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 информационных технологий**

**ОТЧЕТ**

о выполнении лабораторной работы №3

по дисциплине «Программная инженерия»

Выполнил студент группы МО31/2                                 М. В. Варфоломеев

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Курс    3

Отчет принял кандидат педагогических наук, доцент                                                                            Н. Ю. Добровольская

Краснодар

2025 г.

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc208684339)

[1. Назначение и цели создания программы, структура программы 4](#_Toc208684340)

[2. Функциональные требования и модель требований UML 6](#_Toc208684341)

[3. Специальные требования к информационному и математическому обеспечению](#_Toc208684342) 10

[4. Требования к инструментальному ПО](#_Toc208684342) 11

[5. Нефункциональные требования](#_Toc208684342) 12

[Заключение 1](#_Toc208684343)4

**ВВЕДЕНИЕ**

Тема: техническое задание на создание программного продукта.

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

Индивидуальная тема: автоматизация учёта клиентов и услуг в парикмахерской.

Задачи:

1. Установить назначение и общую цель создания программы. Определить структуру программы и состав функциональных задач.
2. Разработать функциональные требования к программе и построить модель требований в нотации UML.
3. Разработать специальные требования к информационному и математическому обеспечению программной реализации задач.
4. Разработать требования к инструментальному ПО, необходимому для разработки и эксплуатации ПП.
5. Установить необходимые нефункциональные требования к ПП.
6. **Назначение и цели создания программы, структура программы**
   1. Назначение программного продукта.

Разрабатываемый программный продукт предназначен для автоматизации деятельности парикмахерских и салонов красоты. Программа охватывает основные бизнес-процессы, связанные с обслуживанием клиентов.

Автоматизируемые виды деятельности включают оперативный учет, управление клиентскими отношениями, планирование работы мастеров, финансовые расчеты.

Объектами автоматизации являются салон парикмахерских услуг, рабочие места администраторов и мастеров, зона приема клиентов. Персонал, работающий с системой, включает администраторов, мастеров-парикмахеров, управляющего и клиентов, использующих как онлайн-запись, так и личное посещение.

* 1. Цели создания программного продукта.

Технические цели включают сокращение времени обработки заявки с 5 до 2 минут, увеличение точности расчетов до 100 процентов, снижение количества ошибок учета до нуля в месяц. Дополнительно планируется уменьшение времени формирования отчетности с 30 до 5 минут и повышение доступности информации о расписании до 100 процентов.

Технологические цели предполагают внедрение онлайн-записи клиентов, оптимизацию загрузки мастеров, создание электронного документооборота и единого информационного пространства для сотрудников.

Социальные цели включают повышение удовлетворенности клиентов, улучшение условий труда персонала, рост профессионального уровня сотрудников и усиление лояльности клиентов.

Критерии оценки достижения целей: количество ошибок в расчетах должно быть сведено к нулю, удовлетворенность клиентов должна вырасти с 75 до 90 процентов, экономия времени администратора должна составить 2 часа в день, загрузка мастеров должна увеличиться с 65 до 80 процентов.

* 1. Структура программы и состав функциональных задач.

Программа имеет модульную структуру и состоит из семи основных функциональных блоков:

Модуль управления клиентской базой обеспечивает ведение карточек клиентов и учет контактной информации, формирование истории посещений.

Модуль управления расписанием предоставляет функции интерактивного планирования записей, учета графика работы мастеров, контроля конфликтов в расписании, управления перерывами, системы бронирования времени.

Модуль учета услуг и прайс-листов включает ведение каталога услуг, динамическое управление ценами, учет продолжительности услуг, формирование комплексных предложений, систему скидок и расчет себестоимости услуг.

Модуль управления персоналом содержит ведение профилей мастеров, учет рабочего графика, систему рейтингов и отзывов, расчет заработной платы, учет невыходов на работу и планирование повышения квалификации.

Модуль финансовой аналитики и отчетности выполняет расчет ежедневной выручки, формирование стандартных и произвольных отчетов, анализ популярности услуг.

Модуль складского учета включает учет материалов, контроль остатков, управление закупками, анализ оборачиваемости запасов и учет стоимости материалов в себестоимости услуг.

1. **Функциональные требования и модель требований UML**

2.1 Диаграмма вариантов использования

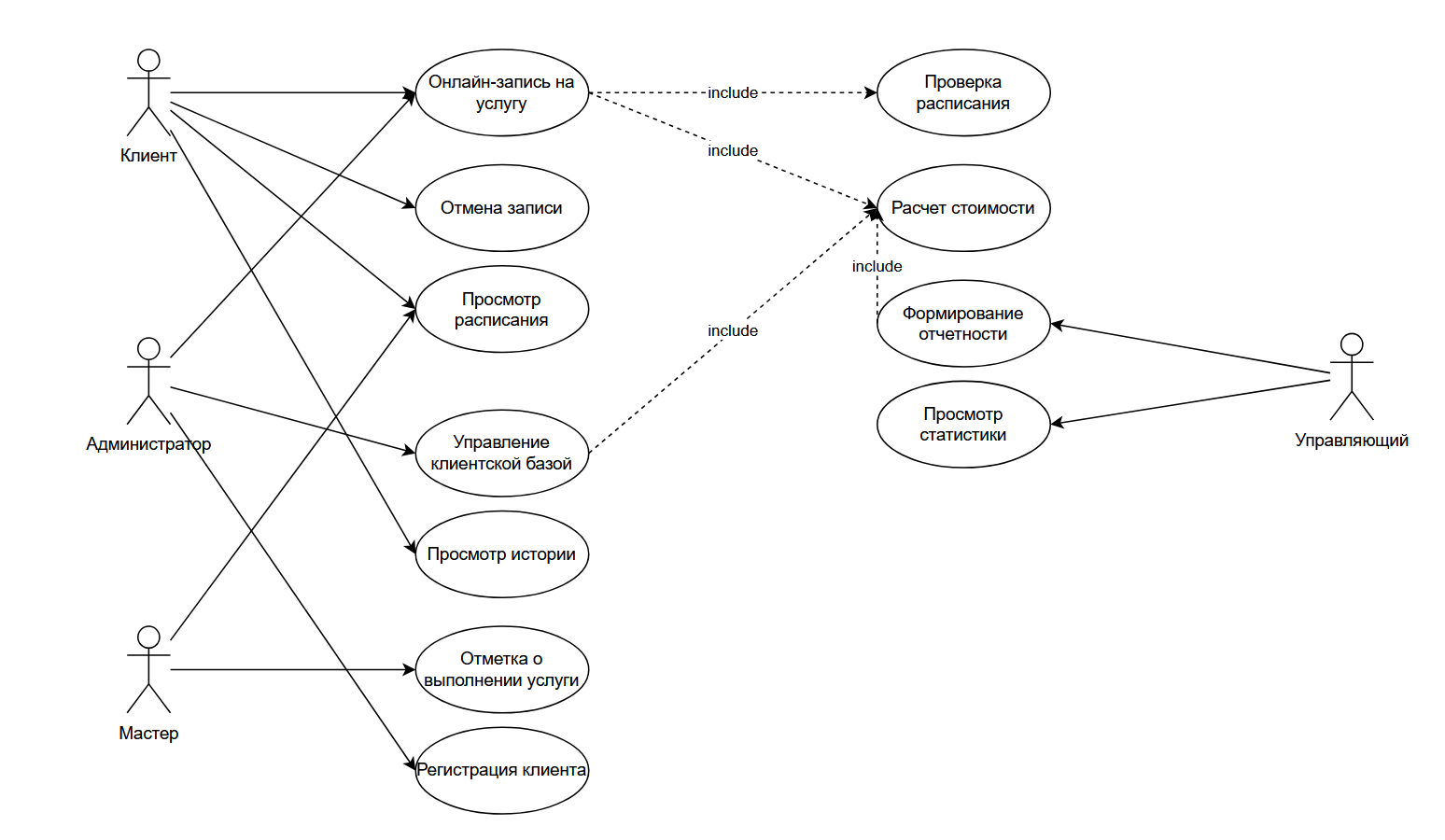


Рисунок 2.1 - Диаграмма вариантов использования ПП "Парикмахерская".

Акторы системы:

* Клиент - физическое лицо, пользующееся услугами
* Администратор - сотрудник по приему клиентов и ведению записей
* Мастер - специалист, оказывающий парикмахерские услуги
* Управляющий - руководитель салона

Основные прецеденты:

Для актора Клиент определены четыре прецедента. Просмотр расписания мастеров позволяет ознакомиться с доступным временем. Онлайн-запись на услугу обеспечивает дистанционное бронирование. Просмотр истории посещений дает доступ к информации о визитах. Отмена записи позволяет аннулировать бронирование.

Для актора Администратор установлены пять прецедентов. Управление клиентской базой охватывает операции с информацией. Регистрация новых клиентов включает внесение первичных данных. Формирование записей клиентов предполагает создание расписания. Расчет стоимости услуг обеспечивает определение цен. Просмотр текущего расписания дает обзор занятости.

Для актора Мастер определены четыре прецедента. Просмотр личного расписания предоставляет информацию о графике. Отметка о выполнении услуги фиксирует завершение работ. Ведение учета выполненных работ включает документальное оформление. Просмотр своей статистики позволяет анализировать показатели.

Для актора Управляющий установлены четыре прецедента. Формирование отчетности включает создание сводных документов. Анализ финансовых показателей обеспечивает оценку эффективности. Управление прайс-листом охватывает установление цен. Просмотр общей статистики дает представление о работе салона.

2.2 Детализация прецедентов

2.2.1 Требования к задаче "Управление клиентской базой".

Программа должна выполнять регистрацию новых клиентов. Программа должна вести историю посещений клиентов. Программа должна обеспечивать поиск клиентов по различным параметрам. Программа должна выполнять сегментацию клиентов.

Входными данными являются форма регистрации клиента. Источником данных служит ручной ввод администратора. Обязательными полями являются ФИО и телефон. Требуется точность формата номера телефона.

Выходными данными являются карточка клиента. Выходными данными являются списки клиентов с фильтрацией. Выходными данными являются статистические отчеты.

Требуется автоматическая валидация номера телефона. Требуется генерация уникального идентификатора клиента. Требуется проверка на дублирование записей.

2.2.2 Требования к задаче "Управление расписанием".

Программа должна отображать расписание мастеров. Программа должна выполнять бронирование временных слотов. Программа должна проверять конфликты в расписании. Программа должна учитывать графики работы мастеров.

Время отклика при проверке доступности не должно превышать 2 секунд. Обновление расписания должно происходить в реальном времени.

Требуется интуитивно понятное графическое представление. Требуется цветовая кодировка статусов записей. Требуется возможность перетаскивания записей.

2.2.3 Требования к задаче "Управление расписанием".

Программа должна вести каталог услуг. Программа должна управлять ценами. Программа должна учитывать продолжительность услуг. Программа должна формировать комплексные услуги.

Требуется контроль целостности данных при изменении цен. Требуется ведение истории изменений прайс-листа. Требуется проверка корректности расчетов.

2.2.4 Требования к задаче "Управление расписанием".

Программа должна рассчитывать ежедневную выручку. Программа должна формировать отчеты за произвольный период. Программа должна анализировать популярность услуг.

Точность финансовых расчетов должна составлять 100 процентов. Требуется защита отчетных данных от изменений.

2.3 Требования к программной реализации задач

2.3.1 Требования к организации хранения данных.

Для хранения данных должна использоваться реляционная база данных PostgreSQL. Версия базы данных должна быть не ниже 13. Таблицы должны быть нормализованы до третьей нормальной формы. Должно быть реализовано ежедневное автоматическое резервное копирование. Должна обеспечиваться транзакционная целостность при операциях записи.

2.3.2 Требования к структуре пользовательского интерфейса.

Интерфейс должен быть разделен на функциональные модули. Все экранные формы должны иметь единый визуальный стиль. Дизайн должен быть адаптивным к различным разрешениям экранов. Навигация должна быть интуитивной с минимальной глубиной вложенности.

2.3.3 Требования к методу программирования.

Должен применяться объектно-ориентированный подход программирования. Архитектура должна быть модульной с четким разделением ответственности. Должны соблюдаться соглашения об именовании переменных и методов. Должна быть реализована комплексная обработка исключительных ситуаций. API и ключевые алгоритмы должны быть документированы. Кодовая база должна поддерживаться в системе контроля версий Git.

2.3.4 Требования к безопасности данных.

Должна быть реализована аутентификация пользователей. Конфиденциальные данные клиентов должны шифроваться. Должна обеспечиваться защита от SQL-инъекций. Должны разграничиваться права доступа по ролям пользователей.

2.3.5 Требования производительности.

Время отклика интерфейса не должно превышать 3 секунд. Система должна поддерживать одновременную работу до 20 пользователей.

1. **Специальные требования к информационному и математическому обеспечению**
   1. Требования к информационному обеспечению.

Программа должна обрабатывать клиентские данные. Данные включают персональную информацию и историю посещений. Программа должна работать с данными мастеров. Данные мастеров содержат профили и специализации. Программа должна управлять учетными данными. Учетные данные включают услуги и расписание. Финансовые данные содержат расчеты и отчетность.

Для хранения данных должна использоваться СУБД PostgreSQL. Обязательными атрибутами клиента являются ФИО и контактный телефон. Формат хранения дат должен соответствовать стандарту ISO 8601. Текстовые поля должны поддерживать кириллические символы.

Должно быть реализовано разграничение прав доступа по ролям. Конфиденциальные данные клиентов должны шифроваться. Доступ к общим данным должен контролироваться.

* 1. Требования к математическому обеспечению.

Программа должна выполнять статистический анализ. Анализ используется для расчета среднего чека. Программа должна применять метод приоритетного планирования. Планирование используется для оптимизации расписания.

Должны быть реализованы алгоритмы поиска свободных окон. Алгоритмы должны учитывать расписание мастеров. Должны быть реализованы алгоритмы расчета стоимости комплексных услуг. Алгоритмы должны учитывать состав услуг.

Требуется валидация входных данных на корректность. Валидация предотвращает ошибки в расчетах. Требуется нормализация данных перед статистическим анализом. Нормализация обеспечивает сопоставимость показателей.

1. **Требования к инструментальному ПО**
   1. Требования к операционной среде.

Серверная часть должна работать под управлением Windows Server 2019. Допускается использование Linux Ubuntu 20.04 LTS. Веб-интерфейс должен работать в последних версиях браузеров.

* 1. Требования к инструментальным средствам разработки.

Для построения UML-диаграмм должно использоваться CASE-средство StarUML. Допускается применение Visual Paradigm. Для создания функциональных схем должен использоваться Draw.io.

Основным языком программирования должен быть JavaScript. Для backend-разработки должна применяться платформа Node.js. Для frontend-разработки должен использоваться фреймворк Vue.js. Для контроля версий должен применяться Git.

* 1. Требования **к готовым программным пакетам**.

В качестве СУБД должна использоваться PostgreSQL 13. Сервер приложений должен быть построен на Node.js. Для frontend-разработки должен использоваться Vue.js 3. Для роутинга должен применяться Vue Router. В качестве IDE должен использоваться VS code. Для backend-фреймворка должен применяться Express.js.

1. **Нефункциональные требования**
   1. Требования **к аппаратному обеспечению**.

Серверная часть должна требовать процессор с 4 ядрами. Тактовая частота процессора должна быть не менее 3 GHz. Оперативная память сервера должна составлять 8 GB. Дисковое пространство сервера должно быть не менее 100 GB. Должен использоваться SSD накопитель. Сетевое подключение сервера должно быть не менее 100 Mbps.

Клиентская часть должна работать на процессоре с 1 ядром. Тактовая частота клиентского процессора должна быть не менее 2.5 GHz. Оперативная память клиента должна составлять 4 GB. Разрешение экрана должно поддерживать 1280x720. Подключение к интернету должно быть не менее 10 Mbps.

* 1. Требования **к производительности**.

Время отклика интерфейса не должно превышать 3 секунд. Система должна поддерживать одновременную работу 20 пользователей. Обработка транзакций должна составлять 100 записей в минуту.

* 1. Требования **к надёжности**.

Время наработки на отказ должно составлять 720 часов. Восстановление после сбоя не должно превышать 30 минут. Целостность данных должна обеспечиваться на уровне 99.9%. Должна быть реализована отказоустойчивость системы.

* 1. Требования **к безопасности**.

Должна быть реализована аутентификация пользователей. Должно применяться шифрование конфиденциальных данных. Должна обеспечиваться защита от SQL-инъекций. Должны разграничиваться права доступа по ролям.

* 1. Требования **к удобству использования**.

Интерфейс должен быть интуитивно понятным. Обучение персонала не должно превышать 4 часов. Навигация должна быть простой и логичной.

* 1. Требования **к сопровождаемости**.

Архитектура должна быть модульной. Код должен быть документирован. Должна обеспечиваться обратная совместимость.

* 1. Требования **к мобильности**.

Реализация должна быть кроссплатформенной. Должна поддерживаться работа в различных браузерах. Интерфейс должен адаптироваться под разные разрешения экранов.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы было разработано техническое задание на создание программного продукта для автоматизации парикмахерской. Определены цели и структура системы. Сформулированы функциональные требования к модулям управления клиентами, расписанием, услугами и отчетностью.

Установлены специальные требования к информационному и математическому обеспечению. Определены требования к инструментальным средствам разработки на Vue.js и Node.js.

Сформулированы нефункциональные требования к производительности, надежности и безопасности системы. Техническое задание полностью готово для перехода к этапу проектирования и разработки программного продукта.