Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Проектирование программного обеспечения»

Тема: Интерфейс бронирования номеров

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

Студент: Жук С.С.

ФИТ 3 курс 2 группа

Преподаватель: Курилец А.В.

# **Описание функциональных требований**

Функциональны требования к системе можно разделить на требования к функционалу для различных ролей приложения – пользователя, гостя и администратора.

Функционал для пользователя:

* регистрация;
* авторизация;
* поиск и фильтрация по критериям;
* бронирование номера в отеле;
* добавление отзыва и рейтинга на отель;
* просмотр всей истории бронирования;
* просмотр и обновление личных данных;
* отмена бронирования номера.
* просмотр списка отелей и номеров.

Функционал для администратора:

* авторизация;
* поиск и фильтрация по критериям;
* отправка уведомлений о бронировании;
* управление пользователями (добавление, удаление, редактирование, блокировка пользователей);
* управление объявлениями;
* проверка объявлений и отзывов.
* просмотр списка отелей и номеров.

Функционал для гостя:

* регистрация;
* авторизация;
* просмотр списка отелей и номеров.
* поиск и фильтрация по критериям;

# **2. Описание программных средств**

Для построения диаграммы вариантов использования применен веб-ресурс Draw.io, разрабатываемый компанией JGraph Ltd и направленный на построение диаграмм. Адрес веб-ресурса – <https://www.drawio.com>. Данный ресурс доступен на всех платформах, имеющих веб-браузер и доступ в Интернет.

В функционал веб-ресурса Draw.io входит построение графиков, смысловых карт, UML-диаграмм, диаграммы Венна, Agile и Kanban досок и многое другое. Он также поддерживает импорт и экспорт файлов в различных форматах, включая PNG, JPEG, PDF и SVG.

# **3. Описание практического задания**

В соответствии с вышеописанными функциональными требованиями к программному средству была построена диаграмма развертывания.

Диаграмма развертывания для интерфейса бронирования номеров «HotelBooking» представлена на рисунке 1.1.

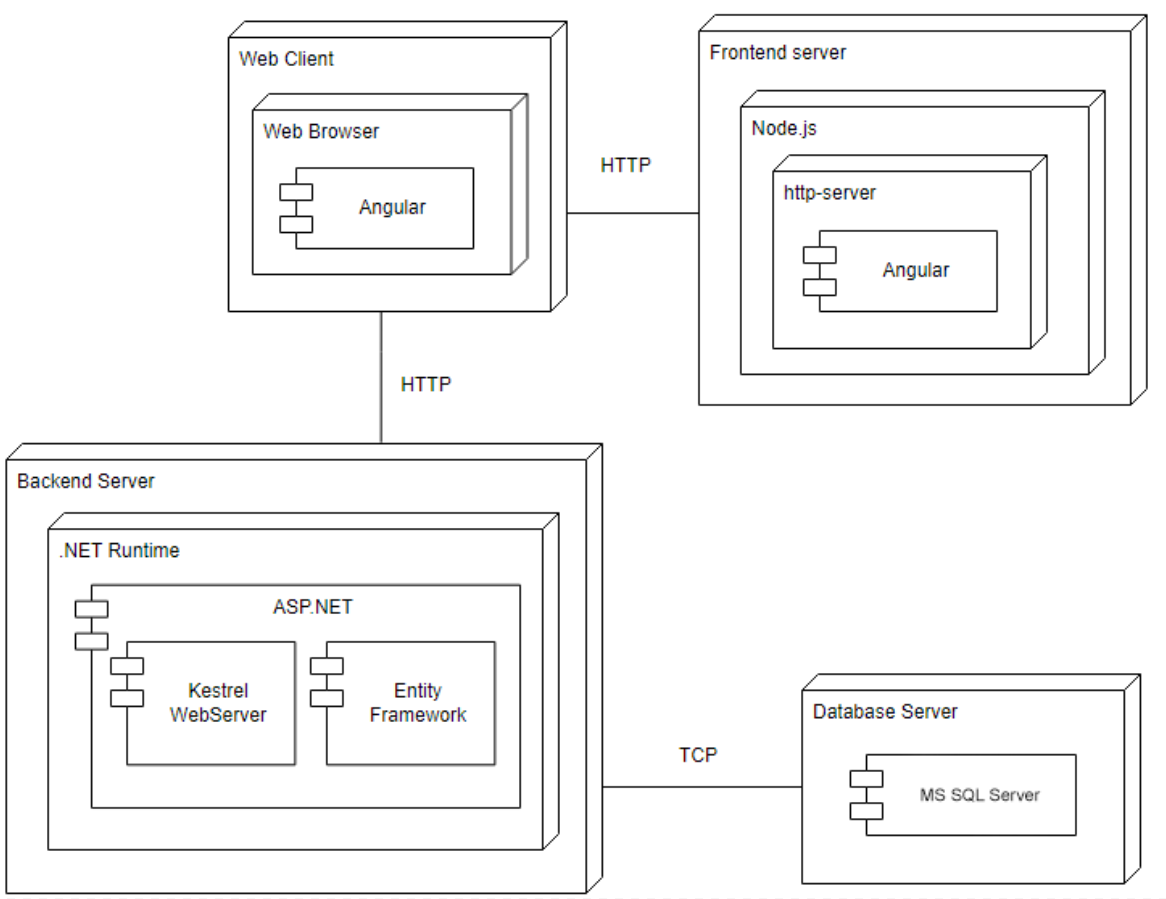


Рисунок 1.1 – Диаграмма развертывания

Как видно из данной диаграммы, система состоит из четырех узлов: Database Server, Backend Server, Web Client и Frontend Server.

Внутри узла Database Server находится компонент MS SQL Server – база данных, отвечающая за хранение данных в системе.

Узел Backend Server содержит в себе узел .NET Runtime – среду выполнения .NET, внутри которой выполняется компонент ASP.NET, содержащий в себе компоненты Kestrel Web Serber – веб-сервер для запуска приложений .NET и Entity Framework – ORM для обеспечения связи между .NET приложением и базой данных – узлом Database Server.

Узел Web Client содержит в себе узел Web Browser, и внутри его компонент Angular – фреймворк для создания клиентской части приложения, которая отображается с помощью узла Web Browser.

Узел Frontend Server содержит в себе узел среды выполнения для фронтенд-фреймвока Node.js, и внутри узел http-server и компонент Angular – сама серверная составляющая фронтеда.

Узлы Database Server и Backend Server связаны через соединение по протоколу TCP.Узлы Backend Server и Web Client связаны соединением по протоколу HTTP. Узлы Web Client и Frontend Server связаны соединением по протоколу HTTP.

**Теоретические вопросы:**

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

**Диаграммы компонентов**: Эти диаграммы представляют собой абстракцию, описывающую структуру системы в виде ее компонентов и взаимосвязей между ними. Они показывают, какие компоненты составляют систему, и как они взаимодействуют друг с другом. Основное назначение — обеспечить понимание модульности системы и упростить управление компонентами.

**Диаграммы развертывания**: Эти диаграммы описывают физическую архитектуру системы, показывая, как программные компоненты распределяются по физическим узлам (серверы, устройства и т. д.). Они помогают визуализировать, как компоненты развертываются и взаимодействуют в реальной среде, а также учитывать такие факторы, как производительность и доступность.

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

**Узел**: Узел представляет собой физическое устройство или программную среду, на которой выполняются компоненты системы. Узлы могут быть серверами, компьютерами, сетевыми устройствами и другими физическими или виртуальными машинами.

**Артефакт**: Артефакт — это конкретный выходной продукт процесса разработки, который может быть физическим или логическим. В контексте системной архитектуры это может быть исполняемый файл, библиотека, документ или любой другой элемент, который создается в процессе разработки.

**Интерфейс**: Интерфейс — это точка взаимодействия между компонентами или узлами, которая определяет, как они будут обмениваться данными или управлять друг другом. Интерфейсы могут быть определены как набор методов, атрибутов или операций, которые доступны для использования другими компонентами.

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

|  |  |
| --- | --- |
| **Тип графического элемента** | **Нотация** |
| 1.Компонент с текстовым стереотипом |  |
| 2.Компонент с пиктограммой стереотипа |  |
| 3.Компонент с предоставляемым интерфейсом |  |
| 4.Компонент имеет порт с предоставляемым интерфейсом |  |
| 5.Компонент с требуемым интерфейсом |  |
| 6.Компонент имеет порт с требуемым интерфейсом |  |
| 7.Компонент имеет несколько портов с предоставляемыми и требуемыми интерфейсами |  |

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

Основные нотации для диаграмм развертывания:

* **Узлы**: Узлы представляются в виде трехмерных блоков или прямоугольников, которые могут содержать другие узлы или артефакты.
* **Артефакты**: Артефакты изображаются как документы или файлы, которые могут быть связаны с узлами. Они обычно обозначаются прямоугольниками с различными значками, чтобы показать их тип.
* **Связи**: Связи между узлами показываются с помощью линий, которые могут обозначать разные типы соединений (например, физические, логические или сетевые).
* **Стереотипы**: Стереотипы могут быть использованы для обозначения типа узла или артефакта (например, <<database>>, <<server>>).

**5) Укажите основные виды связей между компонентами и между узлами.**

Соединения являются разновидностью ассоциации и изображаются отрезками линий без стрелок. Наличие такой линии указывает на необходимость организации физического канала для обмена информацией между соответствующими узлами.