Функциональные требования

Сайт для управления работой самозанятых в IT-сфере должен обеспечивать возможность регистрации и аутентификации самозанятых специалистов. Он должен также предоставлять возможность создания профиля, добавления информации о навыках, опыте и выполненных проектах. Сайт должен позволять самозанятым создавать портфолио проектов, принимать заказы, вести коммуникацию с заказчиками и управлять своим рабочим временем.

Нефункциональные требования

Для сайта управления работой самозанятых в IT-сфере важным является обеспечение высокого уровня безопасности данных, включая защиту конфиденциальной информации о проектах и оплатах. Сайт должен обладать высокой производительностью, быстрой загрузкой страниц и отзывчивым интерфейсом. Также важно обеспечить удобство использования сайта для самозанятых специалистов с интуитивно понятным интерфейсом и простым процессом управления проектами и заказами.

Диаграммы

STAR UML — это инструмент для моделирования, который предоставляет возможности для создания диаграмм UML для проектирования программного обеспечения. Он позволяет разработчикам и аналитикам создавать диаграммы классов, последовательностей, деятельности и другие типы диаграмм, которые помогают визуализировать структуру и поведение системы. STAR UML обладает расширенными возможностями и гибкостью для моделирования различных типов систем.

Диаграмма прецедентов

Диаграмма прецедентов — это вид диаграммы в языке моделирования UML, который отображает функциональные требования системы и взаимодействие её пользователей с этой системой. На диаграмме прецедентов показываются актеры (пользователи или внешние системы) и прецеденты (функциональные возможности или действия), которые они могут выполнять. Эта диаграмма помогает визуализировать основные функции системы и её взаимодействие с внешними сущностями. Диаграмма прецедентов отображена на Рис.1.

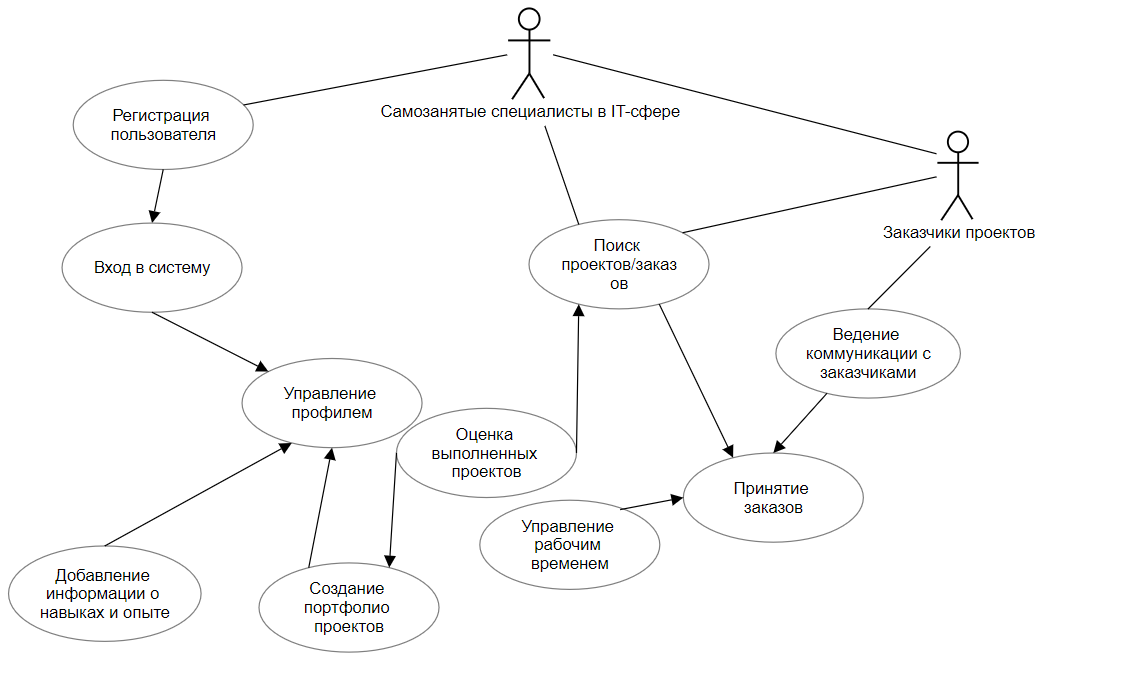


Рис.1 Диаграмма прецедентов

На данной диаграмме прецедентов изображены основные действия и взаимодействия пользователей на сайте для управления работой самозанятых специалистов в IT-сфере. Актерами являются самозанятые специалисты в IT-сфере и заказчики проектов. Регистрация пользователей и вход в систему предшествуют управлению профилем, включая добавление информации о навыках и опыте, а также создание портфолио проектов. Самозанятые специалисты могут принимать заказы, вести коммуникацию с заказчиками и управлять своим рабочим временем. Заказчики могут использовать функцию поиска проектов/заказов и взаимодействовать с выбранными специалистами. Диаграмма предоставляет общее представление о функциональности сайта и взаимодействии его пользователей.

Диаграмма классов

Диаграмма классов — это вид диаграммы в языке моделирования UML, который отображает структуру классов в системе, их атрибуты, методы и взаимосвязи между классами. На диаграмме классов показывается, как классы взаимодействуют друг с другом, какие у них отношения (например, наследование, ассоциация, агрегация) и какие атрибуты и методы они содержат. Эта диаграмма помогает визуализировать структуру системы и её компонентов, что полезно при проектировании и понимании архитектуры программного обеспечения. Диаграмма показана на Рис.2.

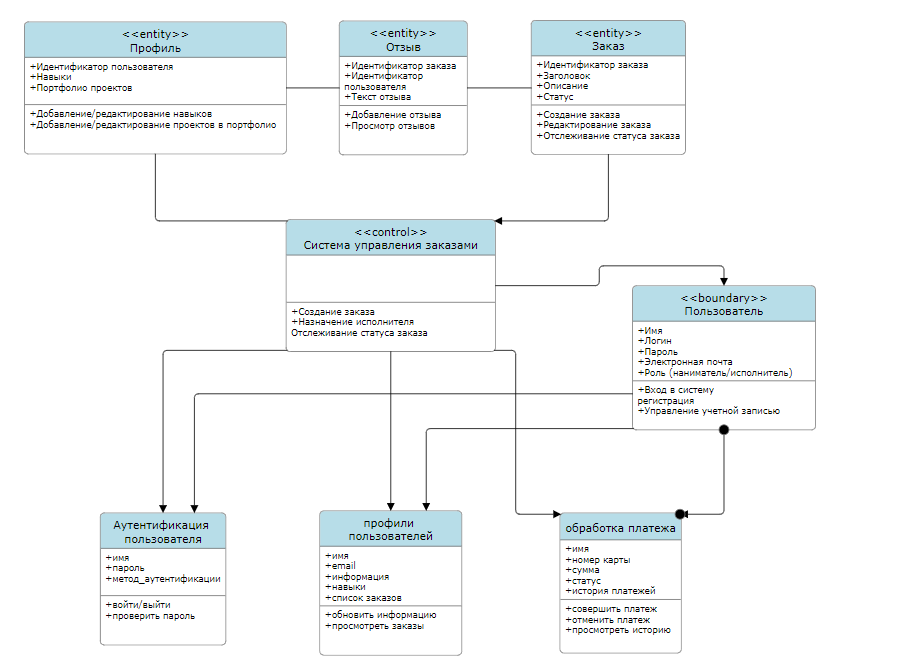


Рис.2 Диаграмма классов.

Эта диаграмма классов отображает структуру системы управления работой самозанятых в IT-сфере. Она показывает, как пользователи взаимодействуют с профилями, заказами и системой управления заказами. Можно увидеть, что классы разделены по типам: профиль, отзыв, заказ. Эти 3 типа объявлены как классы-сущности. Сама система объявлена как контролирующий класс. При помощи системы управления заказами происходит полный поиск информации по пользователю, аутентификации пользователя, профилю, обработке платежа. Эти 4 класса объявлены как граничные, потому что профиль, пользователь, отзыв и заказ взаимодействуют с системой управления заказами.

Диаграмма состояний

**Диаграмма состояний — это графическое представление модели поведения системы, которое отображает все возможные состояния, переходы между ними и действия, вызываемые при переходах. Диаграммы состояний часто используются для моделирования поведения объектов или систем, позволяя легче понять и предсказать их динамику.** Диаграмма состояний приведена на Рис.3.

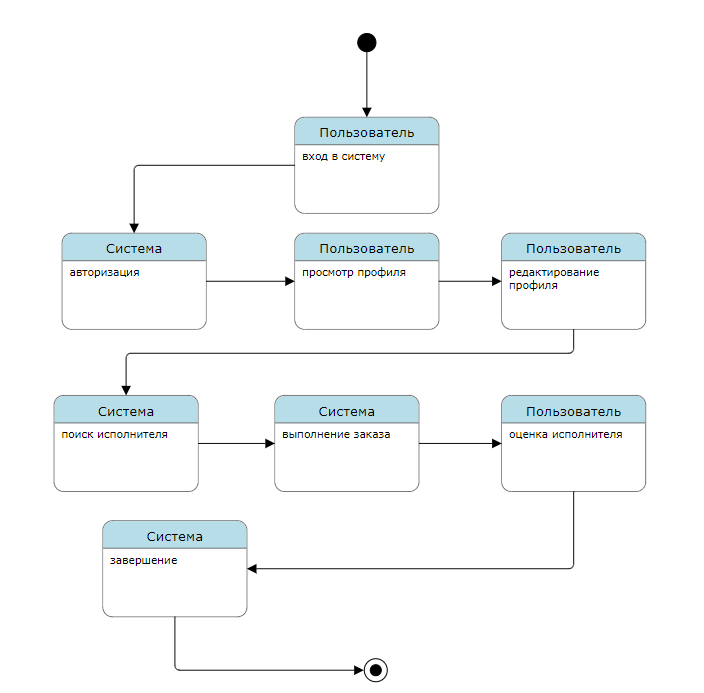


Рис.3 Диаграмма состояний.

Представленная диаграмма состояний отражает основные этапы взаимодействия пользователей с веб-платформой для управления работой самозанятых в IT-сфере. Она начинается с процесса входа пользователя в систему, после чего пользователь может просматривать и редактировать свой профиль, создавать заказы и оценивать исполнителей. Данная диаграмма отражает определенные состояния, которые требуют определенные действия, для перехода в следующее состояние, к примеру, чтобы отредактировать профиль нужно находится в состоянии просмотра профиля, или же для того, чтобы произошла оценка исполнителя нужно выполнить заказ.

Диаграмма активностей

Диаграмма активностей — это вид диаграммы в UML, который позволяет визуализировать последовательность действий или активностей в процессе. Она показывает поток управления или поток данных через серию шагов или активностей, отображая принятие решений, параллельные процессы, временные ограничения и условия. Диаграммы активностей часто используются для моделирования бизнес-процессов, спецификации программного обеспечения или описания операций в системе. Диаграмма показана на Рис.4

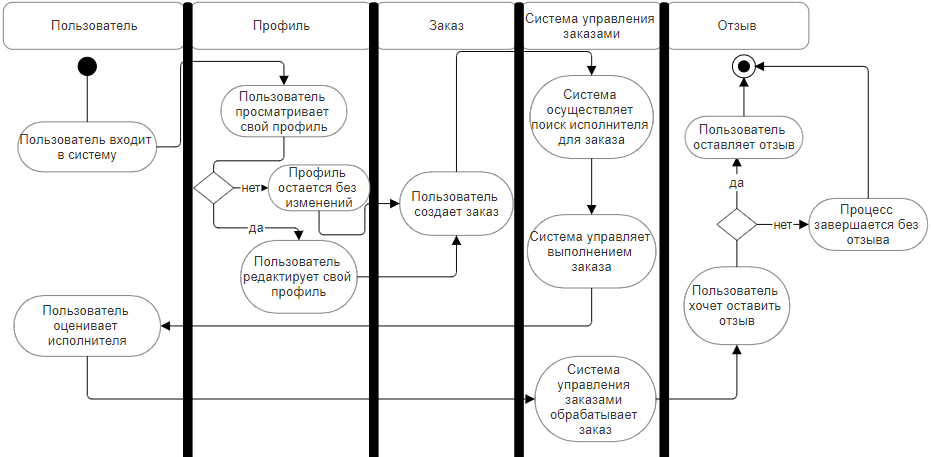


Рис.4 Диаграмма активности.

Эта диаграмма активностей отображает последовательность действий и принимаемых решений в рамках системы управления работой самозанятых в IT-сфере. Пользователь начинает с входа в систему, затем может просматривать и редактировать свой профиль, далее он может принять решение, редактировать свой профиль или нет, если ответ да, то пользователь редактирует свой профиль, если нет, то профиль останется без изменений. Далее пользователь идет создавать заказ, после чего система осуществляет поиск исполнителя для заказа, и система сама управляет выполнением заказа, после чего пользователь оценивает исполнителя. Система управления заказами обрабатывает заказ и задает вопрос пользователи, не хочет ли он оставить отзыв, если да, то пользователь оставляет отзыв, если нет, то процесс завершается без отзыва.

Диаграмма последовательности

Диаграмма последовательности (sequence diagram) — это тип диаграммы в UML, который отображает взаимодействие различных объектов в системе в определенной последовательности для выполнения определенной функциональности. Она позволяет визуально представить обмен сообщениями между объектами в рамках определенного сценария использования или операции. Диаграмма последовательности полезна для понимания взаимодействия между объектами и порядка выполнения операций в системе. Диаграмма последовательности отображена на Рис.5.

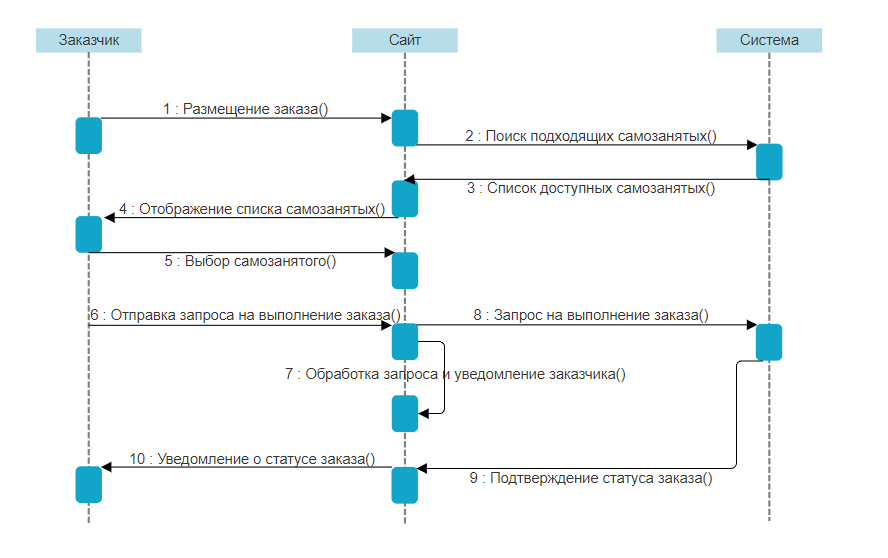


Рис.5 Диаграмма последовательности.

Данная диаграмма последовательности отображает взаимодействие между заказчиком, сайтом для управления работой самозанятых в IT-сфере и системой управления заказами при размещении заказа.

Заказчик размещает заказ на сайте, после чего сайт инициирует поиск подходящих самозанятых через систему управления заказами. Система возвращает список доступных самозанятых на сайт, где заказчик выбирает исполнителя и отправляет запрос на выполнение заказа. Затем сайт обрабатывает запрос и уведомляет как заказчика, так и систему управления заказами о статусе заказа. Система подтверждает статус заказа, и заказчик получает уведомление о его выполнении.

Диаграмма развертывания

Диаграмма развертывания в UML визуализирует физическую структуру системы, показывая, как ее компоненты размещаются на устройствах (например, серверах, компьютерах) и как они взаимодействуют друг с другом. Это помогает понять архитектурные аспекты развертывания системы. Она используется в разработке программного обеспечения для планирования и визуализации размещения компонентов системы на аппаратных устройствах или средах выполнения. Они помогают командам разработки понять, как система будет развернута и какие ресурсы будут использованы. Диаграмма развёртывания отображена на Рис.6.

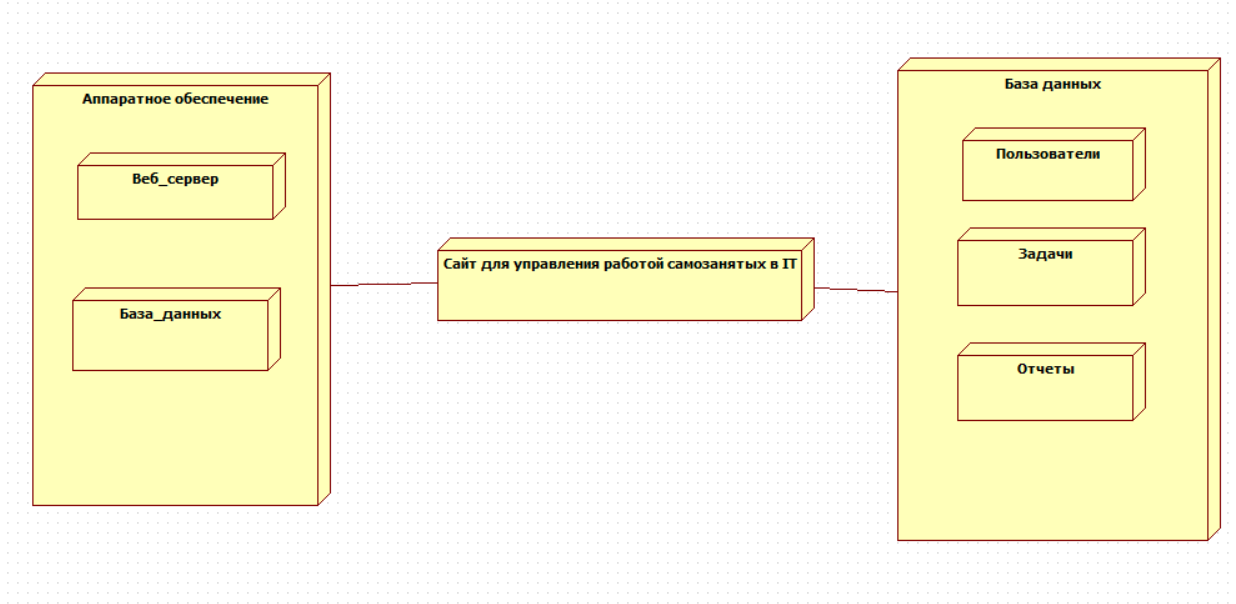


Рис.6 Диаграмма развёртывания

Данная диаграмма развертывания показывает архитектурное размещение компонентов для сайта управления работой самозанятых в IT-сфере. Веб-сервер и база данных размещены на аппаратном обеспечении, с веб-сервером, обеспечивающим HTTP-соединение. Сайт для управления работой самозанятых в IT-сфере размещен на веб-сервере и взаимодействует с базой данных через JDBC для доступа к данным о пользователях, задачах и отчетах.

Диаграмма Компонентов

Диаграмма компонентов в UML представляет собой визуализацию компонентов системы (например, библиотек, исполняемых файлов, модулей), их связей и интерфейсов. Она отображает структуру системы на уровне компонентов и их взаимодействие, что помогает разработчикам понять, как компоненты связаны друг с другом и как они работают вместе для обеспечения функциональности системы. Диаграмма компонентов показана на Рис.7.

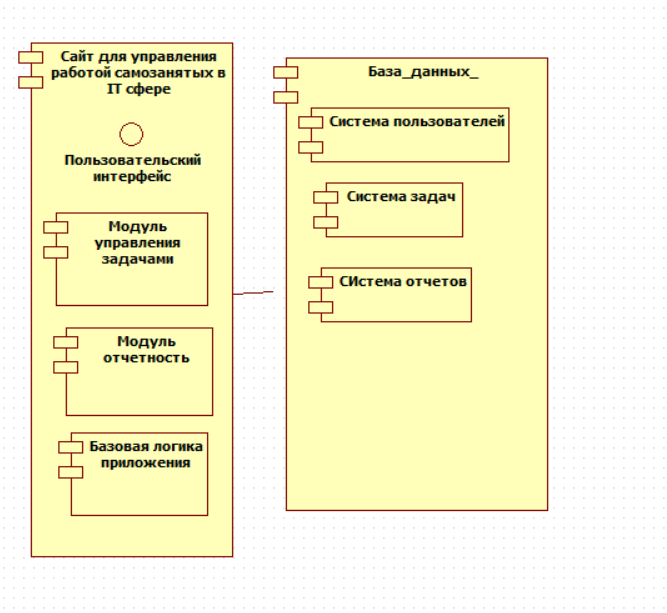


Рис.7 Диаграмма компонентов.

Данная диаграмма компонентов отображает основные компоненты системы "Сайт для управления работой самозанятых в IT-сфере" и их взаимосвязи, такие как пользовательский интерфейс, модуль управления задачами, модуль отчетности, базовая логика приложения. Эти компоненты взаимодействуют через протоколы HTTP и JDBC для обеспечения связи с базой данных, включающей системы пользователей, задач и отчётов.

Базы данных

1. Система пользователей.

Отвечает за хранение информации о зарегистрированных пользователях, включая их личные данные (имя, фамилия, электронная почта), учетные данные (логин, пароль) и дату регистрации.

Обеспечивает аутентификацию пользователей и управление их учетными записями.

2. Система задач:

Занимается отслеживанием задач, их статусом, дедлайнами, приоритетами и назначенными исполнителями.

Позволяет пользователям создавать, назначать и отслеживать задачи, а также контролировать их выполнение.

3. Система отчетов:

Отвечает за фиксацию отчетов о проделанной работе и затраченном времени на задачи.

Позволяет пользователям создавать отчеты, описывающие проделанную работу, и указывать затраченное время для последующего анализа и отчетности.

Каждая система выполняет свои функции для обеспечения эффективного управления работой самозанятых в IT-сфере на сайте.

Пример пользовательского интерфейса

Страница входа Рис.8:

- Поле для ввода логина

- Поле для ввода пароля

- Кнопка "Войти"

- Ссылка на страницу регистрации

Главная страница Рис.9:

- Список задач с отображением статуса, названия и дедлайна

- Фильтры для поиска задач по статусу и приоритету

- Кнопка создания новой задачи

- Ссылки на отчеты и профиль пользователя

Страница создания задачи Рис.10:

- Поля для ввода названия, описания, дедлайна и приоритета задачи

- Выбор исполнителя из списка пользователей

- Кнопка "Создать задачу"

Страница отчетов Рис.11:

- Форма для создания нового отчета с полями для описания работы и затраченного времени

- Список предыдущих отчетов с возможностью просмотра и редактирования

Этот пользовательский интерфейс обеспечивает удобное взаимодействие пользователей с системой управления задачами и отчетностью на сайте для самозанятых в IT-сфере. Схематичные примеры отображения пользовательского интерфейса показаны на Рис 8-11.

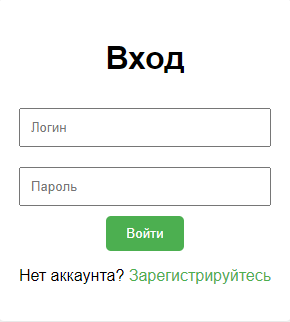


Рис.8 Страница входа

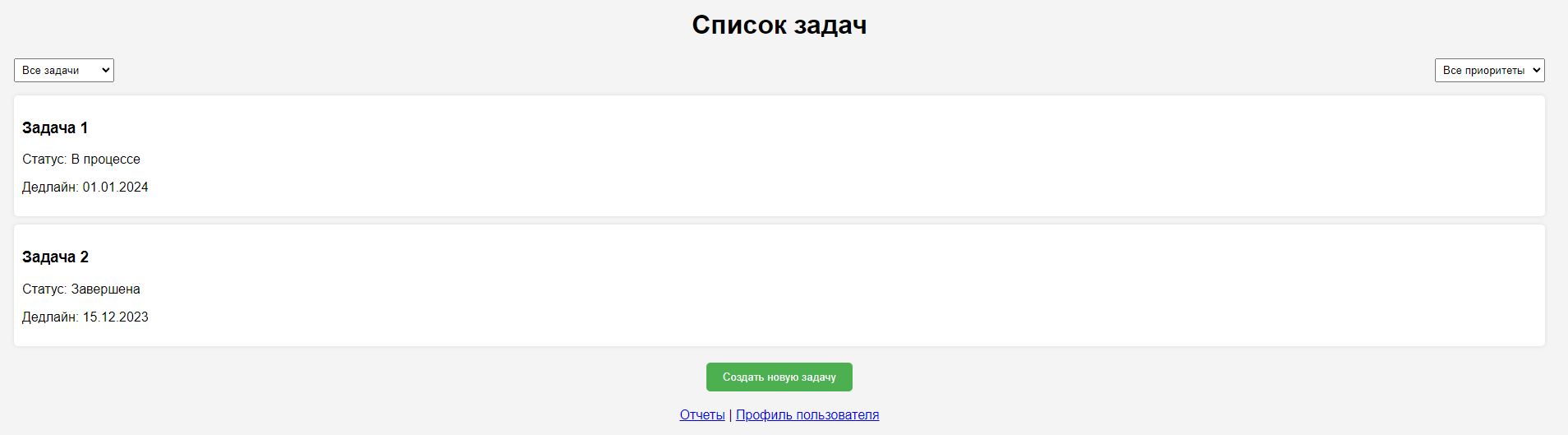


Рис.9 Главная страница

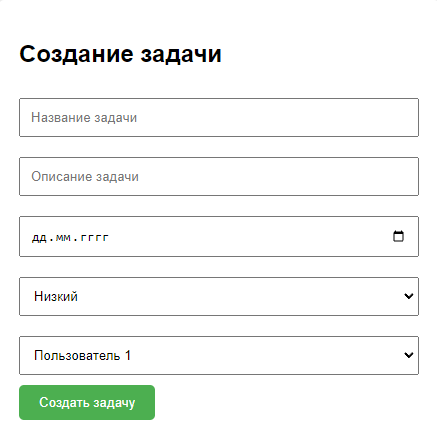


Рис.10 Страница создания задачи

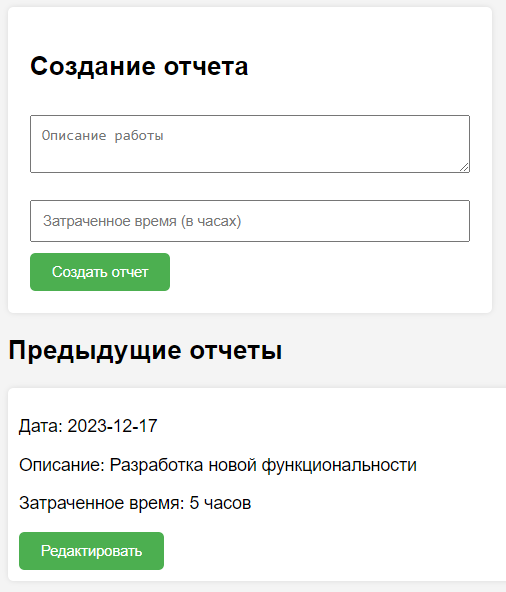


Рис.11 Страница отчетов