Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №3**

**по дисциплине**  
 **«Технологии проектирования программного обеспечения»**

Работу выполнил студент группы 34 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.М. Урюпин

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Отчет принял   
доц. каф. ИТ            А. Н. Полетайкин

Тема – техническое задание на создание программного продукта.

Цель – освоение методики предварительного анализа разрабатываемой программы; освоение задач формулирования функциональных и нефункциональных требований к программной реализации отдельных задач и к программе в целом; выработка навыков разработки технического задания.

Задание:

1. Установить назначение и общую цель создания программы.
2. Определить структуру программы и состав функциональных задач.
3. Разработать функциональные требования к программе:

* требования к входным и выходным данным;
* требования к программной реализации задач;
* специальные требования к математическому обеспечению программной реализации задач;

1. Разработать модель требований в нотации UML
2. Разработать требования к информационному обеспечению (к базе данных)
3. Разработать требования к инструментальному программному обеспечению (к системе управления базой данных (СУБД), к средству разработки программ (IDE), средствам автоматизированного проектирования ПО)
4. Установить нефункциональные требования к программе. Дать не менее 5 наиболее очевидных для данной системы требований из приложения В.

Индивидуальная тема: ПС аренды автомобилей.

**1 Назначение подсистемы и цели создания**

Подсистема предназначена для автоматизации процессов аренды автомобилей, включая бронирование, управление автопарком, обработку платежей и взаимодействие с клиентами. Она обеспечивает удобный доступ к информации о доступных автомобилях, упрощает процесс оформления аренды и снижает нагрузку на сотрудников прокатного бюро. Назначением программы является автоматизация процесса "Аренда автомобилей" для организации " АвтоРент ".

Цель создания подсистемы аренды автомобилей — автоматизировать процесс аренды транспортных средств, обеспечивая удобный интерфейс для клиентов и эффективное управление автопарком. Подсистема должна ускорить обработку заказов, упростить оформление аренды и оплату, а также минимизировать ошибки, связанные с ручной обработкой данных.

**2 Структура программы и состав функциональных задач**

Разрабатываемая программная система (ПС) аренды автомобилей должна обеспечивать автоматизацию ключевых процессов, связанных с управлением автопарком, оформлением аренды, обработкой платежей и взаимодействием с клиентами. Система будет построена по модульному принципу, что обеспечит ее расширяемость и возможность интеграции с дополнительными сервисами.

Основные функциональные задачи, которые должна решать ПС аренды автомобилей

1. Управление пользователями

* Регистрация и авторизация клиентов и сотрудников.
* Разграничение прав доступа (клиенты, менеджеры, администраторы).
* Восстановление пароля и управление профилями пользователей.

1. Каталог автомобилей

* Добавление, редактирование и удаление автомобилей из системы.
* Фильтрация и сортировка автомобилей по параметрам (марка, модель, класс, цена, статус).
* Хранение информации о техническом состоянии и доступности машин.

1. Оформление аренды

* Выбор автомобиля и сроков аренды.
* Автоматический расчет стоимости с учетом скидок и акций.
* Генерация договора аренды с возможностью просмотра и скачивания.

1. Оплата и финансовый учет

* Интеграция с платежными сервисами для онлайн-оплаты.
* Автоматический расчет штрафов за просрочку возврата.
* Управление программами лояльности и скидками.

1. Управление заказами

* Просмотр, изменение и отмена заказов.
* Отслеживание статусов аренды (активная, завершенная, отмененная).
* Ведение истории аренд каждого клиента.

1. Отчеты и аналитика

* Генерация отчетов по доходам, заказам и загруженности автопарка.
* Анализ популярности автомобилей и востребованности услуг.
* Мониторинг активности пользователей и финансовых показателей.

1. Интеграции и дополнительные функции

* Подключение к картографическим сервисам (Google Maps, OpenStreetMap) для отображения точек проката.
* Внедрение системы уведомлений (SMS, email) о статусе заказов.
* Возможность подключения модулей телеметрии и GPS-мониторинга автомобилей.

**Функциональные требования**

**3.1 Требования к входным и выходным данным**

Программная система аренды автомобилей должна обрабатывать различные входные и выходные данные, обеспечивая корректность и безопасность работы.

Входные данные:

* Данные клиентов (ФИО, контактная информация, паспортные данные).
* Данные о доступных автомобилях (модель, марка, характеристики, статус).
* Условия аренды (дата начала и окончания, тариф, дополнительные услуги).
* Платежные данные (сумма аренды, скидки, штрафы, статус оплаты).

Выходные данные:

* Сформированный договор аренды в электронном виде.
* Подтверждение бронирования автомобиля.
* Информация о платежах (статус оплаты, квитанции).
* Обновленные данные об автопарке (арендованный, доступный, на ремонте).

**3.2 Требования к программной реализации задач**

Все данные пользователей, записей на курсы, журнала успеваемости должны храниться в реляционной СУБД. Интерфейс пользователя состоит из формы регистрации, каталога курсов и личного кабинета. Все элементы пользовательского интерфейса находятся должны находиться на главной странице веб-портала.

Система должна поддерживать безопасное хранение данных пользователей и шифрование платежной информации.

Возможность работы в многопользовательском режиме с разграничением прав доступа.

Обеспечение автоматического расчета стоимости аренды и штрафов.

Архитектура должна поддерживать модульность, что обеспечит расширяемость и гибкость системы.

**3.3 Специальные требования к математическому обеспечению**

* Использование алгоритмов расчета стоимости аренды с учетом тарифов, скидок и штрафов.
* Оптимизация поиска автомобилей с учетом предпочтений клиента.
* Реализация динамического анализа спроса и предложения для формирования ценовой политики.
* Поддержка автоматизированного расчета маршрутов на основе картографических API.

**4 Диаграмма вариантов использования**

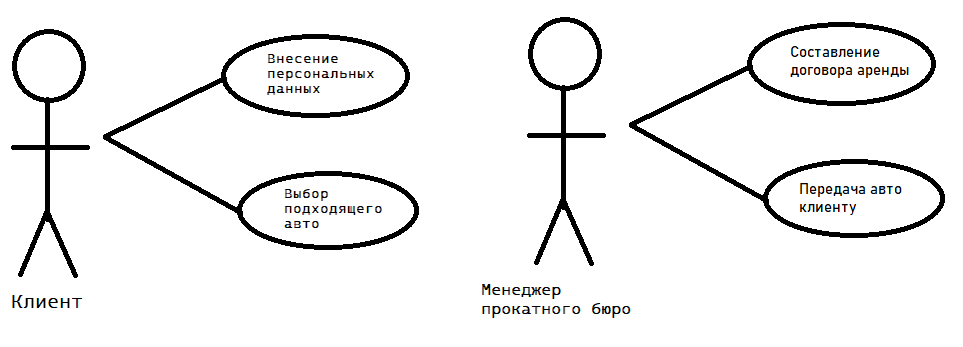


Рисунок 1 - Диаграмма вариантов использования

**5 Требования к информационному обеспечению (база данных)**

Требования к СУБД разрабатываемого ПО

Разрабатываемое ПО использует СУБД Postgresql

1 Требования к поддержке реляционных таблиц

СУБД должна поддерживать структуру базы данных из 4 таблиц: список слушателей автомобилей, список занятости автомобилей, таблица состояний автомобилей, журнал учёта аренды.

2 Требования CRUD

СУБД должна поддерживать принципы CRUD (Create, Read, Update, Delete) для каждой таблицы.

3 Требования к транзакциям

СУБД должна поддерживать структуру выполнения транзакций для обеспечения целостности данных.

4 Требования надёжности

СУБД должна быть надёжной и безопасной, обеспечивать отказоустойчивость и иметь систему Backup`ов (резервные копии данных).

5 Требования к совместимости

СУБД должна быть совместима с выбранным языком программирования разработки.

**6 Требования к инструментальному программному обеспечению**

Требования к IDE разрабатываемого ПО

Разрабатываемое ПО использует IDE Pycharm и Visual Studio Code

1 Требования к поддержке

IDE должна поддерживать язык программирования, используемый в разработке ПО

2 Требования к интегрированным инструментам

IDE должна иметь широкий спектр интегрированных инструментов для функциональной разработки ПО

3 Требования к веб-технологиям

IDE должна иметь поддержку таких веб-технологий как HTML, CSS, JavaScript.

4 Требования к Git

IDE должна быть совместима с системой контроля версий Git для комфортной разработки ПО

5 Требования к расширениям

IDE должна иметь возможность поддержки плагинов и расширений для повышения производительности и функциональности

6 Требования к совместимости

IDE должна быть совместима с операционной системой Windows 10, на которой ведётся разработка ПО

**7 Нефункциональные требования к системе**

Основные нефункциональные требования к ПО:

* Малая стоимость (данное ПО должно быть дешёвым в разработке, чтобы оно могло конкурировать на рынке, но это не должно сказаться на качестве разрабатываемого ПО);
* Быстрота обработки запросов (использование реляционной СУБД должно положительно сказать на скорости обработки запросов данного ПО);
* Простота использования (данное ПО должно иметь интуитивно-понятный интерфейс пользователя);
* **Безопасность данных**: Программная система должна обеспечивать высокий уровень безопасности данных, включая защиту персональных данных клиентов и безопасное хранение финансовой информации, соответствуя актуальным стандартам безопасности и защите от внешних угроз;
* Контроль работы ПО (данное ПО должно предоставлять доступ к отслеживанию различных шагов выполнения обработки запросов).