Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 7.

«Объектно-ориентированное моделирование. Диаграммы поведения UML»

Студент: Ковкель Н.В.

ФИТ 4 курс 4 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# Теоретический материал

Укажите виды диаграмм поведения.

Всего существует 3 вида диаграмм поведения:

* диаграмма деятельности;
* диаграмма состояний;
* диаграмма вариантов использования.

Опишите назначение диаграммы деятельности.

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Опишите основные нотации, которые используются на диаграмме состояний.

Класс (Class), Состояние (State), Состояние (StateEx), Составное состояние (Composite state), Разделитель (Concurrent state), История (History), Глубокая история (Deep history), Начальное состояние (Start state), Конечное состояние (Final state), Синхронизатор/разветвитель (Complex transition), Переход (Transition), Сообщение (Event message), Точка изгиба связей (Point), Комментарий (Note), Коннектор комментария (Note connector).

Укажите виды связей между объектами на диаграмме последовательностей.

* Синхронное сообщение — отправитель передаёт ход управления актёру-получателю, которому необходимо провести в прецеденте некоторое действие. Пока проводимое получателем действие не будет завершено (соответственно, не будет получено ответное сообщение), отправитель теряет возможность производить какие-либо действия. Графически изображается как сплошная линия со стрелкой в виде закрашенного треугольника, после которой идёт прямоугольник, отражающий деятельность объекта, в конце которого находится ответное сообщение.
* Ответное сообщение — данное сообщение является ответом на синхронное сообщение. Обычно, содержит какое-либо возвращаемое изначальному отправителю значение, также возвращающее ему управление (возможность действовать). Графически изображается пунктирной линией с открытой стрелкой.
* Асинхронное сообщение — отправитель передаёт ход управления получателю, которому необходимо провести в прецеденте некоторое действие. Основное отличие от синхронного сообщения состоит в том, что отправитель не теряет возможности совершать другие действия. Графически изображается сплошной линией с открытой стрелкой.
* Потерянное сообщение — сообщение без адресата (есть отправитель, нет получателя).
* Найденное сообщение — сообщение без отправителя.

Какая диаграмма позволяет моделировать параллельные вычисления?

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

# Постановка задачи

# Тема и цель работы

Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения и получение практических навыков создания архитектуры информационной системы с использованием методологии UML. В рамках работы будут разработаны диаграммы деятельности и диаграммы последовательности для информационной системы для мобильного приложения для знакомств.

Диаграмма последовательности отобразит взаимодействие между объектами системы и участниками процесса во временной последовательности. Она покажет такие процессы, как запись клиента на услугу, обработка запросов на просмотр расписания, управление и хранение данных о клиентах и их записях, а также обмен данными между веб-сервером и базой данных. Диаграмма позволит визуализировать, как пользователи системы (клиенты, сотрудники, администраторы) взаимодействуют с различными компонентами системы, такими как сервер базы данных, пользовательский интерфейс и веб-сервер, а также последовательность обмена сообщениями между этими элементами. Это обеспечит наглядное представление потока данных и поможет выявить возможные области для оптимизации.

Диаграмма деятельности будет использоваться для визуализации логической последовательности действий в ключевых бизнес-процессах мобильного приложения для знакомств. Она позволит смоделировать такие процессы, как выбор анкет и взаимодействий с ними, созданием чатом и возможности отправки сообщения, ведение отчетности и аналитики, а также управление личными кабинетами пользователей и его предпочтениями. Диаграмма покажет, какие действия выполняют пользователи системы, в каком порядке они происходят, какие решения принимаются на каждом этапе, и как задачи распределяются по системе. Это поможет более детально проанализировать каждый процесс, выявить узкие места и определить пути для его улучшения.

Создание этих двух UML-диаграмм позволит глубже понять архитектуру системы, оптимизировать бизнес-процессы, улучшить взаимодействие между пользователями и компонентами системы, а также облегчить процесс анализа и разработки полноценного программного обеспечения для мобильного приложения для знакомств.

# Описание функциональных требований

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

Функционально web-приложение должно:

* поддерживать роли гостя, пользователя;

Обеспечивать гостям возможности:

* зарегистрироваться;
* аутентифицироваться,
* авторизоваться.

Обеспечивать пользователям возможности:

* возможность взаимодействовать через свайп-систему;
* редактирование профиля (логин, пароль, аватар, предпочтения);
* получение уведомлений о новых сообщениях и взаимных симпатиях;
* общение с другими пользователями через чат.

Основные страницы веб-приложения:

* страница регистрации;
* страница авторизации;
* главная страница;
* личный кабинет пользователя;
* страница уведомлений;
* чат;
* страница совпадения;
* 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 и администраторов всем необходимым для успешного взаимодействия с приложением.

Таким образом, функциональная модель мобильного приложения знакомств охватывает все основные аспекты взаимодействия между пользователями и системой, обеспечивая удобство и гибкость для различных категорий пользователей (гостей и клиентов). Для гостей предусмотрены простые и интуитивно понятные процессы регистрации и авторизации, которые открывают доступ к расширенным функциям для пользователей и являются обязательными, таким как редактирование профиля, взаимодействие через свайп-систему и получение уведомлений, а так же страница отображения совпадений, которая является ключевой страницей в мобильном приложении для знакомств. Администраторы получают возможность эффективно управлять данными о пользователях и их предпочтениях, что обеспечивает актуальность информации и быстрое реагирование на измененя в потребностях пользователей. Каждая роль и функциональность тесно связаны между собой, что позволяет приложению знакомств оставаться гибким, масштабируемым и удобным для конечных пользователей.

# 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

На диаграмме состояния, представленной для информационной системы мобильного приложения для знакомств, изображен процесс взаимодействия пользователя с системой, представлено на рисунке 4.1.

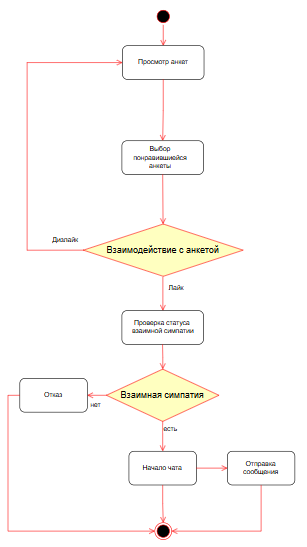


Рисунок 4.1 – Диаграмма состояния

Процесс начинается с того, что клиент просматривает список доступных анкет и выбирает одну из них. Затем он переходит к принятию решения о том как он будет взаимодействовать с анкетой. Если клиент выбирает поставить лайк, то дальше система проверяет существует ли взаимная симпатия. В ином случае клиент продолжает просмотр анкет. Если присутствует взаимная симпатия, то система начинает чат между клиентами которые имеют взаимную симпатию. В случае отсутствия взаимной симпатии клиент может как завершить просмотр анкет, так и продолжить. На последнем этапе клиенту предоставляется возможность отправки сообщения. Таким образом, диаграмма описывает основные шаги процесса использования приложения, включая выбор понравившийся анкеты, взаимодействие с анкетой, и начало общения в чате.

Данная схема процесса способствует упрощению взаимодействия клиента с системой и обеспечивает более эффективное управление процесса поиска партнера в приложении. Важно, что система предусматривает механизмы для обработки ошибок и обратного перехода на предыдущие шаги, что улучшает пользовательский опыт.

Далее рассмотрим диаграмму последовательности, представленной на рисунке 4.2.

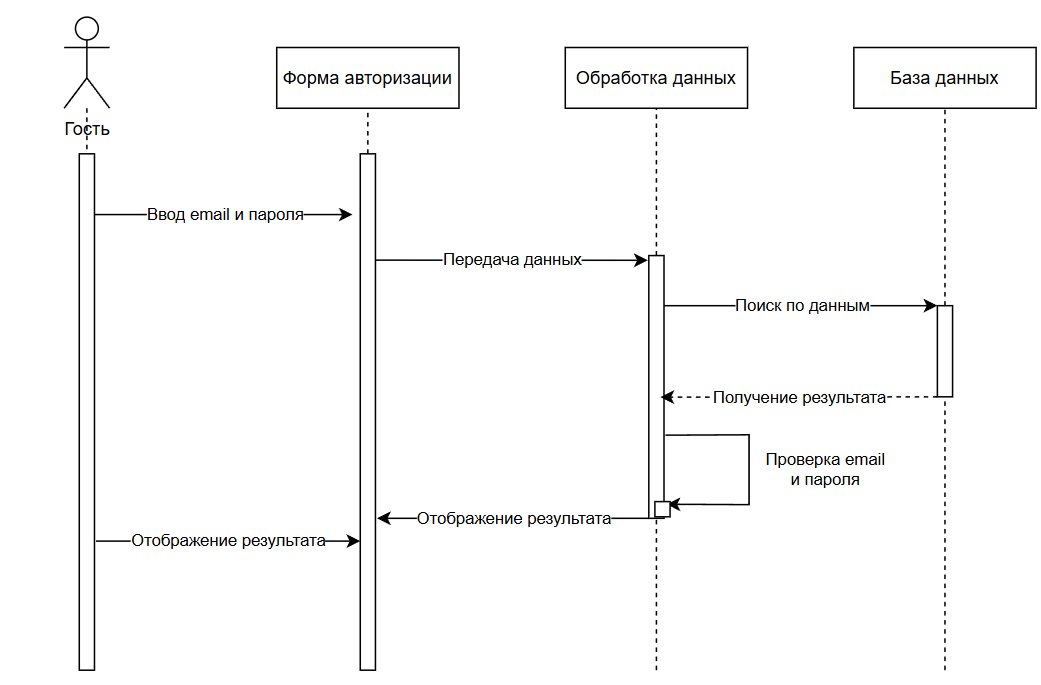


Рисунок 4.2 – Диаграмма последовательности

Диаграмма последовательности отображает процесс авторизации неавторизованного пользователя в системе. Пользователь вводит email и пароль в форму авторизации, после чего данные передаются на обработку. Система обработки данных отправляет запрос в базу данных, чтобы найти соответствие введенных данных с уже существующими записями. База данных выполняет поиск по введенным параметрам (email и паролю) и возвращает результат обратно в систему обработки данных. Затем система проверяет правильность email и пароля. Если данные корректны, пользователю передается сообщение о успешной авторизации. В случае ошибки (например, если email или пароль введены неправильно), пользователь получает уведомление о том, что авторизация не удалась. В итоге, диаграмма демонстрирует, как компоненты (пользовательский интерфейс, обработка данных и база данных) взаимодействуют в процессе авторизации пользователя.

# Заключение

В ходе выполнения лабораторной работы, посвященной разработке информационной системы для мобильного приложения для знакомств, были созданы две ключевые диаграммы: диаграмма последовательности и диаграмма деятельности. Эти диаграммы являются важными инструментами для визуализации процессов и взаимодействий внутри системы, обеспечивая более глубокое понимание её работы и улучшая процесс проектирования.

Диаграмма последовательности была разработана для отображения процесса авторизации пользователя в системе. Эта диаграмма иллюстрирует динамику взаимодействия различных объектов во времени, демонстрируя, как неавторизованный посетитель вводит свои данные, как система обрабатывает их и взаимодействует с базой данных для проверки логина и пароля. Диаграмма показывает, какие сообщения передаются между объектами и как происходит последовательность событий, начиная от ввода данных и до их проверки. Благодаря диаграмме последовательности стало возможным четко увидеть процесс обмена данными и важные моменты, где система может столкнуться с проблемами, такими как неверные данные авторизации.

Диаграмма состояния была создана для моделирования процесса поиска партрера. Она визуализирует различные состояния системы и переходы между ними в процессе записи пользователя. Диаграмма состояния наглядно показывает возможные пути, по которым пользователь может пройти в зависимости от действий или событий, таких как ввод некорректных данных. Она демонстрирует сценарии обработки ошибок и альтернативные исходы, такие как невозможность системы проверить все данные. Это помогает более глубоко понять бизнес-логику системы, а также выявить потенциальные проблемы или узкие места, которые могут привести к задержкам в процессе знакомства.

Сравнение обеих диаграмм позволяет лучше осознать, как взаимодействие объектов в системе (представленное в диаграмме последовательности) непосредственно влияет на более глобальные процессы (смоделированные в диаграмме деятельности).

Таким образом, работа с диаграммами последовательности и деятельности для мобильного приложения для знакомств подтвердила их высокую значимость в процессе проектирования информационных систем. Они позволяют не только структурировать информацию, но и выявлять проблемные зоны, которые могут требовать оптимизации. Использование этих диаграмм способствует более качественному проектированию системы, облегчает коммуникацию между разработчиками и бизнес-заказчиками, а также улучшает принятие решений на этапе реализации проекта.