Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование программного обеспечения»

Отчёт по лабораторной работе №1

Автовокзал

Выполнил: Данилецкий М. В.

ФИТ 3 курс 7 группа

Преподаватель: Якунович А. В.

1. **Введение**

Данный документ представляет собой полный набор требований к разработке программного средства для автовокзала, которое далее будет упоминаться как “BusGo”. Это программное средство будет создано с целью улучшения и оптимизации работы автовокзала, а также повышения уровня обслуживания клиентов и эффективности управления операциями.

Автовокзалы являются важными пунктами транспортной инфраструктуры, предоставляющими услуги по перевозке пассажиров. Эффективное управление автовокзалом требует точной и оперативной обработки информации о рейсах, билетах, пассажирах, а также многих других аспектах его деятельности. Программное средство "BusGo" будет разработано для упрощения и автоматизации этих процессов.

Целевая аудитория "BusGo" включает в себя представителей разных возрастных групп и технической подготовки, а также сотрудников автовокзала, которые интересуются получением актуальной информации о рейсах и бронировании билетов. Данное приложение призвано облегчить их доступ к этой информации, экономя их время и обеспечивая удобство в использовании.

Цель "BusGo" – стать надежным и инновационным решением для автовокзалов, которое будет способствовать улучшению качества перевозок, удовлетворенности клиентов и эффективности операций. Требуется создать приложение, которое будет служить долгосрочным интересам как пассажиров, так и автовокзалов, внося вклад в современную транспортную индустрию.

1. **Анализ предметной области**

**2.1 Назначение проекта**

Назначение проекта "BusGo" заключается в разработке и внедрении программного приложения для автовокзалов с целью совершенствования и оптимизации их операций, а также улучшения обслуживания пассажиров.

**2.2 Цель проекта**

Целью проекта "BusGo" является создание и внедрение программного приложения для автовокзалов с целью:

* Улучшения обслуживания пассажиров: Обеспечение клиентам более удобного, оперативного и информативного опыта при бронировании билетов, получении информации о рейсах и управлении своими поездками.
* Оптимизации операций автовокзалов: Повышение эффективности работы автовокзалов через автоматизацию процессов бронирования, управления бронированиями, мониторинга рейсов и анализа данных.
* Создания инновационного решения: Разработка современного, интуитивно понятного и гибкого программного приложения, которое соответствует современным стандартам и ожиданиям пользователей.
* Увеличения конкурентоспособности автовокзалов: Помощь автовокзалам в привлечении большего количества пассажиров, повышение лояльности клиентов и оптимизация их операций.
* Улучшения аналитики и принятия решений: Предоставление автовокзалам инструментов для сбора и анализа данных о пассажирских потоках, что позволит им лучше понимать потребности своих клиентов и принимать обоснованные решения по управлению ресурсами и маршрутами.

**2.3 Целевая аудитория**

Программное приложение "BusGo" рассчитано на широкий спектр пользователей, включая:

* Пассажиров, ищущих удобный способ бронирования билетов и получения информации о рейсах.
* Работников автовокзалов, ответственных за управление бронированиями и оперативное взаимодействие с клиентами.
* Администраторов и аналитиков, использующих приложение для мониторинга и анализа пассажирских данных и операций.

**3**. **Требования к программному средству**

**3.1 Общие функциональные требования**

1. Интерфейс и удобство использования:

* Предоставление интуитивно понятного интерфейса для всех категорий пользователей;
* Адаптивный дизайн для поддержки разных устройств и экранов.

1. Бронирование билетов:

* Возможность для пассажиров бронировать билеты с выбором маршрута, даты и места.

1. Управление бронированиями:

* Возможность просматривать, изменять и отменять существующие бронирования.

1. Информация о рейсах:

* Предоставление актуальной информации о расписании рейсов, включая даты, времена и места отправления и прибытия.

1. Административные функции:

* Инструменты для управления данными о рейсах, билетах и пассажирах для сотрудников автовокзала;
* Опции для администраторов, позволяющие настраивать и обслуживать систему.

**3.2 Пользовательские требования**

Для клиентов должна быть разработана следующая функциональная часть:

1. Подбор и бронирование билетов:

* Пользователи могут просматривать доступные маршруты, выбирать даты и места отправления, а также бронировать билеты онлайн.

1. Управление бронированиями:

* Возможность просматривать, изменять и отменять существующие бронирования через личные аккаунты.

1. Просмотр информации о рейсах:

* Доступ к актуальной информации о расписании рейсов, маршрутах и стоимости билетов.

1. Личные аккаунты:

* Регистрация и создание личных аккаунтов с сохранением истории бронирований и предпочтений.

1. Сбор данных о пассажирах:

* Заполнение информации о пассажирах, что упрощает процесс бронирования при последующих поездках.

1. Настройка приложения:

* Возможность настройки параметров приложения, таких как язык интерфейса и стандарты безопасности.

Для администраторов должна быть разработана следующая функциональная часть:

1. Управление данными о рейсах:

* Добавление, редактирование и удаление информации о рейсах и маршрутах.

1. Управление бронированиями:

* Просмотр и управление существующими бронированиями, включая изменение и отмену броней.

1. Анализ данных:

* Использование аналитических инструментов для мониторинга и анализа пассажирских данных и операций автовокзала.

1. Управление аккаунтами пользователей:

* Управление аккаунтами пользователей, включая блокировку и разблокировку.

**3.3 Системные требования**

**3.3.1. Требования к архитектуре системы**

1. Модульная архитектура:

* Система должна быть построена на модульной архитектуре, что обеспечит гибкость и масштабируемость.

1. Клиент-серверная архитектура:

* Разделение приложения на клиентскую и серверную части для обеспечения масштабируемости и управления данными.

1. Использование микросервисов:

* Возможность разработки и развертывания отдельных микросервисов для различных функциональных блоков приложения.

**3.3.2. Требования к параметрам оборудования**

1. Серверное оборудование:

* Мощные серверы с достаточным объемом оперативной памяти, процессорами высокой производительности и большими объемами дискового пространства для обработки запросов и хранения данных.

1. Сетевое оборудование:

* Высокоскоростные сетевые адаптеры и коммутаторы для обеспечения быстрого и надежного обмена данными между клиентами и серверами.

1. Системы хранения данных:

* Доступ к надежным и высокопроизводительным системам хранения данных, таким как RAID-массивы или хранилища данных с резервными копиями.

**3.3.3. Требования к параметрам системы**

1. Системная безопасность:

* Обеспечение высокого уровня системной безопасности для предотвращения несанкционированного доступа к серверам и данным. Включает в себя механизмы обнаружения и предотвращения вторжений.

1. Скорость обработки транзакций:

* Система должна быть способной обрабатывать транзакции, связанные с бронированием билетов, с высокой скоростью для предотвращения задержек и сокращения времени ожидания пользователей.

1. Максимальное число одновременных сессий пользователей:

* Система должна поддерживать множество одновременных сессий пользователей, что позволяет множеству клиентов работать с приложением одновременно.

1. Время отклика:

* Система должна обеспечивать быстрый и отзывчивый интерфейс с минимальным временем отклика на действия пользователя.

**3.3.4. Требования к программному интерфейсу**

1. RESTful API:

* Реализация RESTful API для обеспечения взаимодействия с внешними системами и клиентами. API должен быть хорошо документирован и поддерживать стандартные HTTP методы.

1. Форматы данных:

* Поддержка различных форматов данных для обмена информацией, таких как JSON и XML.

1. Аутентификация и авторизация:

* Механизмы аутентификации и авторизации для обеспечения безопасности при взаимодействии с API.

1. Многопоточность:

* Поддержка многопоточности в API для обработки одновременных запросов от множества клиентов.

**3.3.5. Требования к структуре системы**

1. Распределенность: Система должна поддерживать распределенное хранение данных, обеспечивая доступность информации в разных точках системы. Это позволит достичь высокой отказоустойчивости и устранить единую точку сбоя, позволяя системе продолжать функционировать при возникновении проблем в отдельных компонентах. Распределенность данных также обеспечит более эффективное использование ресурсов и более быстрый доступ к информации.
2. Безопасность: Система должна обладать надежными механизмами безопасности для защиты данных, личной информации пользователей и обеспечения конфиденциальности. Это включает аутентификацию и авторизацию пользователей, шифрование данных, защиту от несанкционированного доступа и механизмы резервного копирования для предотвращения потери информации.
3. Интеграция с другими системами: Система должна быть способна интегрироваться с другими внешними системами, такими как системы бронирования билетов, системы учета и другие. Это позволит обмениваться данными и взаимодействовать с внешними системами, улучшая взаимодействие с клиентами и оптимизируя бизнес-процессы.
4. Масштабируемость: Система должна быть масштабируемой и способной распространяться на большое количество серверов, не потеряв при этом работоспособности и эффективности. Масштабирование должно определяться только мощностью аппаратного обеспечения. Это позволит системе удовлетворять потребности растущего числа пользователей и обрабатывать большие объемы данных, сохраняя высокую производительность.
5. Отказоустойчивость и восстановление после сбоев: Система должна быть способна обнаруживать и обрабатывать сбои, минимизировать их влияние на работу системы и обеспечивать быстрое восстановление после сбоев. Это включает механизмы резервного копирования, репликацию данных, мониторинг состояния системы и автоматическое восстановление после сбоев.

**3.3.6. Требования по взаимодействию и интеграции с другими системами**

Интеграция с базой данных автовокзала:

* Система должна взаимодействовать с базой данных автовокзала для получения актуальной информации о рейсах и бронированиях;
* Должна быть обеспечена возможность обновления информации в реальном времени.

**3.5 Предполагаемые технологии**

Frontend-составляющая: HTML5, CSS3, JavaScript.

Backend-составляющая: Node.js.

СУБД: Oracle.

**3.6 Основные аппаратные требования**

Рекомендованные аппаратные требования для серверной машины:

Процессор: Многоядерный процессор с частотой не менее 2.5 ГГц.

Оперативная память: минимум 16 ГБ.

Хранилище: Быстрые и надежные накопители, такие как SSD, с объемом не менее 512 ГБ для операционной системы и приложения.

Сетевой интерфейс: Двойной сетевой интерфейс для обеспечения отказоустойчивости и возможности настройки балансировки нагрузки.

Операционная система: Серверная операционная система, такая как Windows Server или Linux (например, CentOS, Ubuntu Server).

Рекомендованные аппаратные требования для пользовательской машины:

Процессор: Четырехядерный процессор с частотой не менее 2.0 ГГц.

Оперативная память: минимум 4 ГБ.

Хранилище: HDD или SSD с объемом 128 ГБ.

Сетевой доступ: Доступ к высокоскоростному интернет-соединению для связи с серверами приложения и получения обновлений.

Операционная система: Windows 10, macOS, Android 10.

**4. Стадии и этапы разработки**

1. Планирование:

* Определение требований. Сбор и анализ требований к приложению от заказчика и конечных пользователей.
* Планирование проекта. Определение сроков, ресурсов и создание плана разработки.

1. Проектирование:

* Архитектурное проектирование. Разработка общей архитектуры приложения, определение компонентов и связей между ними.
* Проектирование пользовательского интерфейса. Создание дизайна интерфейса приложения, включая макеты и схемы.
* База данных. Проектирование структуры базы данных и выбор используемой технологии.

1. Разработка:

* Написание кода. Разработка и программирование компонентов приложения в соответствии с проектной архитектурой.
* Тестирование. Проведение модульного тестирования для проверки работоспособности отдельных компонентов.

1. Интеграция и тестирование:

* Интеграция. Объединение компонентов приложения и тестирование их взаимодействия.
* Системное тестирование. Проведение тестирования всей системы на соответствие требованиям и выявление ошибок.

1. Отладка и оптимизация:

* Отладка. Исправление ошибок и доработка приложения на основе результатов тестирования.
* Оптимизация. Улучшение производительности, устранение узких мест.

**5. Заключение**

Разработка программного приложения "BusGo" представляет собой масштабное и многогранный процесс, который включает в себя множество этапов и задач. Это приложение предназначено для автовокзалов и обладает потенциалом существенно улучшить и оптимизировать процессы управления автовокзалом, обслуживания клиентов и обработки данных о рейсах и бронированиях.

На каждом этапе разработки, начиная с сбора требований и заканчивая поддержкой и сопровождением, необходимо уделять особое внимание качеству, безопасности и производительности приложения. Система должна быть отзывчивой, надежной и масштабируемой, чтобы обеспечить эффективное взаимодействие с пользователями и бесперебойную работу автовокзала.

По завершении разработки и внедрения приложения, важно обеспечить его поддержку и дальнейшее развитие, учитывая потребности и обратную связь пользователей.

Процесс разработки "BusGo" представляет собой сложный и многозадачный проект, и его успешное завершение требует внимания к деталям, соблюдения сроков и стремления к высокому качеству. Приложение "BusGo" имеет потенциал улучшить работу автовокзалов и облегчить жизнь пассажиров, и успешное выполнение этого проекта может принести значительные выгоды как бизнесу, так и конечным пользователям.