Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Проектирование программного обеспечения»

Отчёт по лабораторной работе №6

Студент: Дрозд А. И.

ФИТ 3 курс 2 группа

Преподаватель: Якунович А. В.

Минск 2023

**Лабораторная работа №6**

**Тема:** 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 архитектуры информационной системы с применением методологии UML.

**Постановка задачи (описание функциональных требований):**

* Управление турами и путевками: добавление новых туров с указанием места назначения, описания, цены, дат начала и окончания; редактирование и удаление существующих туров; просмотр списка доступных туров.
* Отчеты и аналитика: генерация отчетов о продажах, бронированиях и финансовых показателях; создание дашбордов и статистики продаж.
* Интерфейс для пользователей: создание учетных записей для сотрудников с разными уровнями доступа (администраторы, менеджеры склада и операторы).
* Выдача путевок: предоставление туристических путевок клиентам после завершения оплаты.
* Нормативы и правила: контроль соответствия этапов оформления путевки различным законодательным актам, ведомственным инструкциям, должностным нормативам и правилам.

**Описание программных средств:**

* Название приложения: Draw.io
* Версия: 21.8.2
* Разработчик: Jgraph
* Адрес загрузки: https://app.diagrams.net/
* Режим использования: веб-сервис
* Доступность на платформах: поддерживается на всех популярных веб-браузерах (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).

**Описание практического задания:**

На основе поставленной задачи нам было необходимо построить диаграмму развертывания и диаграмму компонентов. В данном случае эти две диаграммы были объединены в одну. Результат показан на рисунке 1:

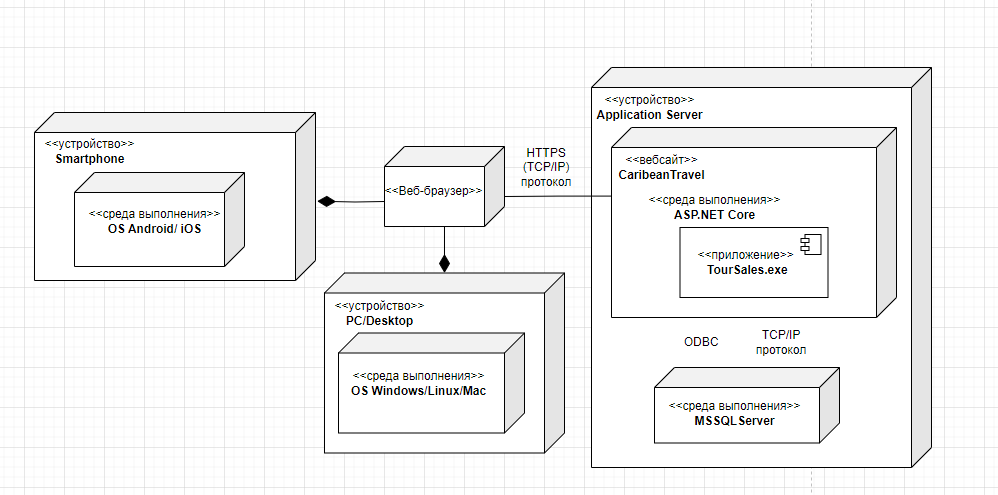
****

Рисунок 1 – Диаграмма развертывания и компонентов

**Список компонент, из которых состоит подсистема:**

1. Операционная система (Android, Linux, Mac, Windows, iOS) – включает в себя такие технологии, как Microsoft Windows, Apple Mac OS, Linux OS, Android, Apple iOS.
2. Приложение – в качестве технологии реализации используется язык программирования C#.
3. PC/Notepad/Smartphone (персональные устройства пользования) – включает основные технологии, используемые для данных видов устройств.
4. Database – используется технология Microsoft SQL Server.
5. ASP.NET Core: Технология реализации – ASP.NET Core.

**Физические устройства:**

1. Устройство «PC/Notepad»:

Основные характеристики – ПК или ноутбук, поддерживающий работу с веб-приложениями. Операционные системы: Windows 10 и выше, Mac OS, Linux.

1. Устройство «Smartphone»:

Основные характеристики – мобильное устройство, позволяющее взаимодействовать с веб-приложением. Операционные системы: ANDROID 7.0 и выше, iOS 8 и выше.

1. Устройство «Application Server»:

Основные характеристики – физический сервер, способный запускать и обслуживать базу данных MS SQL Server, а также содержащий экземпляр приложения TourSales.

**Соединения - протоколы связи между устройствами:**

1. Соединение между «PC/Notepad» или «Smartphone» и «Application server» осуществляется по протоколу связи HTTPS.
2. Соединение между «Application server» и «MSSQLServer» осуществляется по протоколу связи TCP/IP.

**Среды выполнения:**

1. ASP.NET Core: компонент «Приложение» размещается в среде выполнения «ASP.NET Core».
2. Операционная система (Windows/ Linux/ MacOs/ Android/ iOS): компонент «Операционная система» является обязательной частью «Устройства» и размещается там же.
3. Database: компонент «MS SQL Server» размещается в среде выполнения «Сервер приложения».

**Размещение компонентов на устройствах:**

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

**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.

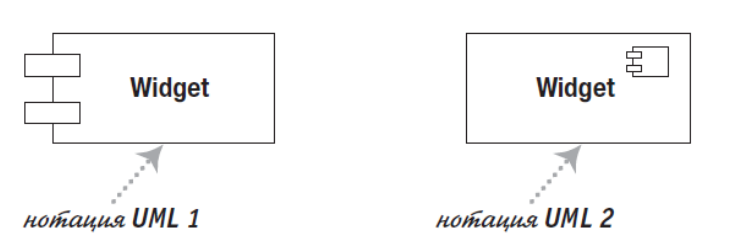


Рисунок 2 – Нотации для представления компонентов

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.