Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО ITMO University**

# ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

**По дисциплине** Проектирование программного обеспечения

**Обучающийся** Ряднов Вениамин Сергеевич

**Факультет** факультет прикладной информатики (ФПИн)

**Группа** К3250

**Направление подготовки** 09.03.03 Прикладная информатика

**Образовательная программа** Прикладная разработка информационных систем

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся | (дата) | \_\_\_\_\_\_\_\_ (подпись) | Ряднов В.С. (Ф.И.О.) |
| Руководитель | (дата) | \_\_\_\_\_\_\_\_ (подпись) | Меркушев А.Е. (Ф.И.О.) |

# СОДЕРЖАНИЕ

1. Общие сведения 4  
   1.1 Полное наименование системы и её условное обозначение 4  
   1.2 Плановые сроки начала и окончания работы по созданию   
   системы 4  
   1.3 Порядок оформления и предъявления заказчику результатов   
   работ 4
2. Назначение и цели создания системы 5  
   2.1 Назначение системы 5  
   2.2 Цели создания системы 5
3. Характеристики объекта автоматизации 6  
   3.1 Существующие проекты 9
4. Требования к системе 10  
   4.1 Требования к системе в целом 10  
   4.2 Требования к функциям (задачам), выполняемым подсистемой 27  
   4.3 Требования к видам обеспечения 33
5. Состав и содержание работ по созданию системы 36
6. Порядок контроля и приемки системы 39
7. Требования к составу и содержанию работ по подготовке   
   системы к вводу 42
8. Требования к документированию 43
9. Источники разработки 44

# 

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1 Полное наименование системы и её условное обозначение

Автоматизированная система управления заказами такси (АС “Обмен заказами такси”). Система предназначена для автоматизации процессов обработки заявок клиентов, распределения заказов между водителями, а также для обеспечения мониторинга выполнения поездок. Она разработана с учётом современных требований к программному обеспечению, включая надёжность, масштабируемость и безопасность данных.

Система включает модули для обработки заявок, управления пользователями и формирования отчётов. Важной особенностью является возможность интеграции с внешними картографическими сервисами, такими как Яндекс.Карты или Google Maps, а также поддержка онлайн-платежей для удобства пользователей.

## 1.2 Плановые сроки начала и окончания работы по созданию системы

Работы над системой начнутся в январе 2025 года. Первый этап включает сбор требований, анализ конкурентов и проектирование архитектуры системы. Второй этап — разработка функциональных модулей, интеграция с внешними сервисами и тестирование. Завершение всех этапов, включая ввод системы в эксплуатацию, запланировано на май 2025 года. В случае необходимости будут внесены дополнительные изменения по результатам тестирования и обратной связи.

## 1.3 Порядок оформления и предъявления заказчику результатов работ

Результаты работы включают техническую документацию, описание архитектуры, функциональных модулей и диаграммы взаимодействий. Также предоставляется руководство пользователя и отчёты по тестированию системы. Все документы оформляются в соответствии с требованиями ГОСТ и передаются в электронном и бумажном форматах. По завершении внедрения проводится обучение персонала, предоставляется инструкция по эксплуатации и технической поддержке системы.

# 2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

## 2.1 Назначение системы

Система предназначена для автоматизации процессов управления заказами такси. Её основная задача — сократить время обработки заявок, повысить точность распределения заказов и упростить взаимодействие между клиентами, водителями и диспетчерами. Дополнительно система позволяет вести учёт выполненных заказов, анализировать эффективность работы водителей и формировать отчёты.

Система будет полезна как для небольших таксопарков, так и для крупных служб такси, предоставляя гибкие настройки и возможность масштабирования.

## 2.2 Цели создания системы

* Повышение эффективности обработки заявок клиентов.
* Оптимизация процессов распределения заказов среди водителей.
* Снижение временных затрат на обработку заказов.
* Обеспечение прозрачности выполнения заказов за счёт мониторинга в реальном времени.
* Интеграция с платёжными и картографическими сервисами для повышения удобства клиентов.
* Возможность формирования отчётов по эффективности работы системы и отдельных пользователей.
* Масштабируемость системы для поддержки роста числа пользователей и заказов.

# 

# 3. ХАРАКТЕРИСТИКИ ОБЪЕКТА АВТОМАТИЗАЦИИ

## 3.1 Существующие проекты

На сегодняшний день на рынке представлено множество аналогичных решений для управления заказами такси. Среди наиболее известных конкурентов можно выделить крупные агрегаторы, такие как Яндекс.Такси, Uber и Gett. Эти платформы предоставляют широкий функционал для клиентов, включая бронирование поездок, онлайн-оплату и отслеживание водителей в реальном времени.

Однако, несмотря на их популярность, многие из существующих решений имеют определённые недостатки. Основные из них включают высокие комиссии, сложности с интеграцией в локальные бизнес-процессы и отсутствие гибкости в настройке функционала под специфические требования отдельных компаний.

Предлагаемая система "Обмен заказами такси" ориентирована на устранение этих проблем. Она будет адаптирована под нужды конкретных организаций, что особенно важно для небольших таксопарков и региональных операторов. Система позволит интегрироваться с внутренними базами данных, а также будет поддерживать кастомизированные отчёты и аналитику.

# 

# 4. ТРЕБОВАНИЯ К СИСТЕМЕ

## 4.1 Требования к системе в целом

Система должна обеспечивать выполнение следующих ключевых функций:  
1. Регистрация и авторизация пользователей с разными уровнями доступа (клиенты, водители, диспетчеры).  
2. Создание, редактирование и удаление заказов с привязкой к клиентам и водителям.  
3. Автоматическое распределение заказов между водителями с учётом их текущего местоположения, загрузки и рейтинга.  
4. Поддержка уведомлений для клиентов и водителей о статусах заказов в режиме реального времени.  
5. Мониторинг выполнения заказов и сбор аналитических данных по работе системы.  
6. Генерация отчётов по выполненным заказам, доходам и производительности водителей.  
7. Интеграция с внешними сервисами, такими как картографические платформы (Google Maps, Яндекс.Карты) и платёжные системы.

Система должна быть доступна через веб-приложение и мобильные приложения для платформ iOS и Android. Интерфейс должен быть адаптивным, удобным и простым в использовании.

## 4.2 Требования к функциям (задачам), выполняемым подсистемой

### 4.2.1 Подсистема обработки заказов

* Приём заявок от клиентов в реальном времени.
* Распределение заказов между водителями с учётом их геолокации и занятости.
* Обновление статуса заказа (ожидает выполнения, в пути, выполнен).
* Отправка уведомлений клиентам и водителям об изменении статусов.

### 4.2.2 Подсистема управления пользователями

* Регистрация новых пользователей с проверкой данных.
* Аутентификация пользователей с применением безопасных алгоритмов шифрования.
* Управление ролями и правами доступа.
* Хранение данных профилей и истории заказов.

### 4.2.3 Подсистема отчётности

* Генерация отчётов по выполненным заказам с разбивкой по датам, клиентам и водителям.
* Анализ производительности водителей на основе данных о количестве выполненных заказов и отзывов клиентов.
* Формирование отчётов по доходам и затратам, связанным с эксплуатацией системы.
* Экспорт отчётов в форматы CSV и PDF для дальнейшей обработки и анализа.

### 4.2.4 Подсистема уведомлений

* Отправка уведомлений о статусе заказов клиентам и водителям в реальном времени.
* Оповещение о критических событиях, таких как отмена заказа или сбой в работе системы.
* Поддержка различных каналов доставки уведомлений, включая SMS, электронную почту и push-уведомления в мобильных приложениях.

## 4.3 Требования к видам обеспечения

### 4.3.1 Информационное обеспечение

* Система должна хранить данные в централизованной базе данных с поддержкой резервного копирования.
* Данные пользователей и заказов должны быть зашифрованы для обеспечения безопасности.
* Предусмотреть возможность импорта и экспорта данных в различных форматах (JSON, XML, CSV).
* Обеспечить контроль целостности данных и защиту от их потери при сбоях.

### 4.3.2 Лингвистическое обеспечение

* Интерфейс системы должен поддерживать мультиязычность с предустановленными русским и английским языками.
* Реализовать возможность добавления дополнительных языков по запросу заказчика.
* Все элементы интерфейса, включая кнопки, уведомления и отчёты, должны быть переведены на выбранные языки.

### 4.3.3 Программное обеспечение

* Серверная часть разрабатывается с использованием современных технологий, таких как Node.js, Python или Java.
* Клиентская часть реализуется на React, Angular или Vue.js для обеспечения кроссплатформенности.
* База данных строится на PostgreSQL или MySQL с поддержкой масштабирования и кластеризации.
* Приложения для мобильных устройств разрабатываются на базе React Native или Flutter для поддержки iOS и Android.

## 4.3.4 Техническое обеспечение

* Серверы должны обеспечивать стабильную работу системы при одновременной нагрузке до 2000 активных пользователей.
* Серверная инфраструктура должна поддерживать горизонтальное масштабирование для увеличения производительности.
* Минимальные требования к серверам:
  + Процессор: не менее 8 ядер.
  + Оперативная память: не менее 16 ГБ.
  + Хранилище: от 500 ГБ с поддержкой RAID-массивов для резервирования данных.
* Возможность развёртывания системы в облачных сервисах, таких как AWS или Google Cloud Platform.

## 4.3.5 Организационное обеспечение

* Предусмотреть обучение персонала, ответственного за администрирование и поддержку системы.
* Разработать руководство пользователя для всех категорий пользователей (клиенты, водители, диспетчеры).
* Организовать службу технической поддержки, доступную 24/7.

### 4.3.6 Методическое обеспечение

* Разработать детальные инструкции по установке, настройке и эксплуатации системы.
* Предоставить методические рекомендации для проведения тестирования и внедрения обновлений.
* Подготовить регламент работы с данными и их защиты в соответствии с законодательными требованиями.
* Создать обучающие материалы для пользователей и администраторов системы.

# 

# 5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

## 5.1 Задачи, решаемые в рамках создания системы

* Анализ требований заказчика и сбор данных о текущих бизнес-процессах.
* Проектирование архитектуры системы, включая базу данных и модули функциональности.
* Разработка веб- и мобильных приложений для клиентов, водителей и диспетчеров.
* Интеграция системы с внешними сервисами карт и онлайн-оплат.
* Тестирование отдельных модулей и системы в целом для проверки работоспособности.
* Подготовка документации, включая технические описания, инструкции и отчёты по тестированию.
* Внедрение системы в эксплуатацию и обучение персонала.
* Постпроектная поддержка, включая исправление ошибок и выпуск обновлений.

## 5.2 Содержание и результаты работ

### Этап 1: Сбор требований и проектирование

* Анализ бизнес-процессов заказчика.
* Описание ключевых функций системы.
* Определение архитектурных решений и схем взаимодействия модулей.
* Разработка макетов интерфейса и прототипов.

**Результаты этапа:**  
- Документ с функциональными требованиями.  
- Прототип интерфейса для демонстрации.  
- Архитектурная схема системы.

### Этап 2: Разработка программного обеспечения

* Написание серверной части с реализацией логики обработки заказов, аутентификации и управления ролями.
* Создание клиентской части для веб- и мобильных платформ.
* Интеграция с картографическими и платёжными сервисами.

**Результаты этапа:**  
- Рабочий прототип системы с базовыми функциями.  
- Исходный код всех компонентов системы.  
- Инструкции по развертыванию приложения.

### Этап 3: Тестирование и оптимизация

* Проведение модульного и интеграционного тестирования.
* Нагрузочное тестирование системы при высоком числе запросов.
* Оптимизация производительности и устранение выявленных ошибок.

**Результаты этапа:**  
- Протоколы тестирования с описанием найденных ошибок и их исправлений.  
- Финальная версия системы, готовая к эксплуатации.

### Этап 4: Внедрение и обучение персонала

* Развёртывание системы на сервере заказчика или в облаке.
* Обучение сотрудников работе с системой.
* Создание службы технической поддержки.

**Результаты этапа:**  
- Инструкции по работе с системой.  
- Подготовленные сотрудники, готовые к использованию системы.  
- Документ о вводе системы в эксплуатацию.

# 

# 6. ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ СИСТЕМЫ

## 6.1 Виды, состав, объём и методы испытаний системы

Проверка работоспособности и качества системы осуществляется поэтапно с использованием следующих методов:

### Функциональное тестирование:

* Проверка соответствия работы системы заявленным требованиям.
* Тестирование всех функций, включая создание, редактирование и удаление заказов, авторизацию, отчёты и уведомления.
* Проверка работы интеграции с картографическими и платёжными сервисами.

### Нагрузочное тестирование:

* Моделирование высокой нагрузки на систему с одновременным использованием нескольких пользователей.
* Измерение производительности в условиях большого количества запросов.
* Тестирование реакции системы на пиковые нагрузки и проверка стабильности работы.

### Интеграционное тестирование:

* Проверка взаимодействия модулей системы между собой.
* Тестирование обмена данными между сервером и клиентскими приложениями.
* Проверка работы внешних API и интеграций.

### Приёмочные испытания:

* Проверка системы заказчиком на соответствие требованиям.
* Демонстрация всех функций системы в рабочей среде.
* Испытания в условиях, максимально приближенных к реальным.

## 6.2 Общие требования к приемке работ по стадиям

Приемка работы осуществляется поэтапно:

1. **Этап разработки прототипа:**
   * Проверка архитектуры и базовых функций.
   * Утверждение макетов интерфейса.
2. **Этап разработки функциональных модулей:**
   * Проверка функциональности каждой подсистемы.
   * Утверждение отчётов по промежуточным результатам.
3. **Этап тестирования:**
   * Утверждение протоколов тестирования.
   * Проверка стабильности и производительности.
4. **Этап внедрения:**
   * Проверка настройки серверов и интеграции.
   * Утверждение готовности к эксплуатации.

## 6.3 Статус приемочной комиссии

Для организации и контроля над приёмкой работ создаётся комиссия, состоящая из:  
- Руководителя проекта.  
- Представителей заказчика.  
- Архитектора системы.  
- Тестировщиков.  
- Разработчиков и администраторов.

Комиссия оценивает соответствие результатов работ техническому заданию, проверяет отчёты по тестированию и принимает окончательное решение о вводе системы в эксплуатацию.

# 

# 7. ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

## 7.1 Организационные мероприятия

* Обеспечение доступа к серверам и оборудованию для установки системы.
* Предоставление тестовой среды для проведения испытаний.
* Организация обучения персонала и технической поддержки.
* Разработка инструкций для работы с системой.

## 7.2 Подготовка инфраструктуры

* Настройка серверного оборудования и сетевого окружения.
* Установка необходимого программного обеспечения и баз данных.
* Проверка совместимости системы с существующими решениями заказчика.

# 

# 8. ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

## 8.1 Перечень документации

Система сопровождается следующими документами:  
- Техническое задание.  
- Руководство пользователя.  
- Руководство администратора.  
- Протоколы тестирования.  
- Инструкции по установке и настройке.  
- Отчёты по тестированию и производительности.

## 8.2 Формат документации

* Электронный формат (PDF, DOCX).
* Бумажные копии для хранения в архиве заказчика.
* Все документы оформляются в соответствии с требованиями ГОСТ.

# 

# 9. ИСТОЧНИКИ РАЗРАБОТКИ

1. ГОСТ 34.602-89 "Техническое задание на создание автоматизированной системы".
2. Методические указания по проектированию ПО.
3. Руководства по шаблонам проектирования (GoF, GRASP).
4. Документация по интеграции с картографическими и платёжными сервисами.
5. Материалы исследований в области автоматизации процессов управления заказами такси.

# 

# ПРИЛОЖЕНИЕ А

Примерные схемы архитектуры системы и макеты интерфейса будут приложены отдельно.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
|  | |  | |