

Команда №4

Сапуparty



Спецификация требований к программному обеспечению

Документ

Содержание

| | |
|--|----------|
| 1 ВВЕДЕНИЕ..... | 3 |
| 1.1 Цель..... | 3 |
| 1.2 Область применения | 3 |
| 2 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ..... | 4 |
| 2.1 Программные интерфейсы | 4 |
| 2.2 Интерфейс пользователя | 4 |
| 2.3 Характеристики пользователей..... | 7 |
| 2.4 Предположения и зависимости | 8 |
| 3 СИСТЕМНЫЕ ТРЕБОВАНИЯ..... | 9 |
| 3.1 Функциональные требования | 9 |
| 3.2 Нефункциональные требования | 9 |

1 ВВЕДЕНИЕ

1.1 Цель

Данный документ представляет собой спецификацию требований к программному обеспечению (SRS) для веб-приложения "To-Do List". Цель этого документа — предоставить подробное описание функциональных и нефункциональных требований к системе. Этот документ предназначен для использования всеми заинтересованными сторонами, включая разработчиков, тестировщиков, менеджеров проектов и конечных пользователей, для обеспечения общего понимания продукта, который будет разработан.

1.2 Область применения

"To-Do List" — это веб-приложение, предназначенное для помощи пользователям в управлении своими задачами и проектами. Система предоставляет два основных режима использования: простой режим для управления личными задачами и расширенный режим для организации задач в рамках проектов и многоэтапных треков.

Основные возможности включают:

- Создание, редактирование и удаление задач.
- Установка уровня сложности и сроков выполнения для задач.
- Отслеживание статуса выполнения задач.
- Организация задач по проектам и трекам.
- Аутентификация и авторизация пользователей для защиты данных.

Приложение "To-Do List" будет охватывать следующие функциональные области:

- Управление задачами: Все операции, связанные с созданием, редактированием, удалением и отслеживанием задач.
- Управление проектами: Функциональность для группировки задач в проекты и треки.
- Аутентификация пользователей: Регистрация, вход и выход из системы.
- Пользовательский интерфейс: интуитивно понятный и адаптивный веб-интерфейс для взаимодействия с системой.

2 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ

2.1 Программные интерфейсы

Программа будет взаимодействовать со следующими внешними системами

- База данных H2: Встроенная реляционная база данных, используемая для хранения данных.

- Веб-браузеры: Для отображения пользовательского интерфейса (HTML, CSS, JavaScript) и обработки взаимодействий с пользователем.

Также в программе используются следующие библиотеки и фреймворки:

- Spring Boot.
 - spring-boot-starter-web - Для создания веб-приложений.
 - spring-boot-starter-thymeleaf - Для серверной обработки HTML-шаблонов.
 - spring-boot-starter-data-jpa - Для взаимодействия с базой данных с использованием JPA.
 - spring-boot-starter-security - Для аутентификации и авторизации.
 - spring-boot-starter-test - Для тестирования.
 - spring-boot-devtools - Для улучшений во время разработки.
 - JDBC-драйвер для H2.
 - Spring Data JPA - Предоставляет поддержку репозитория для JPA.
 - Thymeleaf Extras Spring Security - Интегрирует Spring Security с Thymeleaf.
 - BCryptPasswordEncoder - Для хеширования паролей.
- И со следующими интерфейсами:
- HTTP/HTTPS - Для веб-коммуникации.
 - JDBC (Java Database Connectivity) - Для подключения к базе данных.
 - JPA (Java Persistence API) - Для объектно-реляционного отображения.
 - API Spring Security - Для настройки и управления безопасностью.

2.2 Интерфейс пользователя

Система будет взаимодействовать с пользователем через интуитивно понятный и адаптивный веб-интерфейс, реализованный с использованием Thymeleaf, HTML, CSS и JavaScript. Основные этапы взаимодействия:

- 1) Авторизация и аутентификация:

- Пользователи будут перенаправлены на страницу входа, где они смогут ввести свои учетные данные (логин/пароль).

- После успешной аутентификации пользователь получит доступ к основным функциям системы.

- Система будет поддерживать различные роли пользователей, ограничивая доступ к определенным функциям в зависимости от роли.

2) Управление задачами (To-Do):

- На главной странице или специальной странице задач будет отображаться список всех задач.

- Пользователи смогут создавать новые задачи через форму, просматривать детали существующих задач, редактировать их (название, описание, срок выполнения, статус, приоритет) и удалять.

- Будет предусмотрена возможность фильтрации и сортировки задач по сложности.

3) Управление проектами:

- Пользователи смогут создавать, просматривать, редактировать и удалять проекты.

- Будет возможность привязывать задачи к конкретным проектам.

Мокапы представлены на рисунках 2.1-2.5:

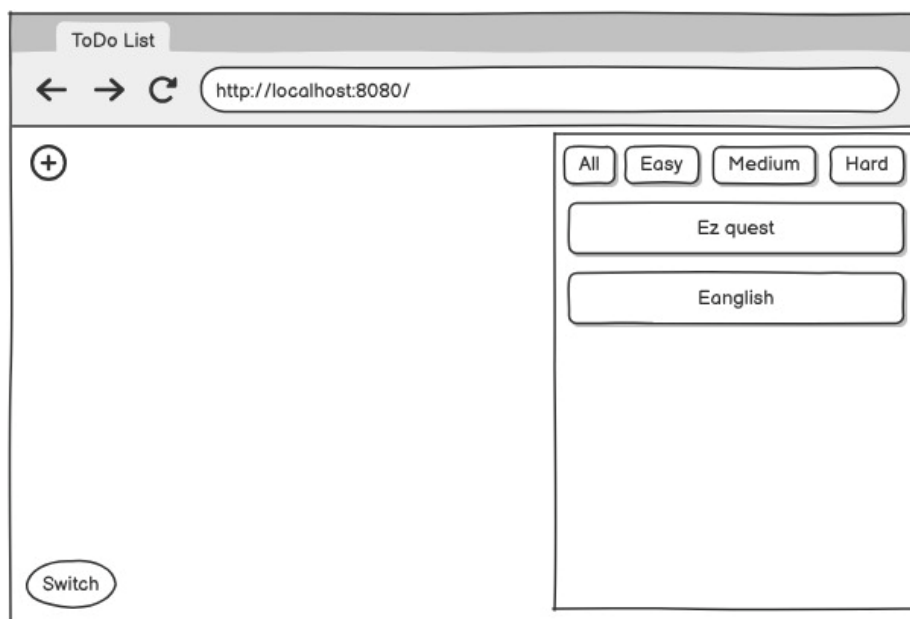


Рисунок 2.1 – Главная страница

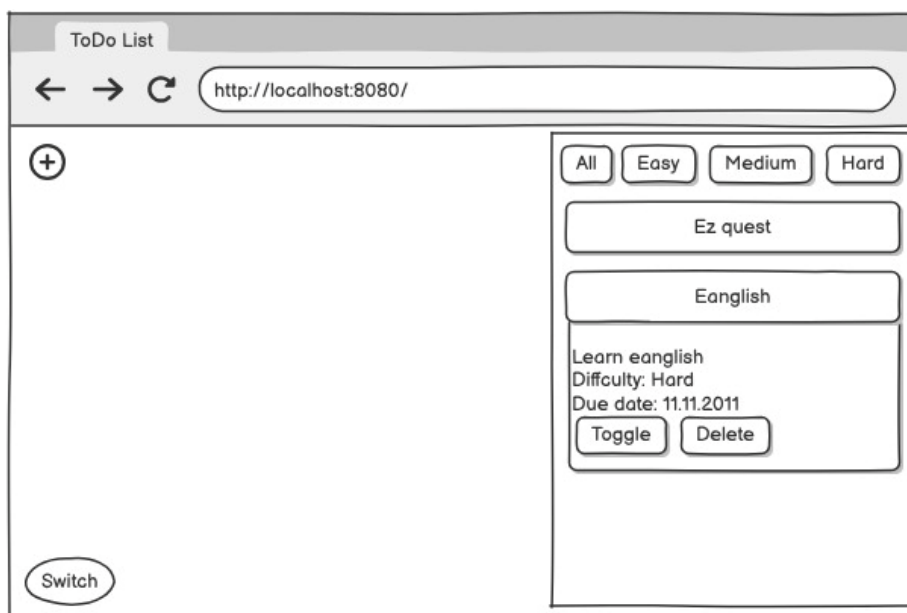


Рисунок 2.2 – Просмотр задач

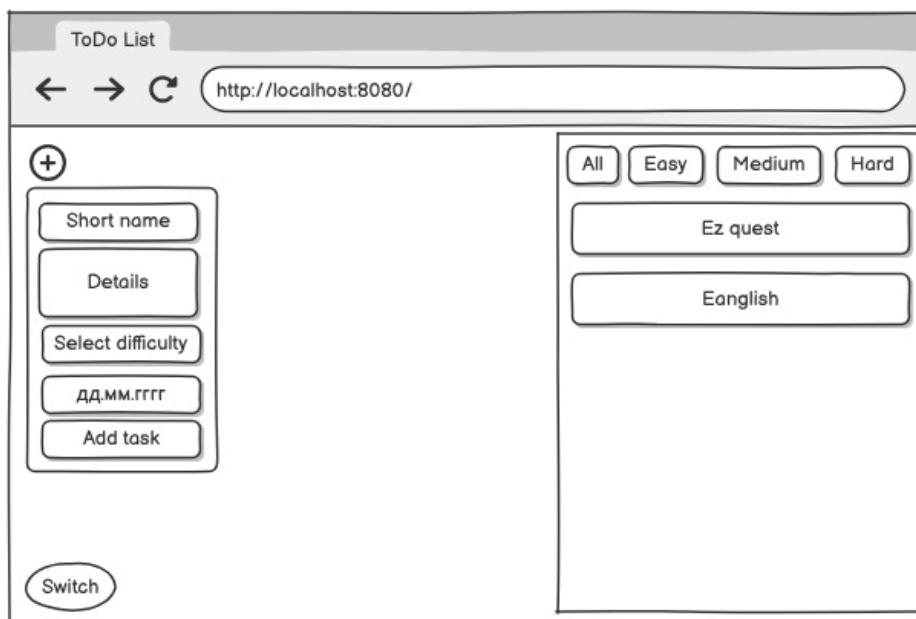


Рисунок 2.3 – Добавление задачи

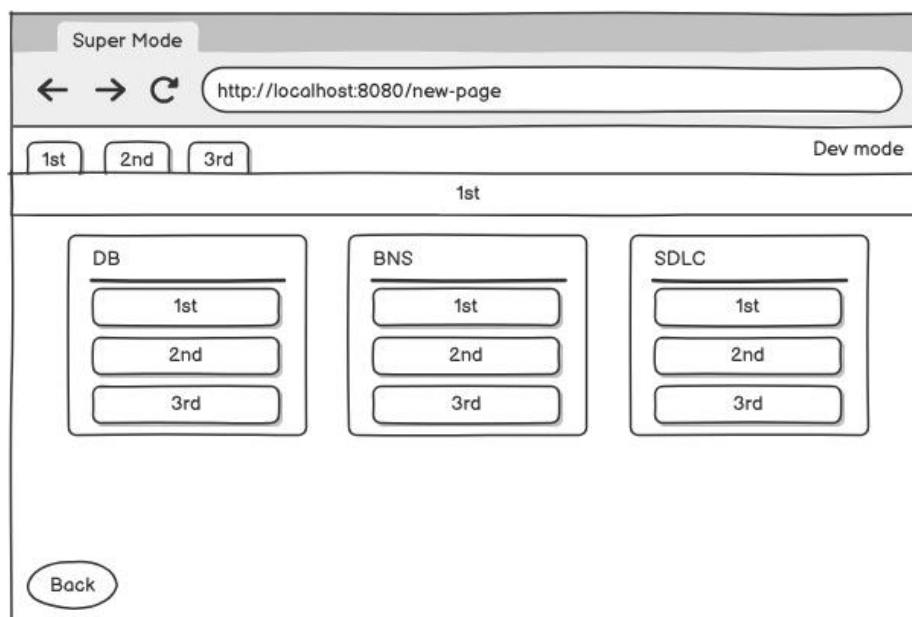


Рисунок 2.4 – Просмотр суперзаданий

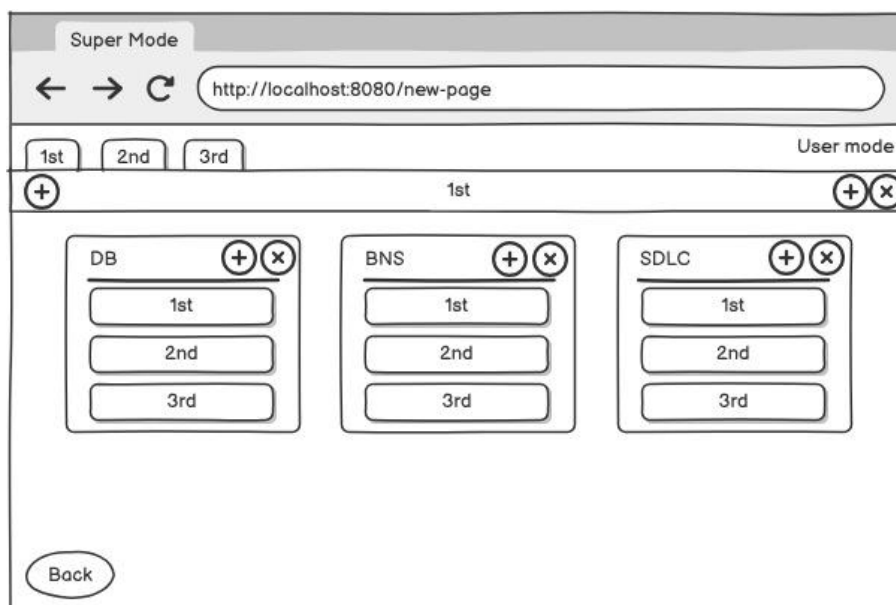


Рисунок 2.5 – редактирования суперзаданий

2.3 Характеристики пользователей

В системе можно выделить несколько групп пользователей.

Обычные пользователи будут использовать систему для управления своими личными задачами, отслеживания прогресса по проектам и учета затраченного времени. Их уровень образования может варьироваться от среднего до высшего, а опыт работы может быть самым разнообразным. Ожидается, что они обладают средней технической грамотностью,

достаточной для комфортной работы с веб-приложениями, заполнения форм и взаимодействия со списками.

Менеджеры проектов будут отвечать за создание и управление проектами, распределение задач, мониторинг их выполнения и анализ отчетов по затраченному времени. Как правило, это специалисты с высшим образованием в области управления, инженерии или IT, имеющие значительный опыт в управлении проектами. Их техническая грамотность выше средней, что позволяет им уверенно работать с веб-приложениями и эффективно анализировать данные.

2.4 Предположения и зависимости

На требования к системе могут влиять различные факторы. Изменения в бизнес-процессах компании или законодательстве напрямую диктуют необходимость адаптации функционала. Технологический прогресс предлагает новые решения или делает текущие устаревшими, требуя обновлений и пересмотра стандартов безопасности. Обратная связь от пользователей является ключевым источником для улучшения удобства и функциональности. Стратегические решения компании могут потребовать расширения возможностей системы для достижения новых целей. Практические ограничения, такие как бюджет и сроки, вынуждают приоритизировать функционал. Конкурентная среда стимулирует внедрение новых функций для поддержания актуальности. Наконец, потребность в интеграции с другими системами и рост числа пользователей с объемом данных требуют внимания к масштабируемости и производительности.

3 СИСТЕМНЫЕ ТРЕБОВАНИЯ

3.1 Функциональные требования

Функциональные требования представлены ниже:

1) Управление задачами (To-Do):

- Пользователи должны иметь возможность создавать, просматривать, редактировать и удалять задачи.
- Задачи должны иметь поля: название, описание, срок выполнения, статус (выполнено/не выполнено), приоритет.
- Возможность фильтрации и сортировки задач по различным критериям.

2) Управление проектами:

- Пользователи должны иметь возможность создавать, просматривать, редактировать и удалять проекты.
- Проекты должны иметь поля: название, описание, дата начала, дата окончания, статус.
- Возможность привязки задач к проектам.

3) Пользовательский интерфейс:

- Интуитивно понятный и адаптивный веб-интерфейс.
- Отображение списка задач и проектов.
- Формы для создания и редактирования задач/проектов/отслеживания времени.

4) База данных:

- Хранение информации о пользователях, задачах, проектах и отслеживании времени.
- Использование реляционной базы данных.

5) Удобство использования.

6) Простой и понятный интерфейс.

7) Минимальное количество шагов для выполнения основных операций.

3.2 Нефункциональные требования

Для данной системы важны несколько атрибутов качества, обеспечивающих её эффективность и долговечность.

Производительность является фундаментальным требованием, поскольку пользователи ожидают быстрой и отзывчивой работы без задержек при выполнении ключевых операций, таких как загрузка списков или создание новых элементов. Измерять её следует по времени отклика для основных действий, стремясь к показателям не более 2 секунд при стандартной нагрузке, а также по пропускной способности, например, до 100 запросов в секунду, при этом постоянно мониторя использование системных ресурсов.

Масштабируемость гарантирует, что система сможет эффективно функционировать при значительном увеличении числа пользователей и объёма данных, что является залогом её долгосрочной жизнеспособности. Это качество измеряется через нагрузочное тестирование, оценку архитектуры на предмет возможности горизонтального расширения и способность поддерживать прогнозируемый рост.

Удобство использования напрямую влияет на принятие системы пользователями и их продуктивность. Интерфейс должен быть интуитивно понятным, адаптивным и позволять выполнять основные операции с минимальным количеством шагов.

Надежность обеспечивает стабильную и бесперебойную работу системы, что критически важно для доверия пользователей. Её измеряют по времени безотказной работы, стремясь к показателю 99.5% в рабочие часы, а также по частоте сбоев и времени, необходимому для восстановления после них. Важна также корректная обработка ошибок с информативными сообщениями и логированием.

Сопровождаемость определяет легкость внесения изменений, исправления ошибок и добавления нового функционала. Это качество оценивается через аудит кода на предмет его структуры, читаемости, соответствия стандартам кодирования и наличия документации, а также по времени, затрачиваемому на исправление дефектов и внедрение новых функций.

Наконец, переносимость позволяет развертывать систему в различных средах с минимальными усилиями, обеспечивая гибкость и независимость от конкретной инфраструктуры. Измерение переносимости включает тестирование развертывания на разных платформах и оценку сложности конфигурационных изменений.