Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Проектирование программного обеспечения»

Лабораторная работа №6

«Проектирование информационных систем»

Студент: Лопатнюк П.В.

ФИТ 3 курс 1 группа

Преподаватель: Якубенко К. Д

Минск 2024

# Теоретические сведения

Физические диаграммы, такие как диаграммы компонентов и развертывания, играют ключевую роль в проектировании и документировании сложных систем. Диаграммы компонентов предназначены для отображения модульной структуры системы: какие программные модули существуют, как они взаимодействуют, и каков их интерфейс. Диаграммы развертывания, в свою очередь, показывают физическое размещение программных компонентов на аппаратных узлах и взаимодействие между этими узлами. Они полезны для описания инфраструктуры системы, что особенно важно для настройки серверов, сети и других ресурсов.

Узел — это физическое или виртуальное устройство, на котором размещаются компоненты системы. Например, сервер, клиентский компьютер или облачный сервис.

Артефакт — это конечный продукт разработки, такой как исполняемый файл, библиотека, скрипт или конфигурационный файл, который развёртывается на узле.

Интерфейс — это точка взаимодействия между компонентами, определяющая способ их связи. Интерфейс может быть представлен в виде API, протокола или контракта.

Для представления компонентов в UML используются различные графические элементы: прямоугольники, разделённые на три секции: название компонента, интерфейсы и зависимости.

Основные элементы диаграммы развертывания включают:

* Узлы (прямоугольники с названием устройства или сервиса). Узлы могут быть вложенными для отображения среды выполнения (например, ОС или виртуальной машины).
* Компоненты (расположенные внутри узлов), показывающие, что развёрнуто на каждом узле.
* Соединения между узлами, обозначающие виды протоколов (HTTP, TCP/IP, SOAP).
* Депенденции и артефакты, отображающие логические зависимости между узлами и данными.

Связи между компонентами: депенденции (показывают, что один компонент зависит от другого), ассоциации (обозначают прямое взаимодействие между компонентами), реализация интерфейса (показывается стрелкой от компонента к интерфейсу).

Связи между узлами: физическое соединение, логическое соединение, иерархическая связь: узел может содержать другой узел (например, сервер содержит виртуальные машины).

# Постановка задачи

Целью данной работы является построение диаграммы компонентов и диаграммы развертывания для интерфейса электронной почты. Это предполагает создание точной и наглядной модели, которая позволит лучше понять структуру системы, взаимодействие её компонентов, функции, которые они выполняют, предоставляемые и требуемые интерфейсы, а также физическое развертывание компонентов в целевой среде.

Диаграмма компонентов является ключевым элементом UML, который позволяет:

* Отразить физическую структуру приложения: определить основные модули (например, клиентское приложение, сервер, база данных, библиотека и т.д.).
* Показать взаимодействие компонентов: какие модули связаны между собой и каким образом осуществляется их взаимодействие (например, через API, зависимости, интерфейсы).
* Указать предоставляемые и используемые интерфейсы, что важно для проектирования взаимосвязанных систем.
* Продемонстрировать зависимости: какие компоненты зависят от других и каким образом.

Диаграмма развертывания представляет физическую инфраструктуру и отображает:

* Узлы: серверы, клиентские устройства или виртуальные машины, которые обеспечивают выполнение компонентов.
* Размещение компонентов: показывает, где физически или виртуально располагаются программные элементы системы.
* Связи между узлами: логические и физические соединения, например, через протоколы TCP/IP, SOAP или HTTP.
* Взаимодействие узлов в сети и их роль в архитектуре приложения.

Задача работы заключается в построении обеих диаграмм для интерфейса электронной почты, включая:

* Определение основных компонентов системы (например, клиент, сервер, база данных, DNS-сервер).
* Выявление интерфейсов и их зависимостей.
* Отображение физического размещения компонентов в узлах.
* Визуализацию сетевых взаимодействий между узлами.

Результатом станет подробная визуализация физической архитектуры интерфейса электронной почты, которая облегчит проектирование, реализацию и последующее развертывание системы, а также поможет в масштабировании и улучшении взаимодействия её элементов.

# Описание программных средств

Visual Paradigm — это многофункциональное программное обеспечение для моделирования и проектирования, которое широко используется как в IT, так и в бизнес-средах. Программа позволяет создавать различные типы диаграмм, такие как UML (Unified Modeling Language), BPMN (Business Process Model and Notation), DFD (Data Flow Diagram), ERD (Entity Relationship Diagram) и многие другие. Она предназначена для визуализации сложных процессов, архитектуры систем и бизнес-структур, что делает её полезной для аналитиков, разработчиков и менеджеров проектов.  
 Программа разработана компанией Visual Paradigm International и доступна как в виде десктопного приложения для платформ Windows, macOS и Linux, так и онлайн через веб-интерфейс. Это дает возможность пользователям работать с диаграммами как на локальных компьютерах, так и в облаке, что упрощает совместную работу над проектами. Visual Paradigm поддерживает функционал для создания моделей в режиме реального времени, что особенно полезно для командного взаимодействия и управления проектами.  
 Одной из ключевых особенностей Visual Paradigm является её интеграция с популярными средами разработки, такими как Eclipse, NetBeans и IntelliJ IDEA, что позволяет seamlessly переносить диаграммы в код или управлять проектами на разных этапах жизненного цикла разработки. Также программа поддерживает экспорт диаграмм в популярные форматы, включая PNG, PDF, SVG, и легко интегрируется с инструментами для версионирования, такими как Git.  
 Visual Paradigm предлагает гибкий подход к управлению процессами, позволяющий вам адаптировать его под свои индивидуальные нужды. Наш принцип Just-in-Time дает вам свободу выбора: вы можете использовать готовые процессы или создавать собственные, используя богатую библиотеку компонентов.

Visual Paradigm предоставляет гибкий подход к моделированию, который подходит для использования как в малых командах, так и в крупных предприятиях.

# Практическое задание

Диаграмма развертывания электронной почты представляет собой физическую архитектуру системы, которая включает в себя несколько ключевых компонентов и их взаимосвязь. Подсистема состоит из клиента электронной почты, сервера электронной почты, базы данных, DNS-сервера и внешних серверов, обеспечивающих взаимодействие с другими почтовыми системами.

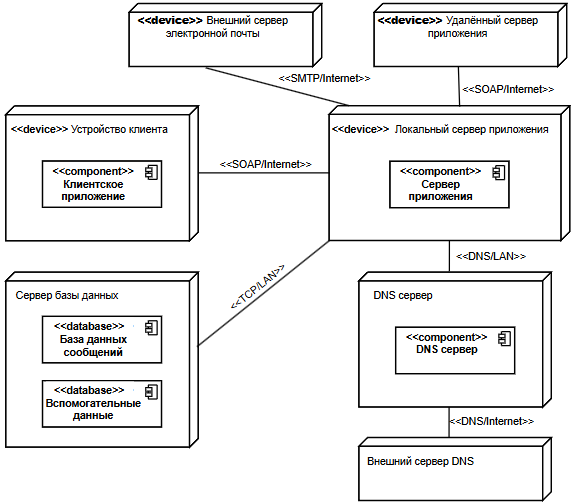


Рисунок 4.1 – Диаграмма развёртывания

Клиент электронной почты выполняет функции отправки, получения и управления сообщениями. Взаимодействие клиента с сервером осуществляется через RESTful API, работающий по протоколу HTTP или HTTPS, что обеспечивает защищённую передачу данных.

Сервер электронной почты, написанный на Node.js с использованием Express.js, является центральным узлом системы. Он обрабатывает все запросы клиента, управляет передачей и хранением сообщений, а также обеспечивает взаимодействие с внешними почтовыми системами через протокол SMTP. Сервер также подключается к базе данных, где хранятся данные пользователей, электронные письма, а также логи и конфигурации системы. Связь между сервером и базой данных осуществляется по протоколу TCP/IP, что гарантирует надёжную передачу данных.

База данных, развернутая на сервере с установленным PostgreSQL, служит хранилищем данных системы. Она содержит информацию о пользователях, их сообщениях, а также настройках и логах системы. Для обеспечения высокой производительности база данных размещена на мощном сервере с современными характеристиками, включающими многопроцессорную архитектуру и высокоскоростные SSD-накопители.

DNS-сервер, настроенный с использованием BIND9, играет важную роль в маршрутизации почтовых сообщений. Он отвечает за разрешение доменных имен в IP-адреса, а также за обработку MX-записей, необходимых для работы почтовой системы. Связь между сервером электронной почты и DNS-сервером осуществляется через DNS-запросы по протоколам UDP или TCP.

Система также взаимодействует с внешними почтовыми и DNS-серверами. Это необходимо для отправки сообщений на другие почтовые системы и получения входящих сообщений. Такое взаимодействие реализуется с использованием стандартных протоколов, таких как SMTP для передачи сообщений и DNS для разрешения адресов.

Физическая инфраструктура системы включает клиентские устройства, серверы и сетевые соединения. Серверы системы, включая сервер электронной почты, сервер базы данных и DNS-сервер, размещены на мощных физических или виртуальных машинах с высокими техническими характеристиками, обеспечивающими надёжность и производительность. Для передачи данных между устройствами используются защищённые протоколы, такие как HTTPS и SMTP.

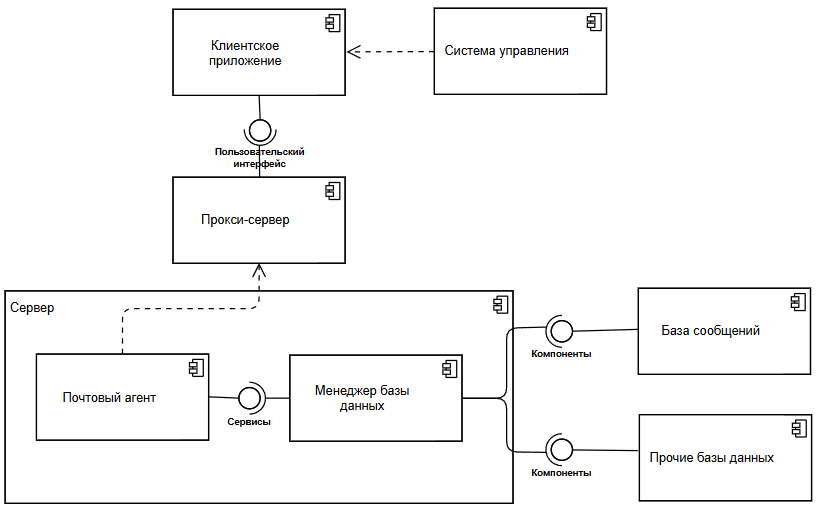


Рисунок 4.2 – Диаграмма компонентов

Диаграмма компонентов отображает физическую архитектуру системы электронной почты, включая основные компоненты, их взаимодействие и размещение.

Основные компоненты:

Клиент электронной почты (WSEmail Client):

* Интерфейс пользователя (GUI): обеспечивает взаимодействие с пользователем для отправки и получения писем.
* Получение писем (Retrieve Messages): загружает входящие сообщения.
* Создание писем (Compose Message): позволяет создавать и отправлять письма.
* Обновление плагинов (Automatic Plugin Updater): отвечает за обновление расширений системы.

Прокси-сервер (WSEmail Proxy):

* Посредник между клиентом и сервером, обрабатывающий запросы на отправку и получение сообщений.

Сервер электронной почты (WSEmail Server):

* Менеджер базы данных (Database Manager): управляет доступом к базе данных сообщений.
* Локальный почтовый агент (Local MTA): отвечает за отправку и получение писем.
* Очередь сообщений (Message Queue): управляет потоками сообщений для обработки.
* DNS-резолвер (DNS Resolver): выполняет разрешение доменных имен.

База данных сообщений (Messages Database):

* Хранит письма, данные пользователей и служебную информацию.

DNS-сервер (DNS Server):

* Обеспечивает маршрутизацию сообщений по доменным именам.

Интерфейс пользователя взаимодействует с прокси-сервером для отправки запросов на сервер.

Прокси-сервер отправляет данные на сервер электронной почты, который взаимодействует с базой данных.

Очередь сообщений обрабатывает входящие и исходящие сообщения, а также передает их внешним серверам.

DNS-резолвер и DNS-сервер обеспечивают маршрутизацию писем между узлами.

Основные протоколы связи:

* HTTP/HTTPS — для связи между клиентом и прокси.
* SMTP/IMAP — для отправки и получения сообщений.
* TCP/IP — для взаимодействия с базой данных.
* DNS-запросы — для маршрутизации сообщений.

Клиент работает в браузере или мобильном приложении. Сервер функционирует на платформе Linux, с использованием Node.js и базы данных PostgreSQL. Такая архитектура обеспечивает надежность и масштабируемость системы.