Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование программного обеспечения»

Отчёт по лабораторной работе №6

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ.

ФИЗИЧЕСКИЕ ДИАГРАММЫ UML.

Цель: Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования структуры информационной системы с применением UML

Выполнил: Коренчук А. В.

ФИТ 3 курс 2 группа

Преподаватель: Якунович А. В.

1. **Постановка задачи**

В данной лабораторной работе необходимо составить диаграмму развёртывания и компонентов, которая должна содержать: список компонент, из которых будет состоять подсистема, существующие связи между компонентами подсистемы, список физических устройств, на которых будет работать подсистема и определить соединения.

1. **Описание программных средств**

Построение моделей выполнялось в программной среде Draw.io. Draw.io — инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.

1. **Практическое задание**

В соответствии с условием была спроектирована следующая диаграмма – рисунок 1.

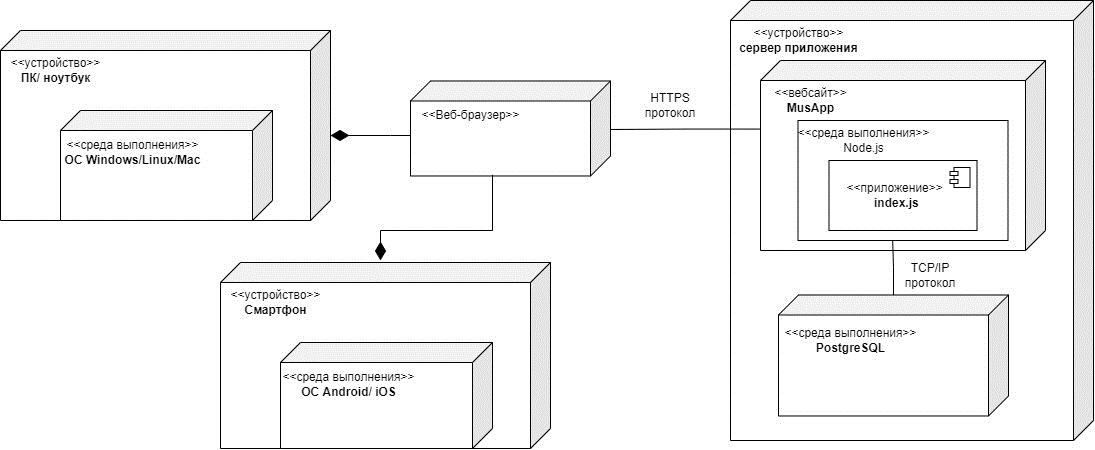


Рисунок 1 – Диаграмма развёртывания и компонентов

Список компонентов подсистемы:

**Список компонентов подсистемы:**

1. Компонент «ПК/ноутбук»:
   * Технология реализации: Общепринятые технологии для ПК/ноутбуков.
2. Компонент «Операционная система (Windows / Linux /Mac OS/ Android/ iOS)»:
   * Технология реализации: Microsoft Windows, Apple Mac OS, Linux, Android, Apple iOS.
3. Компонент «Приложение »:
   * Технология реализации: JS.
4. Компонент «Среда выполнения Node»:
   * Технология реализации: Node.
5. Компонент «PostgreSQL»:
   * Технология реализации: PostgreSQL.

**Список физических устройств:**

1. Устройство «ПК/ ноутбук»:

Основные характеристики: ПК или ноутбук, поддерживающий работу с веб-приложениями.

1. Устройство «Смартфон»:

Основные характеристики: Мобильное устройство, способное взаимодействовать с веб-приложением.

1. Устройство «Сервер приложения»:

Основные характеристики: Физический сервер, способный запускать и обслуживать базу данных PostgreSQL, а также содержащий экземпляр приложения.

**Соединения - протоколы связи между устройствами:**

1. Соединение между «ПК/ноутбуком» или «Смартфоном» и «Сервером приложения»: Протокол связи: HTTPS.
2. Соединение между «Сервером приложения» и «Средой выполнения Базы данных»: Протокол связи: TCP/IP.

**Среды выполнения:**

1. Среда выполнения «Node.js»: Node.js представляет собой среду выполнения для веб-приложений, построенных на движке V8 и ориентированных на выполнение JavaScript на стороне сервера.

Размещение компонентов: «Приложения» размещается в среде выполнения «ASP.NET Core».

1. Среда выполнения «Операционная система (Windows/ Linux/ MacOs/ Android/ iOS)»: Данные ОС являются средой выполнения для приложений, созданных с помощью кроссплатформенной технологии Node.

Размещение компонентов: «Операционная система» является обязательной частью «Устройства».

1. Среда выполнения «PostgreSQL»: Данная среда выполнения содержит в себе экземпляр базы данных PostgreSQL.

Размещение компонентов: «PostgreSQL» размещается в среде выполнения «Сервер приложения».

**Размещение компонентов на устройствах:**

1. Серверное устройство:
   * Все необходимые среды выполнения и компоненты расположены исключительно на серверном устройстве.

**Вывод**: В результате данной лабораторной работы были получены навыки проектирования диаграммы развёртывания для музыкальной площадки, получен теоретические знания о проектировании программного обеспечения в рамках методологий объектно-ориентированного моделирования средствами UML.

**Ответы на теоретические вопросы**

1. **Укажите назначение физических диаграмм: компонентов и развертывания**

Диаграмма компонентов позволяет определить состав программных компонентов в роли которых может выступать исходный, бинарный и исполняемый код.

Диаграмма развёртывания же позволяет определить распределение компонентов системы по её физическим узлам; показать физические связи между всеми узлами реализации системы на этапе ее выполнения; выявить узкие места системы и реконфигурировать её топологию для достижения требуемой производительности.

1. **Дайте определение понятиям: узел, артефакт, интерфейс**

Артефакт – некая физическая сущность, программный компонент, который используется или создаётся во время работы программного обеспечения.

Узел – то, что может содержать программное обеспечение (некоторый физически существующий элемент системы, обладающий вычислительным ресурсом).

Интерфейс – элемент, который служит для спецификации параметров модели, которые видимы извне без указания их внутренней структуры.

1. **Опишите нотации, которые используются для представления компонентов**

Для представления компонентов используются две нотации – рисунок 2, отличаются они визуально.

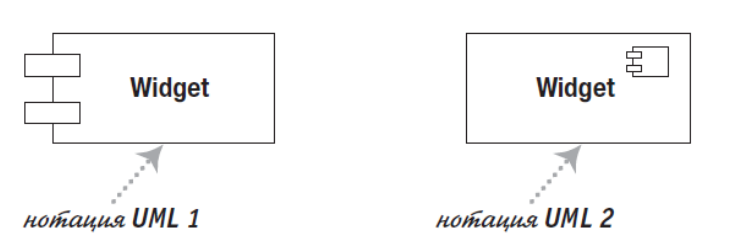


Рисунок 2 – Нотации для представления компонентов

1. **Опишите основные нотации, которые используются для представления архитектуры системы в виде диаграммы развертывания**

Узлы (Nodes): Представляют физические устройства или вычислительные ресурсы, такие как серверы, ПК, мобильные устройства и т.д. Узлы изображаются в виде прямоугольников с именем устройства.

Артефакты (Artifacts): Представляют программные компоненты или данные, которые размещены на узлах. Они изображаются в виде прямоугольников или эллипсов внутри узлов.

Связи (Connections): Показывают взаимодействие между узлами. Они могут быть направленными или двунаправленными линиями, обозначающими поток данных, коммуникации или зависимостей между узлами.

1. **Укажите основные виды связей между компонентами и между узлами**

Сетевые связи (Network Links): Показывают физические сетевые соединения между узлами. Это могут быть Ethernet, Wi-Fi, VPN и другие типы сетевых соединений.

Коммуникационные связи (Communication Links): Представляют логические связи между компонентами системы. Это могут быть протоколы обмена данными, такие как HTTP, TCP/IP, MQTT и другие, которые описывают способы обмена информацией между компонентами.

Зависимости (Dependencies): Показывают, какие компоненты или узлы зависят от других. Например, если один компонент зависит от базы данных, это может быть представлено стрелкой или связью, указывающей на эту зависимость.

Репликация и балансировка нагрузки (Replication and Load Balancing): Если компоненты или данные реплицируются на несколько узлов или происходит балансировка нагрузки между узлами для повышения производительности и надежности, это может быть показано на диаграмме развертывания.

Прокси и шлюзы (Proxies and Gateways): Представляют промежуточные узлы, используемые для обеспечения безопасности, переадресации или трансляции данных между компонентами.