СОДЕРЖАНИЕ

[Введение 4](#_Toc200620062)

[1 Постановка задачи 5](#_Toc200620063)

[2 Описание предметной области 7](#_Toc200620064)

[3 Моделирование проектируемой системы 10](#_Toc200620065)

[4 Разработка функциональных требований к программе 12](#_Toc200620066)

[5 Обоснование выбора средств реализации 15](#_Toc200620067)

[6 Структура программы 17](#_Toc200620068)

# Введение

Актуальность данной работы обусловлена необходимостью автоматизации и оптимизации ключевых бизнес-процессов строительной компании, таких как управление проектами, взаимодействие с клиентами и контроль за выполнением работ. В современных условиях цифровизации отсутствие эффективного программного обеспечения для учета объектов, заявок и ресурсов приводит к потере времени, ошибкам в документации и снижению качества обслуживания клиентов. Разработка специализированного веб-приложения позволит систематизировать данные, ускорить обработку запросов и повысить прозрачность работы компании.

Цель работы — создание веб-приложения для строительной компании, обеспечивающего:

* удобное представление информации о проектах (построенных и строящихся);
* автоматизацию обработки заявок от клиентов;
* управление данными через административную панель.

Задачи:

* описать предметную область;
* промоделировать проектируемую систему;
* разработать функциональные требования к программе;
* обосновать выбор средств реализации программной системы;
* разработать архитектуру программной системы;
* реализовать программную систему согласно выбранным средствам реализации;
* выполнить тестирование и отладку программы;
* составить отчёт о проделанной работе.

# Постановка задачи

Назначение разрабатываемой программы − создание специализированного веб-приложения для автоматизации ключевых процессов строительной компании, включая управление проектами, взаимодействие с клиентами и контроль выполнения работ. Программа должна обеспечить эффективный учёт объектов, обработку заявок и предоставление актуальной информации всем участникам процесса.

Для достижения поставленной цели необходимо решить задачи:

1. Описание предметной области:

* провести анализ текущих бизнес-процессов компании;
* выявить ключевые проблемы и потребности в автоматизации;
* составить подробное описание предметной области;
* разработать модель "AS IS" в нотации IDEF0.

1. Моделирование проектируемой системы:

* создать Use Case диаграмму для всех категорий пользователей;
* разработать модель "TO BE" в нотации IDEF0.

1. Разработка функциональных требований:

* определить требования к основным модулям системы;
* управление проектами;
* обработка заявок;
* сформулировать требования к UI и UX;
* определить требования к безопасности и производительности.

1. Обоснование выбора средств реализации:

* провести сравнительный анализ технологических решений;
* выбрать и обосновать языки программирования;
* выбрать и обосновать фреймворки;
* выбрать и обосновать систему управления базами данных;
* составить таблицу сравнения альтернативных решений.

1. Разработка архитектуры системы:

* разработать структуры программы;
* разработать схему данных;
* разработать схему алгоритмов основной программы и подпрограммы.

1. Реализация программной системы:

настроить среду разработки и необходимые инструменты;

* реализовать пользовательский интерфейс;
* реализовать бизнес-логику приложения;
* реализовать механизмы работы с базой данных;
* реализовать систему аутентификации и авторизации;
* создать административную панель управления;
* ограничить доступ в зависимости от роли пользователя в системе.

1. Тестирование и отладка:

* разработать план тестирования;
* провести модульное тестирование;
* провести интеграционное тестирование;
* провести тестирование пользовательского интерфейса;
* провести нагрузочное тестирование;
* выполнить отладку и оптимизацию кода;
* исправить выявленные ошибки.

1. Документирование и отчетность:

* составить техническую документацию;
* описать API;
* подготовить отчет по практике;
* создать презентационные материалы.

# Описание предметной области

Современные строительные компании функционируют в условиях крайне высокой конкуренции на рынке и постоянно ужесточающихся требований как к срокам выполнения работ, так и к их качеству. В отрасли, где календарные графики измеряются буквально днями, а задержки влекут значительные финансовые потери, эффективность каждого бизнес-процесса становится критически важной. Однако парадоксальным образом многие технологические процессы в строительстве до сих пор организованы по устаревшим принципам, с преобладанием бумажного документооборота, разрозненными системами учета и рутинными операциями, требующими постоянного человеческого контроля. Такая организация неизбежно приводит к значительным потерям как временных, так и материальных ресурсов, снижая общую рентабельность проектов. В этих условиях автоматизация ключевых операций перестает быть просто конкурентным преимуществом - она превращается в стратегическую необходимость для любого бизнеса, стремящегося не просто выжить, но и развиваться в современных рыночных реалиях.

1. Актуальность:

Строительные организации ежедневно сталкиваются с рядом проблем в управлении проектами и взаимодействии с клиентами:

* информация о проектах хранится в разных форматах и местах (Excel, бумажные документы, локальные базы);
* обработка клиентских запросов занимает много времени из-за отсутствия единой системы;
* сложно оперативно получить актуальную информацию о статусе строительных объектов;
* частые ошибки при ручном вводе данных и составлении отчетов.

Эти проблемы существенно снижают эффективность работы компании, увеличивают сроки реализации проектов и ухудшают качество обслуживания клиентов. Разрабатываемая система призвана решить эти вопросы за счет комплексной автоматизации основных бизнес-процессов.

1. Цели и задачи разрабатываемой системы

Цель проекта – создание веб-приложения, которое обеспечит:

* централизованный учет строительных объектов и заявок клиентов;
* автоматизацию обработки запросов;
* удобный инструмент для контроля выполнения работ;
* личный кабинет для клиентов;
* прозрачность данных для руководства и клиентов.

1. Основные задачи системы:

* ведение базы данных объектов (строящиеся, завершенные);
* контроль этапов строительства (графики, сроки, ответственные);
* учет использованных материалов и затрат;
* единая система приема и распределения запросов;
* возможность прикрепления документов (договоры, сметы, фотоотчеты);
* разграничение прав доступа (администраторы, строители, клиенты).

1. Основные модули системы:

Модуль управления проектами:

* каталог объектов с фильтрами (статус, локация, тип здания).

Модуль работы с заявками:

* личный кабинет клиента для подачи запросов;
* административный модуль;
* управление пользователями (роли: администратор, сотрудник, клиент).

1. Преимущества внедрения системы:

* сокращение времени на рутинные операции;
* ускорение обработки заявок за счет единой системы учета;
* повышение точности данных;
* исключение ошибок, связанных с ручным вводом информации;
* централизованное хранение документов;
* улучшение взаимодействия с клиентами;
* прозрачность статусов заказов через личный кабинет;
* оптимизация управления ресурсами.

1. Технические аспекты реализации:

* клиент-серверная архитектура:
* серверная часть: язык: PHP;
* базовый REST API для интеграции;
* система авторизации с ролевой моделью;
* основная схема базы данных;
* база данных: MySQL (простая и быстрая в настройке);
* API: RESTful-подход с обработкой GET/POST-запросов;
* аутентификация: сессии PHP + базовая HTTP-авторизация;
* клиентская часть: чистый HTML, CSS, JavaScript (без фреймворков);
* UI: кастомные компоненты с адаптивным дизайном;
* взаимодействие с сервером: Fetch API для AJAX-запросов;
* безопасность: шифрование данных (HTTPS, защита персональных данных).

Данная система разрабатывается с учетом самых современных требований к корпоративным веб-приложениям и обеспечит строительной компании плавный, но кардинальный переход на качественно новый уровень как в управлении проектами, так и в построении клиентских отношений. В основе решения лежит продуманная модульная архитектура, сочетающая в себе простоту внедрения с широкими возможностями для дальнейшего масштабирования. Использование проверенных, надежных технологий в разработке (таких как PHP 8.2, MySQL 8, современные стандарты JavaScript) позволяет гарантировать стабильную, бесперебойную работу системы даже при минимальных требованиях к ИТ-инфраструктуре компании. При этом система проектируется с расчетом на постепенное внедрение - от базового функционала к комплексной автоматизации всех бизнес-процессов, что делает ее доступной для строительных компаний любого масштаба.

На рисунке 2.1 представлена модель в нотации IDEF0 «AS IS».

Изображение выглядит как текст, Параллельный, линия, снимок экрана

Автоматически созданное описаниеРисунок 2.1 – Модель в нотации IDEF0 «AS IS»

# 3 Моделирование проектируемой системы

Use Case Diagram (диаграмма вариантов использования) – это тип диаграммы в UML (Unified Modeling Language), который используется для описания функциональных требований системы с точки зрения пользователей (или других внешних систем). Она показывает, что система должна делать, а не как она это делает. Диаграмма вариантов использования фокусируется на взаимодействии между пользователями (актерами) и системой для достижения определенных целей.

Основные элементы Use Case Diagram:

* актер: представляет собой пользователя (человека, другую систему или устройство), который взаимодействует с системой. Актер находится за пределами системы и инициирует варианты использования. Обозначается символом человечка;
* вариант использования: представляет собой конкретное действие, которое актер может выполнить с помощью системы для достижения определенной цели. Обозначается овалом;
* система: представляет собой границы системы, то есть определяет, что входит в разрабатываемую систему, а что находится за ее пределами. Обозначается прямоугольником;
* ассоциация: представляет собой связь между актером и вариантом использования. Указывает, что актер участвует в данном варианте использования. Обозначается линией;
* включение: отношение между вариантами использования, указывающее, что один вариант использования всегда включает в себя другой. Используется для выделения общего функционала. Обозначается пунктирной линией со стрелкой, направленной к включаемому варианту использования, и надписью “<include>”;
* расширение: отношение между вариантами использования, указывающее, что один вариант использования может быть расширен другим при определенных условиях. Используется для добавления опционального функционала. Обозначается пунктирной линией со стрелкой, направленной к расширяемому варианту использования, и надписью “<extend>”;
* обобщение: отношение между актерами или вариантами использования, указывающее, что один элемент является более общим, а другой – более специализированным. Обозначается линией с не закрашенным треугольником, направленным к общему элементу.

Назначение Use Case Diagram:

* определение функциональных требований: диаграмма вариантов использования помогает выявить все необходимые функции системы с точки зрения пользователей;
* планирование разработки: она позволяет определить приоритеты разработки на основе важности и частоты использования различных вариантов использования;
* коммуникация с заказчиком: Диаграмма служит наглядным средством общения с заказчиком для согласования функциональных требований;
* тестирование системы: на основе вариантов использования можно разрабатывать тестовые сценарии для проверки работоспособности системы.

На рисунке 3.1 представлена диаграмма вариантов использования.

Изображение выглядит как текст, диаграмма, линия, План

Автоматически созданное описание

Рисунок 3.1 – Диаграмма вариантов использования

# Разработка функциональных требований к программе

1. Назначение задачи

**Функциональное назначение:** программное обеспечение "Putevi" предназначено для автоматизации процессов управления строительными проектами, клиентскими заявками и отзывами в строительной компании. Система обеспечивает:

* учёт и управление проектами (планирование, исполнение, контроль);
* обработку клиентских запросов (консультации, техподдержка, услуги);
* управление заявками на трудоустройство;
* сбор и модерацию отзывов о завершенных проектах;
* разграничение прав доступа для разных категорий пользователей.

**Эксплуатационное назначение:**

* периодичность использования: ежедневно в рабочее время.

Основные пользователи:

* администраторы - круглосуточный доступ для поддержки системы;
* менеджеры проектов - ежедневная работа с проектами и заявками;
* клиенты - доступ по мере необходимости для подачи запросов и отзывов.

Результаты работы будут использоваться:

* руководством компании - для анализа эффективности;
* менеджерами - для оперативного управления проектами;
* HR-отделом - для обработки заявок на трудоустройство;
* маркетингом - для анализа отзывов клиентов.

2. Требования к программе

**Требования к функциональным характеристикам:**

* управление пользователями (crud операции);
* создание и отслеживание проектов (статусы, сроки, бюджет);
* обработка заявок (консультации, услуги, трудоустройство);
* модерация отзывов;
* генерация отчетов.

Входные данные:

* данные пользователей;
* заявки клиентов;
* требования к проекту;
* финансы.

Выходные данные:

* реализованные проекты;
* обработанные заявки;
* данные пользователей;
* состояние проектов.

Временные характеристики:

* время отклика интерфейса <1 сек;
* формирование отчетов <5 сек;
* обработка сложных запросов <10 сек.

**Требования к надежности:**

Устойчивое функционирование:

* доступность 99.5% в рабочее время;
* обеспечение целостности данных.

Контроль данных:

* валидация входных данных;
* проверка целостности при операциях.

Восстановление после сбоев:

* максимальное время восстановления - 15 минут;
* ежедневное резервное копирование.

**Требования к техническим средствам:**

Минимальная конфигурация сервера:

* CPU: 4 ядра;
* RAM: 8 GB;
* HDD: 100 GB (SSD рекомендовано);
* ОС: Linux/Windows Server.

Клиентские требования:

* браузер: Chrome, Firefox, Edge последних версий;
* разрешение экрана: от 1280x720**4.**

**Требования к совместимости:**

Программная совместимость:

* backend: php 8.0+;
* frontend: html5, css3, es6+;
* субд: mysql 8.0+.

Информационная совместимость:

* форматы данных: json, csv (для экспорта);
* api: restful.

Защита информации:

* https обязателен;
* ролевая модель доступа;
* шифрование паролей (bcrypt);
* защита от sql-инъекций.

3. Требования к программной документации

**Самодокументирование кода:**

* обязательные комментарии для всех модулей и сложных функций;
* описание входных/выходных параметров;
* jsdoc/phpdoc для api.

Техническая документация:

* схема базы данных;
* api спецификация;
* руководство по установке.

Пользовательская документация:

* руководство администратора;
* руководство сотрудника;
* инструкция для клиентов.

Эксплуатационная документация:

* план резервного копирования;
* процедуры восстановления;
* чек-лист мониторинга.

**Требования к актуализации:**

* документация должна обновляться при каждом значительном изменении функционала;
* версионирование документации;
* журнал изменений.

# 5 Обоснование выбора средств реализации

основной концепцией разработки является создание веб-приложения для управления строительными проектами, клиентскими заявками и отзывами. выбор веб-технологий обоснован необходимостью обеспечения кроссплатформенности, простоты масштабирования и доступности системы с любых устройств через браузер.

**Выбор инструментов:**

* + 1. **Backend: PHP 8.0+**

Выбор языка PHP обусловлен следующими факторами:

* оптимальная производительность: PHP предоставляет значительные улучшения производительности (компилятор) и низкое потребление ресурсов;
* широкая распространенность: 78% всех веб-сайтов используют PHP, что обеспечивает доступность хостингов и специалистов;
* богатая экосистема: наличие проверенных фреймворков и библиотек для быстрой разработки бизнес-логики;
* безопасность: встроенные механизмы защиты от распространенных уязвимостей веб-приложений;
* простота интеграции: легкость подключения к mysql и работы с api.
  + 1. **Frontend: html5, css3, es6+**

Выбор современных веб-технологий обусловлен:

* кроссплатформенностью: возможность работы на любых устройствах без установки дополнительного ПО;
* интерактивностью: современный javascript (es6+) позволяет создавать динамические интерфейсы;
* адаптивностью: css3 обеспечивает корректное отображение на всех типах устройств;
* производительностью: легковесность и быстрая загрузка страниц;
* совместимостью: поддержка всеми современными браузерами.
  + 1. **СУБД: mysql 8.0+**

Выбор системы управления базами данных обусловлен:

* производительность: улучшенные механизмы индексирования и кэширования в mysql 8;
* надежность: механизмы транзакций и репликации данных;
* масштабируемость: поддержка больших объемов данных и высокой нагрузки;
* безопасность: расширенные механизмы аутентификации и шифрования данных;
* совместимость: оптимальная интеграция с PHP через pdo и mysqli.
  + 1. **Редактор кода: visual studio code**

Выбор среды разработки обусловлен:

* мощная функциональность: поддержка всех выбранных технологий через расширения;
* интеграция с инструментами: встроенный git, docker, ssh клиент;
* производительность: быстрая работа даже с большими проектами;
* кроссплатформенность: доступность на windows, linux и macos;
* экосистема расширений: богатый выбор плагинов для PHP, javascript, mysql.

**Преимущества выбранного стека технологий:**

* быстрота разработки: готовые решения и шаблоны для типовых задач;
* экономическая эффективность: открытое по и доступные специалисты;
* надежность: проверенные временем технологии с хорошей документацией;
* масштабируемость: возможность наращивания функционала без переписывания кода;
* безопасность: встроенные механизмы защиты на всех уровнях.

Выбранный набор технологий оптимально соответствует требованиям проекта по производительности, безопасности и удобству разработки, обеспечивая при этом разумный баланс между скоростью реализации и качеством конечного продукта.

# 6 Структура программы