**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**КУРСОВОЙ ПРОЕКТ**

### НА ТЕМУ

|  |  |
| --- | --- |
| РАЗРАБОТКА ПОДСИСТЕМЫ | |
| «СМАРТФИКС. УЧЁТ ЗАЯВОК НА РЕМОНТ» | |
| Л109. 25КП01. 021 ПЗ | |

(Обозначение документа)

|  |
| --- |
| МДК.02.01 Технология разработки |
| программного обеспечения |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИСПП-35 | | 05.12.2025 | М.А Фролова |
|  | (Группа) | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель |  |  | 05.12.2025 | Ю.С. Маломан |
|  |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

**СОДЕРЖАНИЕ**

Перечень сокращений и обозначений 3

Введение 4

1 Анализ и разработка требований

1.1 Назначение и область применения

1.2 Постановка задачи

1.3 Описание алгоритма функционирования системы

1.4 Выбор состава программных и технических средств

2 Проектирование программного обеспечения

2.1 Проектирование интерфейса пользователя

2.2 Разработка архитектуры программного обеспечения

2.3 Проектирование базы данных

3 Разработка и интеграция модулей программного обеспечения

3.1 Разработка программных модулей

3.2 Реализация интерфейса пользователя

3.3 Разграничение прав доступа пользователей

3.4 Экспорт данных

4 Тестирование и отладка программного обеспечения

4.1 Структурное тестирование

4.2 Функциональное тестирование

5 Инструкция по эксплуатации программного обеспечения

5.1 Установка программного обеспечения

5.2 Инструкция по работе

Заключение

Список использованных источников

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

В настоящем курсовом проекте применяют следующие сокращения и обозначения.

БД – база данных

ОС – операционная система

ПО – программное обеспечение

СУБД – система управления базами данных

API – интерфейс программирования приложения

ASP.NET – активные страницы сервера для .NET

REST API –

**ВВЕДЕНИЕ**

Актуальность разрабатываемого проекта заключается в необходимости автоматизации процесса учёта и обработки заявок на ремонт оборудования и техники. В сервисном центре учёт заявок ведётся вручную или с использованием неунифицированных средств, что приводит к потере данных, дублированию информации и снижению эффективности работы сотрудников. Разработка веб-приложения **«**СмартФикс. Учёт заявок на ремонт**»** позволит централизовать хранение информации, ускорить обработку обращений и повысить прозрачность взаимодействия между клиентами, мастерами и администраторами.

Целью курсового проектирования является разработка комплексного веб-решения для управления заявками на ремонт, обеспечивающего регистрацию, распределение и контроль выполнения ремонтных работ с возможностью онлайн-доступа.

Для достижения поставленной цели требуется решить следующие задачи:

* провести сбор требований целевой аудитории,
* проанализировать информационные источники по предметной области,
* спроектировать архитектуру веб-приложения,
* спроектировать диаграмму вариантов использования приложения,
* выбрать состав программных и технических средств для реализации приложения,
* спроектировать БД,
* спроектировать интерфейс веб-приложения,
* создать БД в выбранной СУБД,
* разработать API для некоторых функций приложения,
* реализовать разграничение прав доступа пользователей,
* разработать интерфейс веб-приложения,
* разработать веб-приложение,
* реализовать работу веб-приложения с сервером БД при помощи REST API,
* выполнить структурное тестирование ПО,
* выполнить функциональное тестирование ПО,
* разработать программную и эксплуатационную документацию.

В результате выполнения поставленных задач будет разработано веб-приложение, обеспечивающее эффективное управление заявками на ремонт и автоматизацию обработки обращений клиентов.

**1 Анализ и разработка требований**

* 1. **Назначение и область применения**

Подсистема **«**СмартФикс. Учёт заявок на ремонт**»** предназначена для автоматизации и упрощения процесса регистрации, обработки и учёта заявок на ремонт оборудования и техники. Система позволяет оптимизировать взаимодействие между сотрудниками сервисной службы и ускорить обработку заявок. Мастера смогут эффективно управлять заявками, отслеживать их и взаимодействовать с клиентами.

* 1. **Постановка задачи**

Необходимо разработать веб-приложение, которое предоставит доступ к следующей функциональности:

- авторизации пользователей,

- добавлению и редактированию заявка на ремонт,

- просмотру информации о заказах,

- формированию отчётов по выполненным и текущим заявкам.

У веб-приложения будут следующие эксплуатационные требования:

- интерфейс должен быть интуитивно понятным

- обязательная авторизация и разграничение прав доступа

- устойчивость к сбоям и сохранение данных при ошибках сети

- время отклика не должно превышать 2 секунды при стандартной нагрузке

Неавторизованный пользователь может просматривать страницу входа, зарегистрироваться.

Клиент может создавать и просматривать свои заявки, отслеживать статусы, редактировать контактные данные.

Мастер видит закреплённые за ним заявки, изменяет статусы и добавляет комментарии по ремонту.

Администратор имеет полный доступ к функциональности.

На рисунке 1 изображена диаграмма вариантов использования приложения.

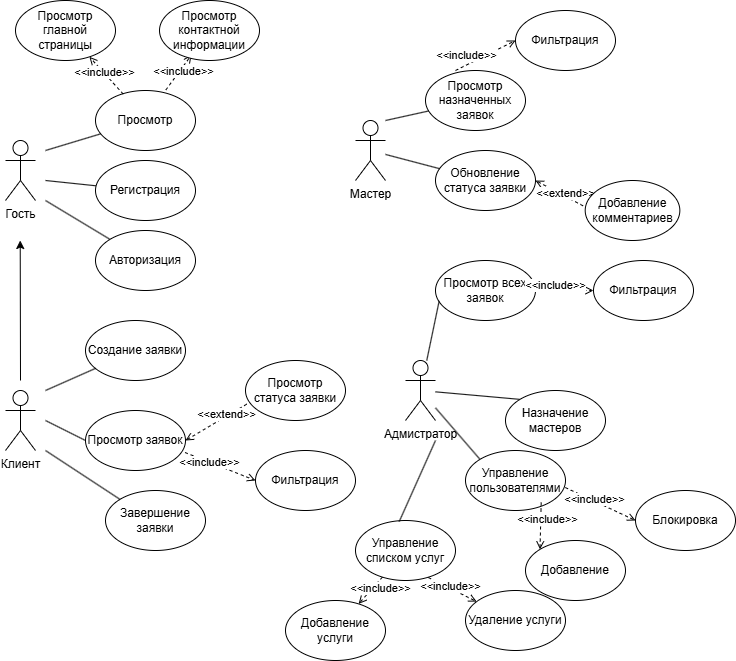


Рисунок 1 – Диаграмма вариантов использования

* 1. **Выбор состава программных и технических средств**

Согласно цели проекта требуется создать подсистему «СмартФикс. Учёт заявок на ремонт» для автоматизации процессов приёма, хранения и обработки заявок на ремонт для учёта заказов.

Разработка подсистемы будет осуществляться на языке C# с использованием фреймворка ASP.NET Core, обеспечивающего создание веб-приложений с чёткой архитектурой и поддержкой MVC-паттерна.

В качестве СУБД выбрана MySQL, отличающаяся стабильностью, простотой в развёртывании, а также наличием большого числа инструментов администрирования и интеграции с .NET-приложениями.

Для контейнеризации и упрощения развёртывания серверной части будет использоваться Docker, обеспечивающий переносимость и изоляцию сервисов. Конфигурация контейнеров будет управляться с помощью docker-compose.

Веб-сервером для публикации и маршрутизации приложения выбран Caddy, предоставляющий встроенную поддержку HTTPS и удобную автоматическую конфигурацию без необходимости ручной настройки SSL-сертификатов.

Разработка будет выполняться в Visual Studio — интегрированной среде разработки, обеспечивающей поддержку ASP.NET Core, работу с Docker и MySQL, а также встроенные средства отладки.  
 Для разработки интерфейса веб-приложения будет использован технологический стек React + Tailwind **CSS**. Данный подход обеспечивает высокую интерактивность пользовательского интерфейса, динамическое обновление данных через REST API и адаптивное отображение на различных устройствах.

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

* ОС Ubuntu версии 18.04 и выше,
* сервер БД: MySQL не ниже 8.0.40,
* процессор с 2 ядрами по 2 ГГц,
* оперативная память объемом 2 ГБ,
* ПО для конфигурирования, управления и администрирования сервера БД: MySQL Workbench последней версии,
* программное обеспечение для работы API: Docker, docker-compose.

Для функционирования системы на стороне клиента достаточны следующие программные и технические средства:

* ОС Windows версей не ниже 10,
* процессор с частотой 1 ГГц или быстрее,
* оперативная память в объеме 1 ГБ и выше,
* свободное место в хранилище 40 МБ,
* постоянное интернет-подключение.

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Бек, К. Экстремальное программирование: разработка через тестирование. – Санкт-Петербург : Питер, 2021. – 224 с. – URL: https://ibooks.ru/bookshelf/376974/reading (дата обращения: 05.11.2024). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.
2. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. – Москва : ФОРУМ : ИНФРА-М, 2023. – 400 с. – URL: https://znanium.com/catalog/product/1895679 (дата обращения: 06.11.2024). – Режим доступа: по подписке. – Текст : электронный.
3. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для проектирования информационных систем : учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. — Москва : ФОРУМ : ИНФРА-М, 2023. — 368 с. - URL: https://znanium.com/catalog/product/1912454 (дата обращения: 13.11.2024). – Режим доступа: по подписке. — Текст : электронный.
4. Тидвелл, Д. Разработка интерфейсов. Паттерны проектирования. 3-е изд. – Санкт-Петербург : Питер, 2022. – 560 с. – Текст : электронный. – URL: <https://ibooks.ru/bookshelf/386796/reading> (дата обращения: \_\_.11.2025). – Режим доступа: для зарегистрир. пользователей.
5. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие. — Москва : КУРС : ИНФРА-М, 2024. — 336 с. – URL: <https://znanium.ru/catalog/product/2083407> (дата обращения: \_\_.11.2025). – Режим доступа: по подписке. – Текст : электронный.