Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 6. Объектно-ориентированное моделирование. Физические диаграммы UML

Студент: Пузиков А. А.

ФИТ 4 курс 6 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# Введение

Парадигма объектно-ориентированного программирования (ООП) является основой для создания современного программного обеспечения и находит широкое применение в разработке информационных систем. Модель объектов, основанная на данной парадигме, позволяет эффективно описывать свойства и функциональные возможности сущностей, взаимодействующих в рамках разрабатываемой системы. Эти объекты не существуют изолированно; они взаимодействуют друг с другом для достижения общей цели — успешного проведения интеллектуальных игр и обеспечения качественного взаимодействия пользователей.

Объектно-ориентированное моделирование (ООМ) является важной методологией разработки программного обеспечения, которая позволяет более эффективно и наглядно проектировать сложные информационные системы. ООМ основывается на принципах объектно-ориентированного программирования (ООП), где основными элементами являются объекты и классы, взаимодействующие между собой. Для представления архитектуры и взаимодействия между компонентами системы широко используется язык моделирования UML (Unified Modeling Language). UML предоставляет стандартизированные средства для описания различных аспектов системы, от логической структуры до физической реализации. Среди таких диаграмм особое значение занимают физические диаграммы — компоненты и развертывания, которые отражают архитектуру системы на более конкретном уровне.

В рамках разработки интеллектуальной платформы PSSIG, предназначенной для проведения игр с элементами социальной сети, необходимо создать структуру, которая обеспечит четкую связь между различными компонентами системы и их размещением на физических устройствах. Платформа PSSIG включает множество модулей, таких как регистрация пользователей, проведение игр, сбор статистики, социальные функции (добавление друзей, создание сообществ), и каждый из них требует точного описания на уровне компонентов. Кроме того, важным аспектом является распределение этих компонентов на физические узлы, такие как серверы, клиентские устройства и базы данных, что делает диаграмму развертывания ключевым элементом проектирования.

Цель данной работы — изучить основные принципы объектно-ориентированного моделирования с использованием UML и применить их для проектирования информационной системы PSSIG. В рамках работы будут созданы диаграммы компонентов и развертывания, которые отразят взаимодействие различных элементов системы и их распределение на физические устройства. Это позволит глубже понять архитектуру платформы и обеспечить её эффективное развитие в дальнейшем.

# Постановка задачи

Система PSSIG (Platform for Social and Intellectual Games) представляет собой платформу для организации интеллектуальных игр с элементами социальной сети. Основная цель проекта — предоставить пользователям возможность участвовать в интеллектуальных играх, собирать и анализировать результаты, обмениваться опытом и взаимодействовать друг с другом. PSSIG объединяет функции социальной сети и игрового сервиса, предоставляя гибкие инструменты для регистрации, создания и участия в играх, а также для управления статистикой и игровыми сообществами. Система поддерживает многопользовательский режим и обеспечивает динамическое взаимодействие между участниками, организаторами и администраторами игр.

Архитектурные и технические требования к системе PSSIG подразумевают использование современных технологий и инструментов для обеспечения стабильной работы и безопасности данных. Клиентская часть платформы разрабатывается на базе React с использованием библиотеки компонентов Shadcn UI для создания удобного и адаптивного интерфейса. Серверная часть реализована на ASP.NET Core Web API, что позволяет создавать высокопроизводительные API и легко интегрировать с клиентской частью. В качестве базы данных используется PostgreSQL, а для повышения производительности и масштабируемости системы применяется Redis для кеширования данных. Система должна обеспечивать безопасное хранение и передачу данных через HTTPS, а также поддерживать многопоточную обработку запросов. Важным элементом архитектуры является API Gateway для маршрутизации запросов и балансировки нагрузки, что позволяет распределить трафик между серверными компонентами.

Основные компоненты платформы PSSIG включают:

* Компонент управления пользователями: регистрация, аутентификация, авторизация, профили пользователей.
* Компонент управления играми: создание, настройка, запуск игр, хранение вопросов и ответов.
* Компонент социальной сети: добавление в друзья, создание сообществ, обмен сообщениями.
* Компонент сбора статистики: обработка и хранение результатов игр, формирование отчетов.
* Компонент взаимодействия с Redis: кэширование данных для повышения производительности.
* Компонент администрирования: управление пользователями, играми, настройками платформы.

Эти компоненты взаимодействуют между собой через REST API, обеспечивая интеграцию различных функциональных модулей и высокую гибкость системы.

# Теоретическая часть

Физические диаграммы, такие как диаграммы компонентов и диаграммы развертывания, играют ключевую роль в процессе проектирования информационных систем. Они помогают визуализировать архитектуру системы на уровне её реализации и физического размещения. Диаграмма компонентов (Component Diagram) предназначена для отображения логических частей системы, таких как модули, библиотеки или сервисы, и их взаимодействия. Эта диаграмма показывает, как различные компоненты связаны друг с другом и какие интерфейсы они предоставляют. Она используется для детального проектирования структуры приложения и упрощает дальнейшее развитие и поддержку кода. Диаграмма развертывания (Deployment Diagram), в свою очередь, фокусируется на физическом размещении компонентов системы на конкретных устройствах (узлах), таких как серверы или клиентские устройства, и описывает, как эти узлы взаимодействуют между собой через сети или другие соединения.

Узел (Node) в UML представляет собой физическое устройство или ресурс, на котором выполняется компонент или группа компонентов. Узлы могут быть серверами, клиентскими устройствами, базами данных или облачными сервисами. Артефакт (Artifact) — это результат процесса разработки, который может быть развернут на узле. Это могут быть исполняемые файлы, базы данных или конфигурационные файлы. Интерфейс (Interface) — это набор операций или методов, которые компонент предоставляет другим компонентам для взаимодействия. Интерфейсы позволяют абстрагироваться от внутренней реализации компонента и сосредоточиться на его внешнем поведении.

Для диаграммы компонентов используются следующие нотации: компонент изображается в виде прямоугольника с двумя маленькими прямоугольниками в верхнем правом углу, указывающими на его сложную природу. Взаимодействие между компонентами отображается с помощью линий, представляющих зависимости, либо с использованием символов интерфейсов.

Диаграмма развертывания содержит узлы, которые изображаются в виде трехмерных кубов. Узлы могут быть связаны между собой линиями, которые символизируют физические или сетевые соединения. Компоненты, размещенные на узлах, отображаются внутри этих узлов, что показывает, на каких устройствах выполняется каждая часть системы. Связи между компонентами также могут включать указание протоколов связи, например, HTTP или HTTPS.

Таким образом, физические диаграммы UML позволяют не только спроектировать структуру системы, но и четко понять, как компоненты будут взаимодействовать на уровне физической реализации.

# Описание программных средств

Для создания UML-диаграмм в рамках проектирования системы PSSIG использовались несколько профессиональных инструментов моделирования. Одним из наиболее популярных решений для разработки диаграмм является StarUML. StarUML — это мощное средство моделирования, поддерживающее создание диаграмм UML различных типов, включая диаграммы компонентов и развертывания. Программа разработана командой MKLab и активно используется для создания моделей систем, основанных на принципах объектно-ориентированного проектирования. Версия, использованная для выполнения данной работы, — StarUML 5.0, которая предлагает широкий спектр возможностей, включая генерацию исходного кода на различных языках программирования и поддержку интеграции с другими средствами разработки.

StarUML поддерживает множество UML-диаграмм, включая диаграммы классов, компонентов, развертывания, объектов и последовательностей. Одной из ключевых возможностей является гибкость настройки панелей инструментов и возможность добавления пользовательских шаблонов и библиотек. Панель инструментов включает такие важные элементы, как узлы (Nodes), компоненты (Components), связи (Dependencies), интерфейсы (Interfaces), артефакты (Artifacts), что позволяет моделировать сложные системы на различных уровнях. В StarUML также поддерживается расширение функционала за счет плагинов, что позволяет адаптировать его под специфические требования проекта.

Кроме StarUML, для работы с UML-диаграммами может быть использована и программа Visual Paradigm. Этот инструмент разработан компанией Visual Paradigm International и доступен в версиях как для образовательных, так и коммерческих целей. Версия Visual Paradigm 17.0, использованная в некоторых частях проекта, предоставляет улучшенные функции для совместной работы, интеграцию с популярными системами контроля версий и поддерживает создание не только UML, но и BPMN-диаграмм. Одним из преимуществ Visual Paradigm является поддержка облачного хранилища и возможности реального времени совместного редактирования моделей.

Visual Paradigm предлагает широкий набор инструментов для работы с UML, таких как диаграммы компонентов и развертывания, и предоставляет удобный интерфейс для работы с различными версиями диаграмм. В программе доступна панель инструментов с множеством UML-элементов, позволяющая легко добавлять компоненты, связи и интерфейсы на диаграмму.

Таким образом, использование этих программных средств обеспечивает качественное и эффективное моделирование архитектуры информационной системы PSSIG, позволяя легко управлять диаграммами и поддерживать интеграцию с другими инструментами разработки.

# Практическая часть

Диаграмма развертывания описывает физическое размещение компонентов на устройствах и протоколы связи между ними.

Диаграмма развертывания представлена на рисунке 4.2.

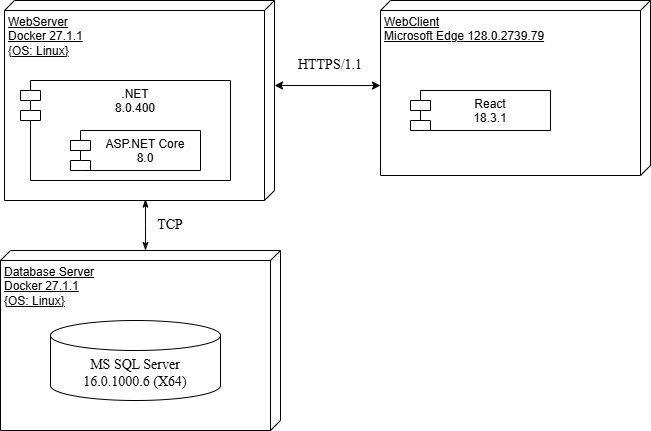


Рисунок 4.2 – Диаграмма развертывания

Основными устройствами системы являются:

* Веб-сервер (на базе IIS), где размещаются компоненты серверной части. Этот сервер обрабатывает запросы пользователей и организаторов, управляет данными игр и пользователей.
* База данных MS SQL — отдельный узел, на котором хранится вся информация о пользователях, играх, результатах и взаимодействиях. База данных подключена к серверу через протокол TCP/IP.
* Клиентские устройства — пользователи взаимодействуют с системой через браузеры на компьютерах или мобильных устройствах, отправляя запросы на веб-сервер через протокол HTTPS.

Таким образом, компоненты системы распределены по различным узлам, обеспечивая масштабируемость и высокую производительность.

# Ответы на вопросы

1) Укажите назначение физических диаграмм: компонентов и развертывания.

Физические диаграммы, предназначены для отображения структуры системы и ее архитектуры на высоком уровне. Диаграмма компонентов описывает взаимодействие и связи между различными модулями системы, что позволяет разработчикам понимать, как компоненты работают вместе и какие технологии используются для их реализации. Диаграмма развертывания показывает, как компоненты системы распределены по физическим узлам, включая серверы, устройства и протоколы связи, что помогает в планировании инфраструктуры и оценке масштабируемости.

2) Дайте определение понятиям: узел, артефакт, интерфейс.

* Узел — это физический объект, на котором размещаются компоненты системы, например, сервер или клиентское устройство. Узел может содержать одно или несколько аппаратных средств и предоставляет среду для выполнения приложений.
* Артефакт — это результат разработки, который может включать исполняемые файлы, документы, модели или другие выходные данные, связанные с компонентами системы.
* Интерфейс — это точка взаимодействия между компонентами или узлами, определяющая, как они обмениваются данными.

3) Опишите нотации, которые используются для представления компонентов (их вариации).

Для представления компонентов в диаграммах используются различные нотации, включая прямоугольники, которые символизируют компоненты, и линии, которые обозначают связи между ними. Кроме того, могут использоваться стереотипы, такие как <<component>>, для обозначения специфики элемента. Также возможно использование иконок для отображения различных типов компонентов, таких как базы данных, веб-сервисы или интерфейсы.

4) Опишите основные нотации, которые используются для представления архитектуры системы в виде диаграммы развертывания.

На диаграммах развертывания используются нотации, такие как узлы, обозначенные прямоугольниками, и артефакты, изображаемые в виде файлов или документов.

5) Укажите основные виды связей между компонентами и между узлами.  
Существуют различные виды связей между компонентами, такие как ассоциации, зависимости и агрегации. Ассоциация указывает на связь между компонентами, зависимость — на то, что один компонент требует другой для своей работы, а агрегация — на более сложные отношения, где один компонент может включать в себя другие. Между узлами также могут быть связи, которые определяют, как данные передаются между различными физическими устройствами, включая протоколы связи и типы передачи данных (например, синхронный или асинхронный обмен).

# Заключение

В ходе выполнения данной работы были разработаны ключевые физические диаграммы для системы PSSIG, включая диаграмму компонентов и диаграмму развертывания. Эти диаграммы сыграли важную роль в формировании четкого представления о структуре системы и её архитектурных решениях. Диаграмма компонентов показала, как различные модули системы взаимосвязаны и взаимодействуют друг с другом, что способствует пониманию процессов, связанных с регистрацией пользователей, управлением играми и сбором статистики. Каждому компоненту была присвоена определенная технология реализации, что позволяет гарантировать соответствие современным стандартам разработки.

Архитектурные решения, принятые при проектировании системы PSSIG, показали свою применимость и правильность. Использование микросервисной архитектуры для разделения функций между различными компонентами позволяет достичь гибкости и масштабируемости системы. Каждый модуль может быть разрабатываем и развёртываем независимо, что упрощает процесс поддержания и обновления системы. Применение таких технологий, как **ASP.NET Core**, **Entity Framework Core**, **PostgreSQL** и **Redis**, обеспечивает надежную и эффективную обработку данных, а также высокий уровень безопасности и производительности.

Далее, диаграмма развертывания продемонстрировала физическое размещение компонентов на узлах системы, что подтверждает правильный выбор архитектурных решений для обеспечения взаимодействия между клиентскими устройствами и серверной частью. Протоколы связи, такие как **HTTP(S)** и **TCP/IP**, обеспечивают надежное взаимодействие между компонентами системы, что является критически важным для функционирования PSSIG.

В заключение, выполненная работа продемонстрировала, что применение методологии UML для проектирования архитектуры информационной системы PSSIG способствует созданию надежной и гибкой платформы для организации интеллектуальных игр и взаимодействия между пользователями. Это, в свою очередь, открывает возможности для дальнейшего расширения функциональности и улучшения качества обслуживания пользователей.