
[RFC] Task Manager Web Application

Summary: Веб-приложение для создания, организации и отслеживания выполнения проектов с возможностью разделения их на подзадачи с учетом приоритета задач и уровня ответственности исполнителя.

Created: Jan 22, 2019

Status: **WIP** | In-Review | Approved | Obsolete

Current Version: 1.0.0

Owner: manu11shet@gmail.com

Target Version: 1.1.0

Документ описывает дизайн, назначение и детали реализации проекта веб-приложения Task Manager, предназначенного для управления задачами в командной разработке продукта.

Актуальность продукта

Рынок веб-приложений для управления задачами является высококонкурентным, однако Task Manager обладает рядом преимуществ, которые делают его актуальным:

- **Простота использования:** Интерфейс приложения интуитивно понятен и удобен для пользователей с любым уровнем подготовки.
- **Гибкость:** Task Manager позволяет настраивать интерфейс и workflow в соответствии с потребностями конкретной компании.
- **Интеграция:** Task Manager может быть интегрирован как с внешними сервисами, такими как календари, системы управления версиями и чаты, так и с внутренними сервисами компании, существенно облегчая и ускоряя процесс разработки.

- **Безопасность:** Приложение может быть использовано в изолированном окружении с целью не допустить утечки чувствительных данных за пределы.
- **Доступная цена:** Task Manager будет доступен по цене для команд любого размера.

Функциональные возможности

- **Создание и управление проектами:**
 - Создание проектов с описанием, сроками и списком участников.
 - Добавление и удаление участников.
- **Создание и управление эпиками:**
 - Создание эпиков (комплекса задач, объединённых общей целью) с описанием.
 - Определение входящих в состав эпика подзадач.
- **Создание и управление задачами:**
 - Создание задач с описанием, приоритетом, сроками и исполнителями.
 - Назначение исполнителей и проверяющих для задач.
 - Прикрепление файлов к задачам.
 - Отслеживание статуса выполнения задач.
 - Комментирование задач.
- **Отслеживание прогресса:**
 - Наглядное отображение прогресса выполнения задач.
 - Отчеты о выполненных и просроченных задачах.

- Настройка уведомлений о событиях.

Архитектура проекта

1. Frontend (Клиентская часть):

- Язык программирования и фреймворк: JavaScript с использованием фреймворка **Nuxt.js** на **Vue.js** для построения интерфейса пользователя.
- **Компонентный подход**: разделение интерфейса на компоненты для легкости разработки, тестирования и обслуживания.
- **Роутинг**: использование маршрутизации для обеспечения навигации между различными представлениями приложения.

2. Backend (Серверная часть):

- Язык программирования и фреймворк: **PHP**, **Symfony**.
- API: разработка **RESTful API** для взаимодействия между клиентской и серверной частями.
- База данных: реляционная база данных **PostgreSQL**.
- Кэширование: использование **Redis** в качестве инструмента кэширования для ускорения доступа к часто используемым данным.

3. База данных:

- **Модель данных**: описание структуры данных для хранения информации о пользователях, задачах, комментариях и других сущностях.
- **Миграции**: использование механизма миграций для управления изменениями в схеме базы данных.

4. Аутентификация и Авторизация:

- **JWT** (JSON Web Tokens): для безопасной передачи информации об аутентификации между клиентом и сервером.
- **OAuth2**: для сторонних аутентификаций и делегирования доступа.

5. **Безопасность:**

- **HTTPS**: защита данных пользователя при передаче между клиентом и сервером.
- **Хеширование паролей**: хранение паролей в зашифрованном виде с использованием алгоритмов хеширования.

6. **Управление состоянием:**

- **Vuex**: для эффективного управления состоянием клиентского приложения.

7. **Уведомления:**

- **WebSocket**: для реального времени в уведомлениях о событиях (например, новые задачи, обновления и т. д.).

8. **Тестирование:**

- Модульное тестирование: использование библиотеки **PHPUnit** для модульного тестирования бэкенда и фронтенда.
- **Интеграционное тестирование**: проверка взаимодействия между компонентами системы.

9. **Деплоймент и масштабирование:**

- **Docker**: контейнеризация приложения для упрощения деплоймента и масштабирования.
- Облачные сервисы: развертывание на облачной платформе **AWS**.

10. **Мониторинг и логирование:**

- Инструменты мониторинга: настройка мониторинга с помощью Prometheus для отслеживания производительности и доступности приложения.
- Централизованный журнал событий: предоставление визуальных средств для отслеживания и обработки событий с помощью **Grafana** для быстрого выявления и решения проблем.
- Использование **Loki** для централизованного сбора и анализа логов.

11. Сервисы и Интеграции:

- **API интеграции:** создание API для интеграции со сторонними сервисами и инструментами разработки:
 - Сервисы аутентификации: **Google Auth**.
 - Сервисы уведомлений: **Slack**.
 - Сервисы CI/CD: **GitLab CI/CD**.
 - Сервисы видеоконференций: **Google Meet**.
 - Сервисы хранения файлов: **Google Drive**.
 - Системой управления версиями продукта: **Github**.

User-Product Communication Strategy

1. Документация:

- a. Разработка подробной онлайн-документации с инструкциями по использованию основных функций приложения.
- b. Создание видеоуроков и обучающих материалов, объясняющих процесс создания и управления задачами в приложении.

2. Интерфейс и Взаимодействие:

- a. Создание интуитивно понятного пользовательского интерфейса с простым созданием, редактированием и отслеживанием задач.

- b. Включение подсказок и всплывающих окон для объяснения основных функций при первом входе в приложение.

3. Уведомления и Оповещения:

- a. Регулярное отправление уведомлений о сроках выполнения задач, новых комментариях или изменениях в статусе.
- b. Уведомляет пользователей о важных событиях, таких как изменения в задачах, комментарии и т.д.
- c. Возможность настройки предпочтительных способов получения уведомлений (электронная почта, push-уведомления).

4. Поддержка и Обратная Связь:

- a. Предоставление чат-поддержки в реальном времени для оперативного решения проблем и ответа на вопросы.
- b. Форма обратной связи в приложении для сбора комментариев и предложений по улучшению.

5. Обновления и Релизы:

- a. Публикация подробных записей об обновлениях, включая новые функции, улучшения и исправления.
- b. Рассылка электронных писем с уведомлением о предстоящих обновлениях и инструкциями по изменениям.

- 6. **Сайт проекта:** На сайте проекта будут представлены описание продукта, документация, блог и контактная информация.

- 7. **Социальные сети:** Task Manager будет иметь страницы в социальных сетях, где будут публиковаться новости, обновления и обучающие материалы.

8. **Email:** Пользователи смогут подписываться на рассылку новостей и уведомлений о событиях.

9. **Внутренняя система чата:**

- a. Обеспечивает мгновенную связь между членами команды.
- b. Позволяет обсуждать задачи, делиться файлами и координировать работу.

10. **База знаний:**

- a. Содержит справочную информацию о функциях приложения, советах по работе и т.д.

User Flow (UX-сценарии)

1. **Создание проекта:**

Действия:

- Нажать кнопку "Создать проект".
- Ввести название проекта, описание, начальную и конечную дату.
- Выбрать руководителя проекта.
- Добавить участников проекта.
- Нажать кнопку "Создать".

Примечания:

- Руководитель проекта может добавлять и удалять участников проекта.
- Участники проекта могут просматривать информацию о проекте, но не могут его редактировать.

2. Редактирование проекта:

Действия:

- Открыть страницу проекта.
- Нажать кнопку "Редактировать".
- Изменить название проекта, описание, начальную или конечную дату.
- Добавить или удалить участников проекта.
- Нажать кнопку "Сохранить".

Примечания:

- Только руководитель проекта может редактировать проект.

3. Изменение состава участников проекта:

Действия:

- Открыть страницу проекта.
- Нажать кнопку "Участники".
- Добавить или удалить участников проекта.
- Нажать кнопку "Сохранить".

Примечания:

- Руководитель проекта может добавлять и удалять участников проекта.
- Участники проекта могут просматривать информацию о проекте, но не могут его редактировать.

4. Удаление проекта:

Действия:

- Открыть страницу проекта.
- Нажать кнопку "Удалить".
- Подтвердить удаление проекта.

Примечания:

- Только руководитель проекта может удалить проект.
- При удалении проекта все его задачи и эпики будут удалены.

5. Просмотр проекта:

Действия:

- Открыть страницу проекта.
- Просмотреть информацию о проекте: название, описание, начальную и конечную дату, руководителя проекта, участников проекта, задачи и эпики.

Примечания:

- Все участники проекта могут просматривать информацию о проекте.

6. Создание эпика:

Действия:

- Открыть страницу проекта.
- Нажать кнопку "Создать эпик".
- Ввести название эпика, описание и приоритет.

- Добавить задачи в эпик.
- Нажать кнопку "Создать".

Примечания:

- Эпики могут быть созданы только в рамках проекта.

7. Удаление эпика:

Действия:

- Открыть страницу эпика.
- Нажать кнопку "Удалить".
- Подтвердить удаление эпика.

Примечания:

- При удалении эпика все его задачи будут удалены.

8. Редактирование эпика:

Действия:

- Открыть страницу эпика.
- Нажать кнопку "Редактировать".
- Изменить название эпика, описание, приоритет.
- Добавить или удалить задачи из эпика.
- Нажать кнопку "Сохранить".

Примечания:

- Эпики могут быть отредактированы только в рамках проекта.

9. Изменение состава задач эпика:

Действия:

- Открыть страницу эпика.
- Нажать кнопку "Задачи".
- Добавить или удалить задачи из эпика.
- Нажать кнопку "Сохранить".

Примечания:

- Задачи могут быть добавлены в эпик только из того же проекта, что и эпик.

10. Просмотр эпика:

Действия:

- Открыть страницу эпика.
- Просмотреть информацию о эпике: название, описание, приоритет, задачи.

Примечания:

- Все участники проекта могут просматривать информацию о эпике.

11. Создание задачи:

Действия:

- Открыть страницу проекта или эпика.
- Нажать кнопку "Создать задачу".

- Ввести название задачи, описание, приоритет, срок выполнения, исполнителя.
- Нажать кнопку "Создать".

Примечания:

- Задачи могут быть созданы как в рамках проекта, так и в рамках эпика.

12. Назначение задачи:

Действия:

- Открыть страницу задачи.
- Нажать кнопку "Выбрать исполнителей".
- Выбрать исполнителя задачи.
- Нажать кнопку "Сохранить".

Примечания:

- Задачи могут быть назначены только участникам проекта.

13. Отслеживание выполнения задачи:

Действия:

- Открыть страницу задачи.
- Просмотреть информацию о задаче: название, описание, приоритет, срок выполнения, исполнителя, статус.

Примечания:

- Исполнитель задачи может изменить статус задачи.

- Пользователь может отслеживать выполнение назначенных ему задач на странице "Мои задачи".
- На странице "Мои задачи" отображается информация о статусе задачи, исполнителе, сроке выполнения и других параметрах.
- Участники проекта могут просматривать историю изменений задачи.
- Информация о выполнении задачи может быть отсортирована и отфильтрована.

14. Удаление задачи:

Действия:

- Открыть задачу.
- Нажать кнопку "Удалить".
- Подтвердить удаление задачи.

Комментарии:

- Удаление задачи доступно только исполнителю, владельцу проекта или администратору.
- При удалении задачи все связанные с ней комментарии и файлы также будут удалены.

15. Редактирование задачи:

Действия:

- Открыть задачу.
- Нажать кнопку "Редактировать".
- Изменить название задачи, описание, исполнителя, статус, приоритет, срок выполнения.

- Нажать кнопку "Сохранить".

Комментарии:

- Редактирование задачи доступно только исполнителю, владельцу проекта или администратору.
- При изменении исполнителя задачи ему будет отправлено уведомление.

16. Комментирование задачи:**Действия:**

- Открыть задачу.
- В поле "Добавить комментарий" ввести текст комментария.
- Нажать кнопку "Добавить".

Комментарии:

- Комментировать задачу могут все участники проекта.
- Все комментарии к задаче будут отображаться в истории задачи.

17. Просмотр задачи:**Действия:**

- Открыть задачу.
- Просмотреть информацию о задаче: название, описание, исполнитель, статус, приоритет, срок выполнения, комментарии, файлы.

Комментарии:

- Просматривать задачу могут все участники проекта.

- Участники проекта могут фильтровать задачи по статусу, исполнителю, приоритету и т.д.

18. Выполнение задачи:

Действия:

- Исполнитель получает уведомление о новой задаче.
- Исполнитель открывает задачу.
- Исполнитель изменяет статус задачи на "В работе".
- Исполнитель добавляет комментарий о ходе выполнения задачи.
- При завершении задачи исполнитель изменяет статус задачи на "Выполнена".
- Проверяющий получает уведомление о выполненной задаче.
- Проверяющий проверяет задачу.
- Проверяющий одобряет или отклоняет задачу.

Комментарии:

- Исполнитель задачи может изменить статус задачи.
- При изменении статуса задачи всем участникам проекта будет отправлено уведомление.

19. Отчетность:

Действия:

- Открыть раздел "Отчеты".
- Выбрать тип отчета: по проектам, по исполнителям, по датам.
- Задать параметры отчета.

- Нажать кнопку "Сформировать отчет".
- Просмотреть отчет.

Комментарии:

- Отчеты доступны только владельцу проекта и администратору.
- Отчеты могут быть экспортированы в PDF или Excel.
- Пользователь может генерировать отчеты о выполненных и просроченных задачах.