проектирования и предоставило возможность создания наглядных схем, что способствует лучшему пониманию логики работы системы всеми заинтересованными сторонами проекта.

# Описание практического задания

В ходе выполнения практического задания необходимо построить диаграмму вариантов использования.

Диаграмма вариантов использования для системы «EventFlow» — это как карта взаимодействий между пользователями и системой. Она помогает визуализировать, кто что делает и как всё связано. Основными составляющими диаграммы являются актёры, прецеденты и связи. Актёры представляют собой роли, которые выполняют конечные пользователи в системе, включая, например, администратора, пользователя и гостя. Прецеденты отражают варианты действий, которые могут выполнить в системе актёры, такие как запись на услугу, просмотр услуг или оставление отзыва.

Актёры взаимодействуют с системой через различные варианты использования (прецеденты), а связи между ними помогают показать, как всё это происходит. Эти связи могут обозначать обязательные шаги, такие как необходимость авторизации перед выполнением определенных действий, а также дополнительные опции, например, возможность оставить отзыв после завершения услуги. Такой подход обеспечивает ясность в понимании функционала системы и помогает разработчикам более эффективно планировать её дальнейшее развитие.

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

При построении диаграммы вариантов использования для программного средства «EventFlow» используются 2 актера:

* Гость. Это конечный пользователь сервиса, не прошедший регистрацию и/или авторизацию.
* Пользователь. Это конечный пользователь сервиса, прошедший авторизацию и имеющий возможность взаимодействовать через свайп-систему с другими пользователями, редактировать профиль, получать уведомления и общаться через чат.

Важность такой диаграммы заключается в её способности не только отразить текущие требования системы, но и послужить основой для дальнейших улучшений и изменений, поскольку все сценарии использования легко адаптируются к изменяющимся условиям разработки и бизнес-процессам. Кроме того, диаграмма вариантов использования служит важным инструментом для коммуникации между членами команды, обеспечивая единое понимание процессов и взаимодействий. Это особенно полезно на начальных этапах проекта, когда требуется согласование видения системы между всеми участниками, включая бизнес-аналитиков, дизайнеров и разработчиков.

Таким образом, диаграмма вариантов использования системы «EventFlow» является не только инструментом для визуализации функциональных возможностей и взаимодействий, но и важной частью проектной документации. Она позволяет команде разработчиков и заинтересованным сторонам легко понять требования и логику работы системы, улучшает коммуникацию между участниками проекта и служит основой для внесения изменений и улучшений в систему. Это делает диаграмму неотъемлемой частью процесса проектирования и разработки, способствуя созданию продукта, который максимально удовлетворяет потребности пользователей и бизнес-задачи.

Диаграмма вариантов использования представлена на рисунке 4.1.

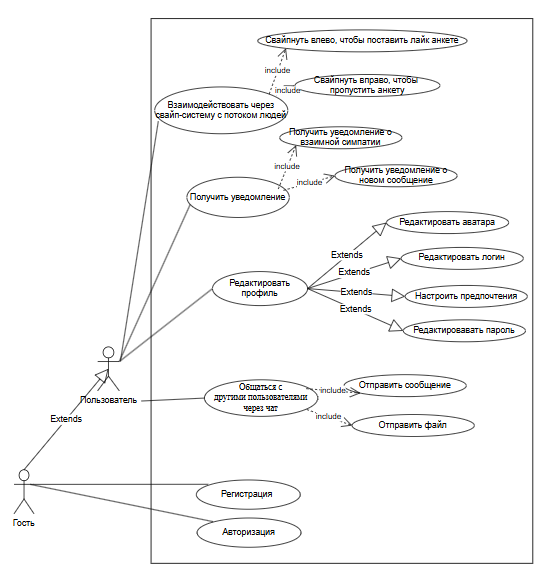


Рисунок 4.1 – Диаграмма вариантов использования

Диаграмма показывает зависимости между вариантами использования с помощью прямых и пунктирных линий, включая связи <<include>> и <<extend>>. Связь <<include>> обозначает, что один вариант использования включает другой, что помогает избежать дублирования. Например, указание рейтинга при добавлении отзыва.

Связь <<extend>> указывает на опциональные шаги, которые могут добавляться к основному процессу, например, возможность оставить отзыв после покупки товара. Эти зависимости упрощают понимание бизнес-логики системы и облегчают управление изменениями в будущем.

# Заключение

В ходе выполнения лабораторной работы была разработана UML-диаграмма вариантов использования для мобильного приложения «EventFlow» для знакомств. Этот процесс позволил не только визуализировать основные сценарии взаимодействия пользователей с приложением, но и определить ключевые роли, их взаимосвязь с функциональными возможностями системы. Благодаря этому стало возможным более четко сформулировать требования к системе, понять поведение пользователей, а также выявить потенциальные проблемы, которые могут возникнуть в процессе использования сервиса.

Создание диаграммы позволило выявить важные функциональные блоки, а также потенциальные риски, такие как ошибки при отправке сообщений, некорректная работа с уведомлениями или проблемы с процессом активации чата. Благодаря такому анализу стало возможным заранее предусмотреть механизмы, направленные на улучшение качества работы системы и минимизацию ошибок. Например, можно внедрить функции проверки данных или системы уведомлений, предупреждающие пользователей о возможных проблемах с приложением.

Построение UML-диаграммы дало возможность рассмотреть различные сценарии использования приложения, такие как регистрация, авторизация, свайп-система для поиска партнёров, отправка сообщений, а также активация чата при взаимной симпатии. Это помогло разработчикам создать более интуитивный и удобный интерфейс, а также выделить ключевые функции, которые необходимы для обеспечения качественной работы приложения. В результате «EventFlow» станет более отзывчивым к потребностям пользователей, предоставляя им удобный и функциональный сервис для знакомств. Диаграмма не только служит инструментом для визуализации процессов, но и станет основой для дальнейшего тестирования и улучшения системы. Она позволяет команде разработчиков сосредоточиться на потребностях пользователей и на создании наиболее востребованного функционала, что повысит конкурентоспособность приложения на рынке мобильных приложений для знакомств.

Таким образом, выполнение данной лабораторной работы позволило существенно укрепить навыки объектно-ориентированного моделирования и проектирования информационных систем с использованием UML. Полученные результаты будут использованы как основа для дальнейшей разработки и совершенствования приложения, а также для улучшения пользовательского опыта. Процесс анализа и визуализации функциональных возможностей помог не только структурировать требования к системе, но и создать платформу для обсуждения и внедрения будущих улучшений, что способствует созданию более качественного и удобного продукта для всех категорий пользователей.