



МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖИ РЕСПУБЛИКИ
КРЫМ

Государственное бюджетное образовательное учреждение высшего
образования

Республики Крым

«Крымский инженерно–педагогический университет имени Февзи
Якубова»

(ГБОУВО РК КИПУ имени Февзи Якубова)

Факультет экономики, менеджмента и информационных технологий
Кафедра прикладной информатики

Направление подготовки 09.04.03 Прикладная информатика

Магистерская программа

«Прикладная информатика в информационной сфере»

МАГИСТЕРСКАЯ РАБОТА

Проектирование и разработка веб–платформы для управления IT–
проектами

Студента 2 курса

группы М–И–22

очной формы обучения

Абибулаева Ленура

подпись

Научный руководитель:

к.техн.н. Абдурайимов Л. Н.

подпись

Работа допущена к защите

зав. кафедрой прикладной информатики,
д.пед.н., проф. _____ Сейдаметова З.С.
«_____» _____ 2024 г.

Симферополь – 2024

Аннотация

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
ГЛАВА 1. ОСОБЕННОСТИ СОЗДАНИЯ ВЕБ–ПЛАТФОРМ ДЛЯ УПРАВЛЕНИЯ ИТ–ПРОЕКТАМИ.....	11
1.1 Особенности веб–платформ для управления ИТ–проектами..	11
1.2 Сравнительный анализ веб–платформ для управления ИТ– проектами.....	12
1.3 Теоретические основы управления ИТ–проектами	15
1.4 Основные требования к разрабатываемой системе.....	18
Вывод к первой главе	22
ГЛАВА 2. ПРОЕКТИРОВАНИЕ И ВЫБОР СРЕДСТВ РАЗРАБОТКИ ВЕБ–ПЛАТФОРМЫ	24
2.1. Основные этапы разработки веб–платформы для управления ИТ–проектами	24
2.2. Создание абстрактной модели веб–платформы	26
2.3. Скетч–разработка интерфейса и дизайна веб–платфомы	31
2.4. Выбор технических средств и программного инструментария для разработки системы	32
Выводы ко второй главе	34
ГЛАВА 3. РАЗРАБОТКА ВЕБ–ПЛАТФОРМЫ ДЛЯ УПРАВЛЕНИЯ ИТ–ПРОЕКТАМИ	36
3.1. Проектирование и разработка базы данных.....	36
3.2. Структура разрабатываемой системы и пользовательский интерфейс	37
Вывод к третьей главе	40
Заключение.....	42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	44

ВВЕДЕНИЕ

Актуальность темы. Тема "Проектирование и разработка веб-платформы для управления IT-проектами" имеет большую актуальность в настоящее время, так как в условиях развития информационных технологий и увеличения количества IT-проектов возникает необходимость в эффективной управляемости проектами.

Веб-платформа для управления IT-проектами представляет собой программный инструмент, который позволяет скоординировать работу участников проекта, управлять ресурсами, бюджетом, сроками выполнения задач, а также контролировать качество продукта. Веб-платформы также могут предоставлять возможность для мониторинга процесса выполнения проекта и генерирования отчетов о проделанной работе.

Таким образом, данная тема дипломной работы является актуальной, так как веб-платформы для управления IT-проектами могут значительно повысить эффективность процесса управления и помочь компаниям сохранить конкурентоспособность на рынке. Кроме того, веб-платформы могут сэкономить время и снизить затраты на управление проектами, что также является важным аспектом для компаний.

Также стоит отметить, что в настоящее время существует большое количество программных продуктов для управления IT-проектами, однако каждая компания имеет свои уникальные потребности и требования, поэтому создание индивидуальной веб-платформы может стать оптимальным решением для решения конкретных задач.

Примеры существующих платформ и их недостатки:

JIRA:

Преимущества: Широко используется в индустрии, обладает мощными возможностями для управления задачами и багами, поддерживает интеграцию с другими инструментами Atlassian.

Недостатки: Сложность в настройке и использовании для новых пользователей, высокая стоимость лицензии, особенно для небольших компаний, перегруженность интерфейса из-за обилия функций.

Trello:

Преимущества: Простота в использовании, интуитивно понятный интерфейс, визуальное представление задач в виде карточек на доске.

Недостатки: Ограниченные возможности для сложных проектов, отсутствие продвинутых функций для отчетности и аналитики, сложности с масштабированием для больших команд.

Asana:

Преимущества: Удобный интерфейс, поддержка разнообразных видов представления данных (списки, доски, календари), хорошая интеграция с другими инструментами.

Недостатки: Ограничения бесплатной версии, недостаточная гибкость в настройке сложных рабочих процессов, высокие цены на премиум-функции.

Microsoft Project:

Преимущества: Глубокая функциональность для планирования и управления проектами, интеграция с экосистемой Microsoft.

Недостатки: Сложность в освоении для пользователей без специализированного обучения, высокая стоимость лицензий, необходимость установки на локальные машины (хотя есть и онлайн-версия).

Таким образом, несмотря на наличие на рынке различных платформ для управления IT-проектами, многие из них имеют свои недостатки, которые могут снижать их эффективность в определенных сценариях. Это подчеркивает необходимость разработки новой веб-платформы, которая бы учитывала современные требования и предоставляла более гибкие, доступные и интуитивно понятные решения для управления IT-проектами.

Степень научной разработанности проблемы.

Целью магистерской работы является проектирование и разработка веб–платформы для управления IT–проектами, направленной на оптимизацию работы и синергию команды разработчиков программного обеспечения.

Для осуществления поставленной цели, были выдвинут следующие задачи:

- ✓ Изучить специфику работы команды IT–разработчиков;
- ✓ Проанализировать существующие веб–платформы для управления IT–проектами и выявить их достоинства и недостатки;
- ✓ Спроектировать модель работы веб–платформы для управления IT–проектами с помощью UML–диаграмм;
- ✓ Разработать дизайн интерфейса системы с помощью графического редактора Figma;
- ✓ Выбрать рациональные программные средства разработки требуемой платформы;
- ✓ Спроектировать базу данных для работы веб–платформы;

Объект исследования – является веб–платформа для управления IT–проектами.

Предмет исследования – является процесс разработки веб–платформы для управления IT–проектами.

Научная новизна и практическая значимость работы. Научная новизна магистерской квалификационной работы заключается в разработке и предложении новой архитектуры веб–платформы для управления IT–проектами, которая оптимизирует процессы планирования, выполнения и контроля IT–проектов. Данная работа включает в себя следующие элементы научной новизны:

Интеграция современных технологий и инструментов: В работе используется комбинация передовых технологий, таких как Python, Flask, MySQL и phpMyAdmin, что позволяет создать высокоэффективную и масштабируемую платформу.

Создание системы уведомления в Telegram: Пользователям будут приходить сообщения в Telegram. В данных сообщениях будет информация о получении новой задачи, о получении правок от тимлида и тп.

Разработка уникальной архитектуры системы: Проект предлагает уникальную архитектурную модель, включающую микросервисный подход, который обеспечивает гибкость, модульность и устойчивость системы к изменениям.

Практическая значимость магистерской квалификационной работы заключается в возможности непосредственного применения разработанной веб-платформы для управления IT-проектами в реальных условиях. Основные аспекты практической значимости включают:

Улучшение процессов управления проектами: Внедрение разработанной платформы позволяет компаниям и организациям оптимизировать процессы планирования, выполнения и контроля IT-проектов, что ведет к снижению временных и финансовых затрат.

Повышение продуктивности команд: Платформа предоставляет удобные инструменты для совместной работы, управления задачами и отслеживания прогресса, что способствует повышению продуктивности и эффективности работы команд.

Гибкость и масштабируемость: Разработанная архитектура позволяет легко адаптировать платформу под конкретные требования и масштабировать её в зависимости от роста проекта или организации.

Таким образом, данная магистерская квалификационная работа имеет как научную, так и практическую ценность, внося вклад в развитие теоретических основ и предлагая готовое решение для практического применения в сфере управления IT-проектами.

Публикации и апробации. Основные положения исследования опубликованы и представлены на:

– Межвузовская научно–практическая студенческая конференция «Информация как социальная ценность: правовые, технические и программные методы защиты», 29–30 ноября 2022 г.

– Межвузовская научно–практическая студенческая конференция «Информация как социальная ценность: правовые, технические и программные методы защиты», 7–8 декабря 2023 г. Тема доклада: «Особенности разработки системы управления ИТ–проектами»

– Научно–практическая конференция для студентов и молодых ученых «Молодая наука», 8 ноября 2023г.

– Опубликовано в сборнике РИНЦ Крымского инженерно–педагогического университета имени Февзи Якубова «Информационно–компьютерные технологии в экономике, образовании и социальной сфере». (Тема статьи: «Инновационные подходы к проектированию и разработке веб–платформы для эффективного управления ИТ– проектами») (Симферополь, 2023).

Структура и объем работы.

В введении

В первой главе

Во второй главе

Третья глава

В заключении

ГЛАВА 1. ОСОБЕННОСТИ СОЗДАНИЯ ВЕБ–ПЛАТФОРМ ДЛЯ УПРАВЛЕНИЯ ИТ–ПРОЕКТАМИ.

1.1 Особенности веб–платформ для управления ИТ–проектами.

Веб–платформа для управления ИТ–проектами — это интегрированная онлайн–система, предоставляющая широкий спектр инструментов и функций для эффективной разработки, координации и управления проектами в области информационных технологий. Такие платформы созданы с учетом потребностей профессионалов в управлении проектами, разработчиков, менеджеров и других участников команды, позволяя им взаимодействовать и вести проекты в реальном времени. Далее перечислено более подробное описание ключевых характеристик веб–платформ для управления ИТ–проектами:

Управление задачами: Создание, назначение, отслеживание и приоритизация задач. Это позволяет команде четко определить свои обязанности и следить за ходом выполнения задач [1].

Управление проектами: Создание проектов, определение их целей, планирование, анализ рисков и мониторинг выполнения. Платформа предоставляет гибкость в настройке структуры проекта в соответствии с требованиями конкретной задачи.

Коллаборация и коммуникация: Интегрированные средства общения, такие как чаты, форумы и комментарии к задачам, содействуют активной коммуникации внутри команды, что способствует обмену идеями и решением проблем.

Планирование и графики: Создание графиков проекта с определением зависимостей между задачами, установка сроков и ресурсов. Это помогает лучше предвидеть и управлять временными и ресурсными аспектами проекта.

Управление рисками: Идентификация, анализ и управление рисками проекта. Это включает в себя оценку вероятности событий, которые могут повлиять на успех проекта, и разработку стратегий их управления.

Мониторинг прогресса: Отслеживание выполнения задач и проектов в реальном времени, что позволяет оперативно реагировать на изменения ситуации и корректировать план при необходимости.

Документирование и хранение данных: Централизованное хранение и управление документацией, файлами и данными проекта, обеспечивающее доступность и актуальность информации для всех участников.

Аналитика и отчетность: Сбор и анализ данных, предоставление отчетов по различным аспектам проекта, что позволяет оценить эффективность работы и выявить возможности для улучшений.

Интеграция с другими инструментами: Возможность интеграции с другими популярными инструментами, такими как системы контроля версий, инструменты для обмена файлами, средства связи и другие, для удобства и эффективности работы.

Безопасность: Обеспечение безопасности данных средствами шифрования, управления правами доступа и другими средствами, гарантирующими конфиденциальность и целостность информации [2].

1.2 Сравнительный анализ веб–платформ для управления IT–проектами.

Существует множество веб–платформ для управления IT–проектами, каждая из которых имеет свои уникальные характеристики, преимущества и недостатки. Далее представлен сравнительный анализ нескольких популярных веб–платформ: Jira, Trello и Asana.

Jira – это инструмент управления проектами, который помогает оптимизировать работу команды.

Управление задачами и проектами:

Предоставляет мощные инструменты для создания и управления задачами, а также для организации проектов в соответствии с различными методологиями (Scrum, Kanban, etc.).

Гибкость и настройка:

Jira известна своей гибкостью и возможностью настройки. Пользователи могут адаптировать рабочие процессы под нужды своей команды, используя разнообразные плагины и настройки.

Интеграция:

Обеспечивает интеграцию с другими инструментами разработки, такими как Bitbucket, Confluence и другими популярными инструментами.

Сложность интерфейса:

Интерфейс Jira может быть сложным для новых пользователей из-за множества функций и настроек.

Тарификация:

Имеет платные тарифы, предоставляя дополнительные функции и возможности в зависимости от уровня подписки [3].

Trello:

Интуитивный интерфейс:

Известен своим простым и интуитивно понятным интерфейсом, что делает его идеальным для небольших команд и проектов.

Доски и карточки:

Работа с досками и карточками для управления задачами, их перемещение между столбцами и организация проекта.

Бесплатная версия:

Предоставляет бесплатную версию с основными функциональностями, что делает его доступным для стартапов и небольших команд.

Ограниченность функционала:

Может быть ограничен в возможностях для крупных и сложных проектов. Отсутствие некоторых продвинутых функций.

Преимущества:

Простой и интуитивно понятный интерфейс, что делает его идеальным для небольших команд и проектов.

Гибкая структура досок и карточек для управления задачами.

Бесплатная версия с базовым функционалом.

Недостатки:

Относительная ограниченность в возможностях, что может быть недостаточно для крупных и сложных проектов.

Отсутствие возможности для продвинутого отслеживания времени и аналитики [4].

Asana:

Интуитивность и гибкость:

Обеспечивает интуитивный интерфейс и гибкость в управлении задачами, проектами и командами.

Поддержка методологий:

Поддерживает различные методологии управления проектами и предоставляет инструменты для их эффективного внедрения.

Интеграции:

Имеет интеграции с различными сторонними приложениями и инструментами, что облегчает взаимодействие с другими средствами разработки.

Аналитика:

Обеспечивает базовые средства аналитики и отчетности, но, возможно, не такие мощные, как у Jira.

Тарификация:

Предоставляет различные тарифные планы с разным уровнем функциональности.

Преимущества:

Интуитивный интерфейс, подходящий как для новичков, так и для опытных пользователей.

Гибкость в управлении задачами, проектами и командами.

Поддержка широкого спектра методологий управления проектами.

Недостатки:

Некоторые пользователи отмечают, что для использования некоторых продвинутых функций требуется временное приспособление.

Возможно, не такие мощные инструменты аналитики и отчетности, как у Jira [5].

Каждая из этих платформ имеет свои преимущества и ограничения, и выбор зависит от уникальных требований вашей команды и характера проектов. Относительно Trello, ориентированного на простоту, до Jira, предоставляющего мощные инструменты для сложных проектов, команды могут выбирать в соответствии с их конкретными потребностями. Asana предлагает баланс между гибкостью и интуитивностью, что делает его привлекательным для разнообразных команд.

1.3 Теоретические основы управления IT–проектами

Управление IT–проектами представляет собой сложный процесс, включающий планирование, выполнение и контроль всех аспектов проекта с целью достижения его целей в установленные сроки, в рамках бюджета и с требуемым качеством. Для эффективного управления IT–проектами разработаны различные методологии и принципы, которые обеспечивают структурированный подход к выполнению задач и управлению ресурсами. Рассмотрим основные из них:

Waterfall (Каскадная модель):

Описание: Каскадная модель является одной из самых старых и традиционных методологий управления проектами. Она предполагает последовательное выполнение фаз проекта: от анализа требований до разработки, тестирования, внедрения и сопровождения.

Преимущества: Четко структурированный процесс, который легко планировать и управлять. Подходит для проектов с четко определенными требованиями и неизменными условиями.

Недостатки: Недостаточная гибкость, высокая вероятность накопления ошибок и изменений к концу проекта, что может привести к значительным затратам на их исправление [6].

Critical Path Method (CPM) и Program Evaluation and Review Technique (PERT):

Описание: CPM и PERT – это методики, используемые для планирования и управления проектами с большим количеством взаимозависимых задач. CPM фокусируется на критическом пути, определяя последовательность задач, которые напрямую влияют на сроки завершения проекта. PERT использует вероятностные оценки времени выполнения задач.

Преимущества: Позволяют точно определить временные рамки и оптимизировать график выполнения задач.

Недостатки: Могут быть сложными в применении для очень крупных или динамично изменяющихся проектов [7].

Scrum: Scrum – это фреймворк для управления проектами, основанный на итеративном и инкрементальном подходе. Проект разбивается на спринты (короткие временные промежутки, обычно от 1 до 4 недель), в течение которых команда работает над выполнением задач из бэклога продукта.

Преимущества: Высокая адаптивность к изменениям, регулярная обратная связь, улучшенная командная работа и прозрачность процессов.

Недостатки: Требуется высокий уровень самоорганизации команды, может быть сложным для крупных проектов без соответствующей настройки.

Kanban: Kanban – это метод управления проектами, который фокусируется на визуализации рабочих процессов и управлении потоком задач. Основным инструментом является доска Kanban, разделенная на столбцы, отражающие стадии выполнения задач.

Преимущества: Простота в использовании, улучшение визуализации и прозрачности процессов, возможность плавного внедрения изменений.

Недостатки: Менее структурированный подход по сравнению с Scrum, может не предоставить достаточного уровня планирования для сложных проектов [8].

Lean: Lean методология фокусируется на минимизации потерь и оптимизации процессов. В IT-проектах Lean использует принципы устранения лишних шагов, улучшения качества и быстрой доставки ценности клиенту.

Преимущества: Уменьшение затрат и времени на выполнение задач, повышение качества продукта, улучшение удовлетворенности клиентов.

Недостатки: Требуется культурных изменений в организации, может быть сложным для внедрения в традиционных компаниях.

Гибридные методологии

Scrumban: Scrumban – это гибридная методология, объединяющая элементы Scrum и Kanban. Она использует итерации и спринты из Scrum, а также визуализацию и управление потоком задач из Kanban.

Преимущества: Комбинирует лучшие стороны обоих подходов, позволяя командам адаптироваться к изменениям и улучшать процессы.

Недостатки: Может быть сложно определить оптимальный баланс между Scrum и Kanban для конкретного проекта.

Agile–Waterfall Hybrid:

Описание: Этот подход сочетает в себе элементы каскадной модели и гибких методологий. В рамках такой методологии планирование и начальные фазы выполняются по Waterfall, а разработка и тестирование – по Agile.

Преимущества: Подходит для проектов, требующих четкого начального планирования, но нуждающихся в гибкости на этапах разработки.

Недостатки: Может быть сложно управлять взаимодействием между фазами, требуются опытные менеджеры проектов.

Важные аспекты управления IT-проектами

Управление требованиями:

Процесс сбора, анализа, документирования и управления требованиями проекта. Включает в себя взаимодействие с заинтересованными сторонами для обеспечения понимания и согласования требований.

Управление рисками:

Идентификация, анализ и разработка стратегий для минимизации рисков, связанных с выполнением проекта. Включает в себя разработку плана управления рисками и постоянный мониторинг.

Управление ресурсами:

Эффективное распределение и использование ресурсов, включая персонал, оборудование и финансовые средства. Включает в себя планирование, распределение и мониторинг использования ресурсов.

Управление качеством:

Обеспечение соответствия проекта установленным стандартам качества. Включает в себя процессы планирования качества, контроля и улучшения.

Управление коммуникациями:

Обеспечение эффективного обмена информацией между всеми участниками проекта. Включает в себя планирование, реализацию и мониторинг коммуникационных процессов.

Таким образом, теоретические основы управления IT-проектами включают в себя разнообразные методологии и подходы, которые помогают эффективно организовать и контролировать процесс выполнения проектов, адаптируя их под конкретные условия и потребности [9].

1.4 Основные требования к разрабатываемой системе.

Сбор и анализ требований — один из самых важных этапов в процессе разработки веб-платформы для управления IT-проектами. На этом этапе определяются ожидания и потребности всех заинтересованных сторон, что

позволяет заложить прочную основу для последующих этапов проектирования и разработки.

Сбор требований:

Сбор требований представляет собой процесс выявления и документирования всех потребностей и ожиданий, связанных с будущей системой. Существует несколько методов сбора требований, каждый из которых имеет свои преимущества и подходит для различных ситуаций:

Интервью:

Описание: Проведение бесед с ключевыми заинтересованными сторонами [10].

Преимущества: Позволяет получить глубокое понимание потребностей и ожиданий, выявить скрытые требования.

Недостатки: Требуется значительных временных затрат, возможность субъективного восприятия ответов.

Опросы и анкеты:

Описание: Распространение анкет среди широкой аудитории для сбора информации о требованиях.

Преимущества: Быстрое получение данных от большого числа респондентов, стандартные ответы упрощают анализ.

Недостатки: Ограниченные возможности для глубинного понимания, низкий уровень отклика [11].

Рабочие совещания (workshops):

Описание: Организация совместных встреч с группами заинтересованных сторон для обсуждения требований.

Преимущества: Позволяет выявить требования в режиме реального времени, способствует взаимопониманию и согласованию.

Недостатки: Требуется значительных организационных усилий, возможна доминирующая роль отдельных участников.

Анализ документов:

Описание: Изучение существующих документов, связанных с проектом (бизнес–планы, технические задания, спецификации).

Преимущества: Обеспечивает доступ к официальной информации, помогает выявить явные и скрытые требования.

Недостатки: Ограниченная актуальность данных, возможно недостаточное понимание контекста.

Наблюдение:

Описание: Наблюдение за пользователями в процессе их работы для выявления требований.

Преимущества: Позволяет выявить реальные потребности и проблемы, которые могут не осознаваться пользователями.

Недостатки: Затруднительно применять для распределенных команд, возможны искажения из–за присутствия наблюдателя [12].

Анализ требований

Анализ требований включает систематизацию, уточнение и согласование всех собранных данных. Этот этап необходим для формирования четкого и полного представления о том, что должно быть реализовано в проекте.

Классификация требований:

Функциональные требования: Определяют, какие функции и возможности должна предоставлять система (например, управление задачами, отслеживание времени, генерация отчетов).

Нефункциональные требования: Определяют качество и характеристики системы (например, производительность, безопасность, удобство использования).

Существует несколько этапов обработки требований к ПО:

- Приоритизация требований
- Уточнение и согласование требований
- Документирование требований

- **Валидация требований**

Сбор и анализ требований — это критически важный этап, который закладывает основу для успешной реализации веб–платформы для управления ИТ–проектами. Грамотно проведенный процесс сбора и анализа требований позволяет минимизировать риски, связанные с непониманием ожиданий клиентов, и создать продукт, соответствующий их потребностям и требованиям [13]

Для обоснования требований к разработке веб–платформы для управления ИТ–проектами, был проведен опрос среди студентов кафедры прикладной информатики ГБОУ РК КИПУ им. Февзи Якубова. В опросе приняли участие 50 студентов, которые активно участвуют в различных проектах и используют различные инструменты для их управления. Результаты опроса приведены в приложении 1.

Выводы из опроса.

На основе полученных данных можно сделать следующие выводы:

Высокий интерес к управлению задачами: Большинство студентов считают управление задачами основной функцией, что подчеркивает необходимость создания интуитивно понятного интерфейса для управления задачами и проектами.

Проблемы с текущими инструментами: Основные проблемы включают сложность в использовании, недостаток функционала и перегруженность интерфейса. Это указывает на необходимость создания более простого и функционального решения.

Необходимость интеграции: Студенты подчеркивают важность интеграции с другими сервисами и инструментами, что позволит улучшить рабочий процесс и взаимодействие.

Важность нефункциональных требований: Удобство использования, безопасность данных и высокая производительность являются ключевыми аспектами, которые должны быть учтены при разработке платформы.

Эти результаты помогут сформировать четкое понимание требований и ожиданий пользователей, что является критически важным для успешной разработки веб–платформы для управления IT–проектами.

Вывод к первой главе

В первой главе магистерской квалификационной работы рассмотрены ключевые аспекты, связанные с созданием веб–платформ для управления IT–проектами. Проведенный анализ и исследование позволили выявить следующие важные выводы:

Веб–платформы для управления IT–проектами обладают рядом уникальных характеристик, таких как возможность доступа из любого места и в любое время, интеграция с различными инструментами и сервисами, поддержка совместной работы команд, а также высокая степень масштабируемости и гибкости. Эти особенности делают их незаменимыми инструментами для современных IT–команд, позволяя эффективно планировать, отслеживать и управлять проектами.

В результате сравнительного анализа различных существующих веб–платформ, таких как Jira, Trello, Asana и других, были выявлены их сильные и слабые стороны. Анализ показал, что каждая из платформ обладает уникальными функциями и возможностями, которые могут быть полезны в разных контекстах. Однако, ни одна из существующих платформ не удовлетворяет полностью всем требованиям, предъявляемым к современным системам управления IT–проектами, что подчеркивает необходимость разработки новых решений, учитывающих современные тенденции и потребности пользователей.

Исследование теоретических основ управления IT–проектами позволило понять основные подходы и методы, используемые в данной области. Это включает в себя такие методологии, как Agile, Scrum, Kanban и Waterfall, каждая из которых имеет свои преимущества и недостатки.

Понимание этих методологий и их применимость в различных сценариях является критически важным для успешного управления проектами и разработки соответствующих инструментов.

На основании проведенного анализа были сформулированы основные требования к разрабатываемой веб–платформе. Эти требования включают в себя удобный и интуитивно понятный интерфейс, возможность интеграции с другими системами и сервисами, высокая производительность и надежность, а также гибкость и масштабируемость системы. Учитывая эти требования, была разработана концепция новой веб–платформы, которая максимально удовлетворяет потребности современных IT–команд.

Таким образом, проведенное исследование в первой главе заложило прочную основу для последующей разработки веб–платформы для управления IT–проектами. Выявленные особенности, сравнительный анализ существующих решений, изучение теоретических основ и формулировка требований к системе позволили создать четкое представление о направлениях и приоритетах в разработке, что значительно повысит вероятность успешной реализации проекта.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И ВЫБОР СРЕДСТВ РАЗРАБОТКИ ВЕБ-ПЛАТФОРМЫ

2.1. Основные этапы разработки веб-платформы для управления IT-проектами

Разработка веб-платформы для управления IT-проектами включает в себя несколько ключевых этапов, каждый из которых важен для успешного завершения проекта. В этом разделе рассмотрены основные этапы разработки, начиная с анализа требований и заканчивая поддержкой и обслуживанием системы после её развертывания (Рис. 2.1).



Рисунок 2.1. Этапы разработки

На этапе анализа требований проводится детальный сбор и анализ функциональных и нефункциональных требований посредством общения с заказчиками и пользователями. Затем создаются спецификации, включающие описание всех функций и характеристик системы. После этого определяется приоритет наиболее важных и срочных требований, которые должны быть реализованы в первую очередь. Результаты сбора требований приводятся в приложении 1.

Во время проектирования архитектуры системы определяется архитектурный стиль, например, микросервисная архитектура. Создаются диаграммы компонентов, развертывания и последовательностей для моделирования системы. Также выбирается стек технологий для разработки frontend, backend и баз данных.

На этапе разработки пишется код и создается функциональность системы. Разработка backend включает реализацию микросервисов, создание RESTful API и интеграцию с базами данных и внешними сервисами. Разработка frontend заключается в создании пользовательского интерфейса и взаимодействии с API. Затем все компоненты системы объединяются и проверяется их совместимость.

Тестирование системы проводится для обеспечения ее качества и надежности. Проводятся юнит-тестирование для проверки отдельных компонентов, интеграционное тестирование для проверки взаимодействия между компонентами, системное тестирование для полной проверки системы на соответствие требованиям и приемочное тестирование для проверки системы пользователями на соответствие их ожиданиям и требованиям.

Этап развертывания включает подготовку инфраструктуры, настройку серверов, баз данных и сетевых компонентов. Затем происходит установка и настройка всех компонентов системы на сервере или в облаке, а также миграция данных из старых систем или других источников в новую платформу.

Основные этапы разработки веб-платформы включают анализ требований, проектирование архитектуры, разработку, тестирование и развертывание, что помогает обеспечить успешную реализацию проекта и соответствие требованиям пользователей.

Основные этапы разработки веб-платформы для управления IT-проектами включают в себя анализ требований, планирование проекта, проектирование архитектуры, разработку, тестирование, развертывание, поддержку и обслуживание, а также постоянное улучшение системы. Следование этим этапам помогает обеспечить успешную реализацию проекта, соответствующую требованиям заказчиков и пользователей.

2.2. Создание абстрактной модели веб–платформы

UML (Unified Modeling Language) является универсальным языком, открытым стандартом, использующим графические нотации для создания абстрактных моделей систем, известных как UML–модели. UML применяется не только для моделирования программного обеспечения, но и для моделирования бизнес–процессов, системного проектирования и отображения организационных структур.

Использование UML помогает разработчикам программного обеспечения договориться о графических обозначениях для представления общих понятий, таких как классы, компоненты, обобщение, агрегация и поведение. Это позволяет им сосредоточиться на проектировании и архитектуре системы.

В рамках данной работы были разработаны различные UML–диаграммы, включая диаграммы прецедентов, классов, последовательностей и деятельности объектов. Эти диаграммы используются для определения, визуализации, проектирования и документирования системы.

UML–диаграммы прецедентов

Диаграммы прецедентов (рис. 2.2) используются для моделирования системы с точки зрения прецедентов (или вариантов использования). Они часто применяются для моделирования контекста системы, подсистемы или класса, а также для определения требований к поведению этих элементов.

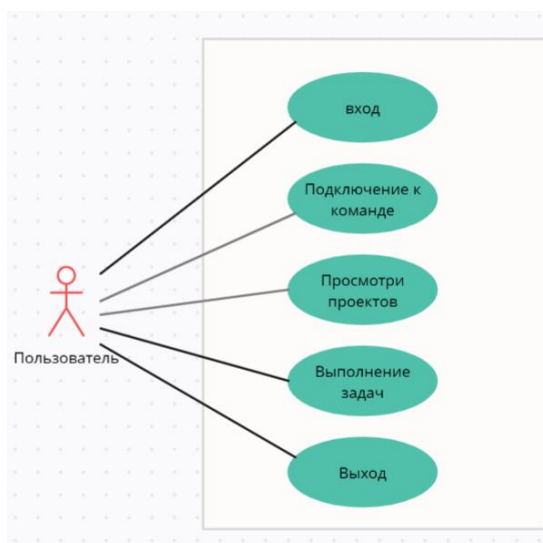


Рисунок 2.2. Диаграмма прецедентов

Диаграммы прецедентов играют важную роль в визуализации, спецификации и документировании поведения элементов. Они облегчают понимание систем, подсистем или классов, представляя их использование извне в соответствующем контексте. Эти диаграммы также важны для тестирования систем в процессе проектирования и для анализа их структуры при обратном проектировании.

В UML диаграммы прецедентов помогают визуализировать поведение системы, подсистемы или класса, чтобы пользователи могли понять их использование, а разработчики – реализовать соответствующий элемент. Обычно такие диаграммы включают прецеденты, акторов, зависимости, обобщения и ассоциации. Прецеденты, изображаемые в виде эллипсов, структурируют поведенческие сущности в модели. Прецедент представляет собой функциональность системы, предоставляющую пользователю значимый и измеримый результат.

UML–диаграммы классов

UML (Unified Modeling Language) диаграммы классов используются для статического моделирования структуры системы, отображая классы, их атрибуты, методы и отношения между ними. Эти диаграммы важны для проектирования и документирования систем, поскольку помогают

разработчикам и аналитикам лучше понять структуру и взаимодействия различных компонентов системы (рис. 2.3).

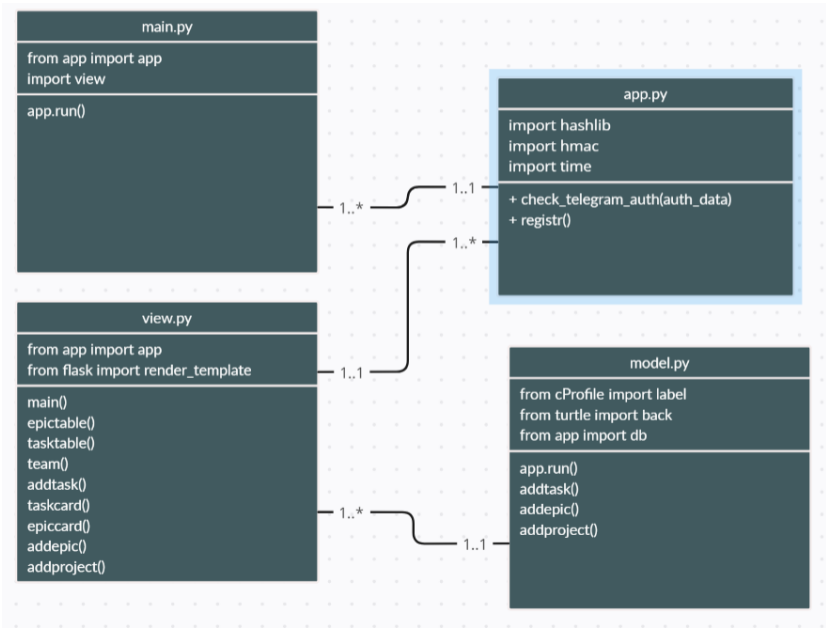


Рисунок 2.3. Диаграмма классов

Диаграммы классов представляют собой основную сущность, называемую классом, которая описывает объект или концепцию с общими характеристиками. Класс включает имя, атрибуты (свойства) и методы (функции), которые этот класс может выполнять.

Отношения между классами играют ключевую роль в UML-диаграммах. Ассоциация показывает связь между двумя классами, агрегация представляет отношения "часть–целое", где часть может существовать независимо от целого. Композитная агрегация (композиция) указывает на то, что части не могут существовать отдельно от целого. Обобщение описывает отношение наследования, где подкласс наследует свойства и методы суперкласса. Зависимость указывает на то, что изменения в одном классе могут влиять на другой класс.

Диаграммы классов также включают интерфейсы, которые представляют собой абстрактные классы, определяющие набор методов, которые должны быть реализованы классами. Примечания добавляются для пояснений или комментариев к диаграмме.

Диаграммы классов в UML являются мощным инструментом для визуализации и документирования структурных аспектов системы. Они помогают четко определять компоненты системы и их взаимодействия, что обеспечивает основу для дальнейшего проектирования и реализации программного обеспечения.

UML–диаграммы последовательности

Диаграмма последовательности (рис. 2.4) иллюстрирует, как объекты взаимодействуют друг с другом, упорядочивая эти взаимодействия по времени. На ней показаны только те объекты, которые участвуют в конкретных взаимодействиях, без указания статических связей с другими объектами. Ключевым аспектом диаграммы последовательности является динамика взаимодействия объектов во временной шкале.

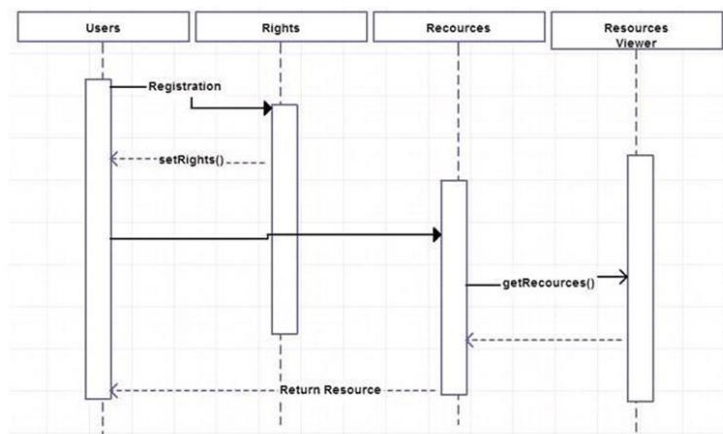


Рисунок 2.4. Диаграмма последовательности

Диаграмма имеет два измерения. В одном из них объекты представлены слева направо вертикальными линиями, каждая из которых называется линией жизни объекта. В верхней части каждой линии жизни графически изображен объект в виде прямоугольника. Линия жизни объекта (object lifeline) обозначается пунктирной вертикальной линией и показывает период времени, в течение которого объект существует в системе и может участвовать в взаимодействиях. Если объект постоянно присутствует в системе, его линия жизни проходит от верхней до нижней части диаграммы.

Объект, инициирующий взаимодействие, находится крайним слева на диаграмме. Объекты, которые взаимодействуют с ним, располагаются правее. Таким образом, объекты на диаграмме последовательности упорядочены в зависимости от их активности в процессе взаимодействия.

UML–диаграммы деятельности объектов

Диаграмма деятельности (рис. 2.5) используется для моделирования динамических аспектов поведения системы. Это своего рода блок–схема, которая показывает, как управление переходит от одной деятельности к другой. Диаграммы деятельности применяются для моделирования динамических аспектов системы, обычно для описания последовательных (а иногда и параллельных) шагов вычислительного процесса.

Они также могут отображать жизненный цикл объекта, демонстрируя его переходы из одного состояния в другое на различных этапах потока управления. Диаграммы деятельности могут использоваться самостоятельно для визуализации, спецификации, конструирования и документирования динамического поведения множества объектов. Они также подходят для моделирования потока управления при выполнении операций. В отличие от диаграмм взаимодействий, которые акцентируют внимание на переходах управления между объектами, диаграммы деятельности описывают переходы от одной деятельности к другой.

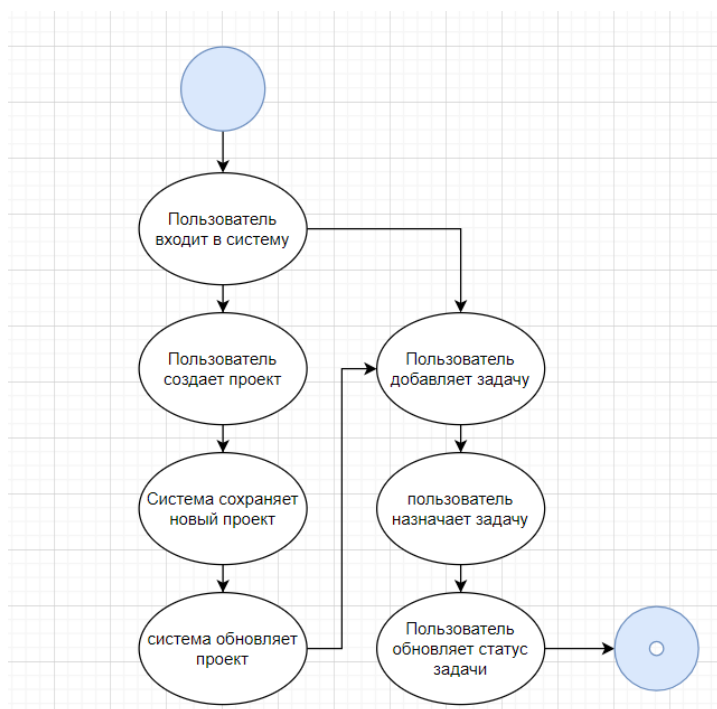


Рисунок 2.5. Диаграмма деятельности

2.3. Скетч–разработка интерфейса и дизайна веб–платформы

Скетч представляет собой схематичное изображение приложений, включающее все навигационные блоки, изображения, интерактивные элементы и структуру. Создание скетчей является ключевым этапом в разработке нового проекта, на котором принимаются концептуальные решения по интерфейсу будущего приложения. Этот этап дополняет и уточняет предварительное техническое задание, после чего начинается процесс создания скетчей. Данный этап необходим для решения нескольких задач:

- Помогает выбрать наиболее перспективный вариант из множества идей.
- Экономит время клиента и дизайнера.
- Снижает финансовые затраты, связанные с возможными исправлениями на более поздних этапах разработки.

На рис. 2.6 представлен скетч веб–платформы для управления IT–проектами сделанный в Figma.

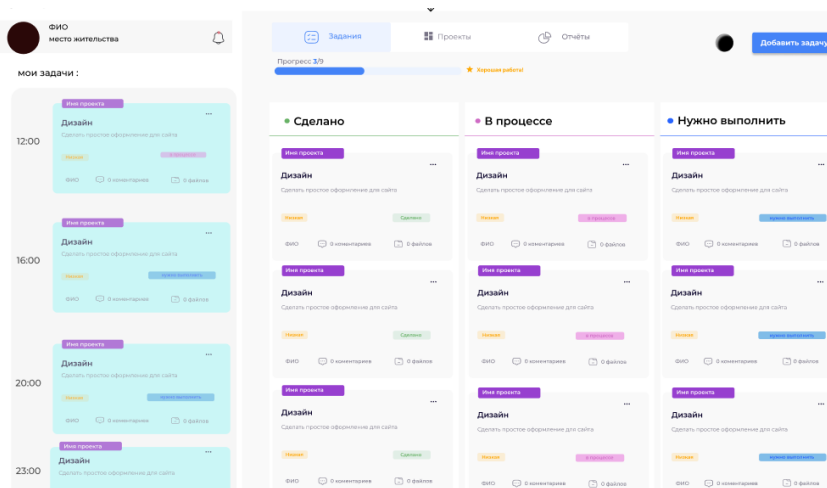


Рисунок 2.6. Скетч веб–платформы для управление IT–проектами

2.4. Выбор технических средств и программного инструментария для разработки системы

Разработка веб–платформы для управления IT–проектами требует использования различных инструментов и технологий. Ниже приведено описание основных средств разработки, используемых в данном проекте.

Python

Python — это высокоуровневый язык программирования, известный своей простотой и читаемостью кода. Его обширная стандартная библиотека и богатая экосистема сторонних пакетов делают Python идеальным выбором для разработки веб–приложений. В контексте данного проекта Python используется для серверной части приложения, где он обрабатывает бизнес–логику, взаимодействует с базой данных и управляет пользовательскими запросами. Сильные стороны Python включают поддержку нескольких парадигм программирования (объектно–ориентированного, процедурного и функционального), а также наличие множества фреймворков и библиотек для веб–разработки.

MySQL

MySQL — это популярная реляционная система управления базами данных с открытым исходным кодом. Она используется для хранения и управления данными проекта, включая информацию о пользователях, задачах,

проектах и других связанных данных. MySQL обеспечивает высокую производительность, надежность и масштабируемость, что делает её подходящим выбором для веб-приложений. Для обеспечения целостности данных и выполнения сложных запросов MySQL поддерживает язык SQL (Structured Query Language).

phpMyAdmin

phpMyAdmin — это веб-интерфейс для управления MySQL базами данных. Он позволяет администраторам баз данных выполнять различные задачи, такие как создание и удаление баз данных, таблиц, полей и индексов; выполнение SQL-запросов; управление пользователями и привилегиями; и импорт и экспорт данных. В рамках данного проекта phpMyAdmin используется для удобного управления базой данных MySQL, что упрощает процессы настройки и администрирования базы данных.

Flask

Flask — это легковесный веб-фреймворк для Python, который используется для разработки веб-приложений. Flask следует философии "микрофреймворка", предоставляя минимальный набор инструментов и компонентов, необходимых для создания веб-приложения, что дает разработчикам большую гибкость и контроль над архитектурой проекта. Flask поддерживает расширения, которые могут добавить функциональность в базовое приложение, такие как аутентификация, ORM (объектно-реляционное отображение), формы, загрузка файлов и многое другое. В данном проекте Flask используется для создания серверной логики, обработки HTTP-запросов и формирования ответов, а также для интеграции с другими компонентами системы, такими как базы данных и сторонние API.

Эти инструменты вместе обеспечивают мощную и гибкую платформу для разработки, развертывания и управления веб-приложением для управления IT-проектами.

Выводы ко второй главе

Во второй главе магистерской квалификационной работы были рассмотрены ключевые аспекты проектирования и выбора средств разработки веб–платформы для управления IT–проектами. Проведенное исследование позволило сделать следующие важные выводы:

В процессе разработки веб–платформы были выделены несколько ключевых этапов, включая анализ требований, проектирование архитектуры, разработку, тестирование и развертывание. Каждый этап имеет свои уникальные задачи и цели, которые необходимо тщательно прорабатывать для обеспечения успешной реализации проекта. Последовательное выполнение всех этапов позволяет создать качественное и функциональное решение, соответствующее потребностям пользователей.

На этапе создания абстрактной модели был разработан концептуальный дизайн системы, включающий основные компоненты и их взаимодействие. Это позволило получить общее представление о структуре и функциональности будущей платформы. Создание такой модели помогает минимизировать риски и потенциальные проблемы на более поздних этапах разработки, обеспечивая ясное видение проекта и его архитектуры.

Важным этапом проектирования стало создание скетчей интерфейса и дизайна платформы. Этот процесс включал в себя разработку начальных набросков и макетов, которые помогли визуализировать пользовательский интерфейс и его элементы. Скетч–разработка позволила предварительно оценить удобство использования и эстетическую привлекательность системы, что способствует созданию интуитивно понятного и приятного для пользователя интерфейса.

В результате тщательного анализа были выбраны наиболее подходящие технические средства и инструменты для разработки веб–платформы, включая Python, Flask, MySQL и phpMyAdmin. Эти технологии обеспечивают высокую производительность, гибкость и надежность системы, а также

позволяют эффективно управлять данными и обеспечивать масштабируемость проекта. Выбор соответствующих инструментов играет ключевую роль в успешной реализации и дальнейшей поддержке веб–платформы.

Таким образом, вторая глава магистерской квалификационной работы заложила прочный фундамент для практической реализации проекта. Выявленные этапы разработки, создание абстрактной модели, скетч–разработка интерфейса и выбор технических средств обеспечили комплексный подход к проектированию веб–платформы. Это позволит создать эффективное и надежное решение для управления IT–проектами, соответствующее современным требованиям и ожиданиям пользователей.

ГЛАВА 3. РАЗРАБОТКА ВЕБ-ПЛАТФОРМЫ ДЛЯ УПРАВЛЕНИЯ ИТ-ПРОЕКТАМИ

3.1. Проектирование и разработка базы данных

Для обеспечения работы веб-приложения была спроектирована база данных «itplaner» (рис.3.1). Данная БД состоит из шести таблиц:

- Users – хранит в себе данные о пользователях
- Teams – содержит информацию о командах разработчиков
- Team_members – включает информацию о принадлежности пользователей к той или иной команде и их роли.
- Tasks – содержит информацию о задачах созданных в веб-приложении
- Task_statuses – хранит информацию о том на какой стадии находится задача, например "новая", "в работе", "завершена"
- Subtasks – содержит информацию о задачах созданных в веб-приложении, кроме того, о том частью какой задачи является подзадача

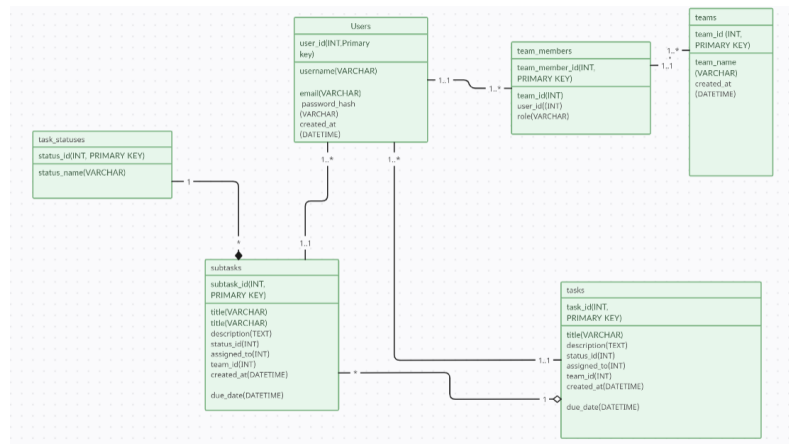


Рисунок 3.1. База данных «itplaner»

Также был создан файл «config.py», который будет содержать настройки подключения к БД. Листинг данного файла имеет следующий вид:

```

class Configuration:

    #All config

    DEBUG = True

    CSRF_ENABLED = True
    
```

```
SECRET_KEY = 'asfasfasfas'
```

```
#DataBase
```

```
SQLALCHEMY_TRACK_MODIFICATIONS = False
```

```
SQLALCHEMY_DATABASE_URI=
```

```
'mysql+pymysql://root:@localhost/itplaner'
```

3.2. Структура разрабатываемой системы и пользовательский интерфейс

На рисунке 3.2 представлена структура системы в виде схемы. Данная схема отображает все веб–страницы:

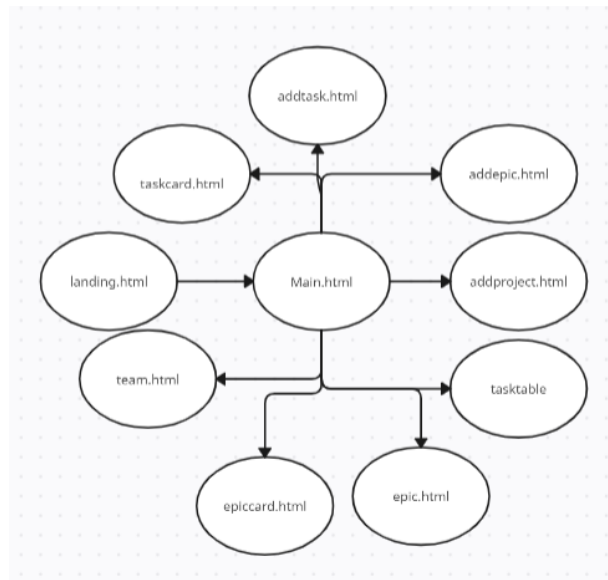


Рисунок 3.2. Схема структуры веб–платформы

При первом заходе на веб–платформу пользователя встречает гостевая страница «landing.html» (рис. 3.3). На данной странице отображается информация о веб–платформе и контактные данные.

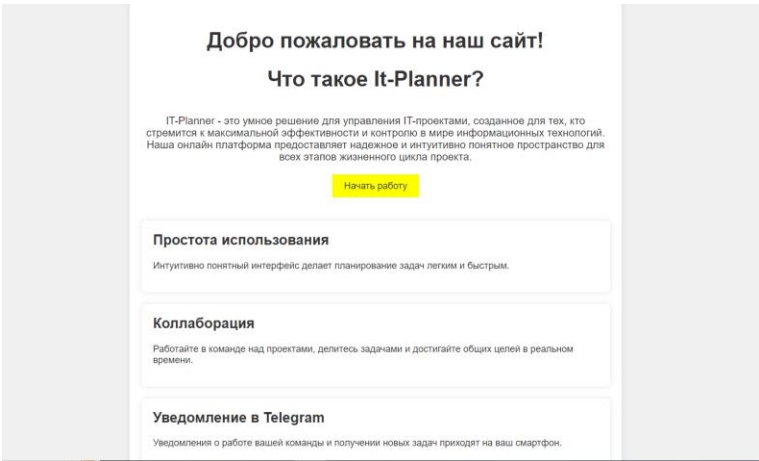


Рисунок 3.3. lending.html

Главная страница «main.html» является основной рабочей зоной (рис.3.4).

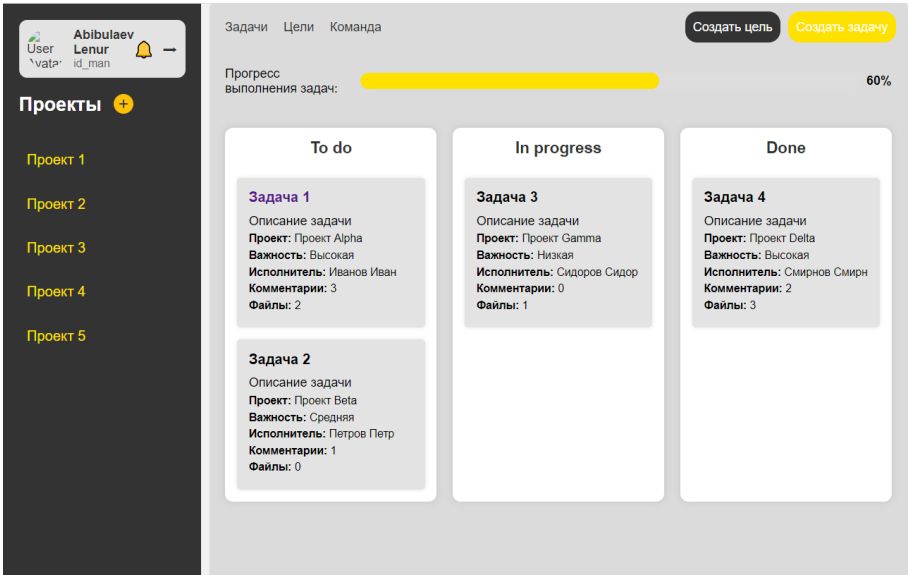


Рисунок 3.4. Основная страница «main.html»

Данная страница содержит в себе весь необходимый функционал для управления проектами. Возможность создавать «Задачу» и «Подзадачу» есть только у тимлида (рис.3.5, рис.3.6).

The screenshot shows a web interface with a top navigation bar containing 'Задачи', 'Цели', and 'Команда'. On the right of the bar are two buttons: 'Создать цель' (black) and 'Создать задачу' (yellow). The main content area is titled 'Создание цели'. It contains the following fields: 'Название цели' (text input), 'Описание цели' (text area), 'Дата начала' (date picker with 'дд.мм.гггг' format), and 'Срок выполнения' (date picker with 'дд.мм.гггг' format). At the bottom of the form is a yellow button labeled 'Создать цель'.

Рисунок 3.5. Создание задачи.

The screenshot shows a web interface with a top navigation bar containing 'Задачи', 'Цели', and 'Команда'. On the right of the bar are two buttons: 'Создать цель' (black) and 'Создать задачу' (yellow). The main content area is titled 'Добавить задачу'. It contains the following fields: 'Название проекта' (text input), 'Цель' (text input), 'Имя руководителя' (text input), 'Название задачи' (text input), 'Описание задачи' (text area), 'Крайний срок' (date picker with 'дд.мм.гггг' format), 'Важность задачи' (dropdown menu with 'Низкая' selected), and 'Загрузить файл' (file upload button with 'Выберите файл' and 'Файл не выбран' text). At the bottom of the form is a yellow button labeled 'Добавить'.

Рисунок 3.6. Создание подзадачи

Возможность создавать проекты есть у всех пользователей (рис.3.7). Пользователь, который создал проект, автоматически становится тимлидом для этого проекта, что дает ему возможность добавлять задачи и подзадачи.

Рисунок 3.7. Создание проекта

На рисунке 3.4 изображена открытая вкладка задачи, на ней можем видеть прогресс выполнения задач и три колонны «To do», «In progress», «Done». При нажатии на задачу, откроется окно с в котором можно вносить изменения, связанные с работой над этой задачей(рис.3.8).

Рисунок 3.8. Карточка задачи

Предложенная веб–платформа предоставляет интуитивно понятный интерфейс и все необходимые функции для успешной работы.

Вывод к третьей главе

В третьей главе магистерской квалификационной работы были рассмотрены основные аспекты разработки веб–платформы для управления IT–проектами. Проведенное исследование и практическая реализация включали следующие ключевые элементы:

В ходе проектирования и разработки базы данных была создана структура, обеспечивающая надежное хранение и эффективное управление данными, необходимыми для функционирования платформы. База данных была спроектирована с учетом требований производительности, масштабируемости и безопасности. Это включало разработку таблиц, установление связей между ними и создание индексов для оптимизации выполнения запросов. Такой подход позволил создать основу для надежного хранения данных, что критически важно для работы системы управления IT-проектами.

В процессе проектирования структуры системы были определены основные компоненты и их взаимодействие. Разработанная архитектура включала серверную часть, клиентскую часть и базу данных. Была применена микросервисная архитектура, которая обеспечивает модульность, гибкость и масштабируемость системы. Каждый компонент системы был тщательно спроектирован для обеспечения высокой производительности и надежности. Это позволило создать систему, которая может эффективно справляться с нагрузками и легко адаптироваться к изменяющимся требованиям.

Разработка пользовательского интерфейса включала создание интуитивно понятных и удобных для пользователя элементов управления и навигации. Были использованы современные принципы UX/UI дизайна, чтобы обеспечить положительный пользовательский опыт. Интерфейс системы был разработан с учетом потребностей различных категорий пользователей, включая менеджеров проектов, разработчиков и других участников команды. Такой подход позволяет пользователям легко ориентироваться в системе, выполнять необходимые действия и эффективно управлять проектами. Таким образом, третья глава магистерской квалификационной работы обобщает результаты проектирования и разработки веб-платформы для управления IT-проектами.

Заключение

Магистерская квалификационная работа была посвящена проектированию и разработке веб–платформы для управления IT–проектами. В процессе исследования и разработки были рассмотрены и решены ключевые задачи, связанные с проектированием, выбором технологий и реализацией системы. Итогом данной работы стало создание комплексного решения, отвечающего современным требованиям и обеспечивающего эффективное управление IT–проектами.

На первом этапе работы был проведен детальный анализ существующих решений и теоретических основ управления IT–проектами. Были определены ключевые требования к разрабатываемой платформе, что заложило прочный фундамент для дальнейших этапов разработки. Анализ показал, что существующие платформы имеют определенные недостатки и не всегда полностью удовлетворяют потребности пользователей, что подчеркнуло необходимость создания нового решения.

Во второй главе работы были рассмотрены этапы проектирования веб–платформы. Были разработаны абстрактные модели системы, проведена скетч–разработка интерфейса и дизайна, а также выбран технический инструментарий для реализации проекта. Выбор таких технологий, как Python, Flask, MySQL и phpMyAdmin, обеспечил гибкость, производительность и масштабируемость системы. Проектирование структуры системы и интерфейса позволило создать удобный и интуитивно понятный инструмент для пользователей.

В третьей главе была подробно описана реализация веб–платформы. Проектирование и разработка базы данных обеспечили надежное хранение и управление данными, что критически важно для эффективного функционирования системы. Разработанная архитектура системы, основанная на микросервисном подходе, обеспечила модульность и возможность масштабирования. Пользовательский интерфейс был разработан с учетом

современных принципов UX/UI дизайна, что обеспечило положительный пользовательский опыт и удобство работы с системой.

В результате проведенного исследования и разработки была создана мощная и надежная веб-платформа для управления IT-проектами, которая сочетает в себе лучшие практики и современные технологии. Платформа обладает высокой производительностью, гибкостью и масштабируемостью, что позволяет эффективно справляться с различными задачами управления проектами. Благодаря интуитивно понятному интерфейсу, пользователи могут легко ориентироваться в системе и эффективно выполнять свои задачи.

Данная работа внесла значительный вклад в развитие инструментов управления IT-проектами, предложив новое решение, которое может быть использовано как в небольших командах, так и в крупных организациях. Полученные результаты и созданная система могут служить основой для дальнейших исследований и разработок в данной области, способствуя улучшению процессов управления проектами и повышению их эффективности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Комплексное руководство по разработке: от мобильных приложений до веб-технологий: учебное пособие для вузов/ А.Н. Баланов. – Санкт-Петербург: Лань, 2024.– 412с.
2. Система управления проектами организации: анализ подходов и существующих программных решений/ [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/sistema-upravleniya-proektami-organizatsii-analiz-podhodov-i-suschestvuyuschih-programmnyh-resheniy>
3. Jira /Официальный сайт [Электронный ресурс] – Режим доступа: <https://blog.skillfactory.ru/glossary/jira/>
4. Trello /Официальный сайт [Электронный ресурс] – Режим доступа: <https://trello.com/ru/tour>
5. Asana /Официальный сайт [Электронный ресурс] – Режим доступа: <https://asana.com/ru/uses>
6. Управление ИТ-проектами : учебное пособие / А. Г. Доррер, М. Г. Доррер, А. А. Попов. — Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2019. — 174 с
7. Основы управления ИТ-проектами : учебное пособие / составители Е. Р. Кирколуп [и др.]. — Барнаул : АлтГПУ, 2017. — 176 с.
8. Методические основы управления ИТ- проектами : учебное пособие / В. И. Грекул, Н. Л. Коровкина, Ю. В. Куприянов. — 2-е изд. — Москва : ИНТУИТ, 2016. — 473 с.
9. Введение в управление проектной деятельностью: основы формирования, управления и коммерциализации инновационных проектов : учебно-методическое пособие / Д. Ю. Миронова, И. В. Баранов, Е. Е. Помазкова, О. Н. Румянцева. — Санкт-Петербург : НИУ ИТМО, 2022. — 89 с.
10. Методы разработки веб-приложений : учебно-методическое пособие / А. М. Садыков. — Иваново : ИГЭУ, 2019. — 72 с.

11. Проектирование программных систем : учебное пособие / М. Б. Туманова, Е. К. Михайлова, Е. А. Муравьева. — Москва : РТУ МИРЭА, 2023. — 138 с.
12. Зубкова, Т. М. Технология разработки программного обеспечения / Т. М. Зубкова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 252 с.
13. Головнин, О. К. Введение в системное программирование и основы жизненного цикла системных программ : учебное пособие / О. К. Головнин, А. А. Столбова. — Самара : Самарский университет, 2021. — 172 с.
14. Голубева, А. А. Разработка и анализ требований : методические указания / А. А. Голубева. — Москва : ТУСУР, 2018. — 85 с.