МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ01. Разработка модулей программного обеспечения для компьютерных систем**

**Тема: «Разработка программного модуля «Управление задачами разработки»**

Студент

Збруев Антон Владимирович

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики от колледжа:

*Махнев Александр Анатольевич*

Руководитель практики от организации:

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* Кузнецова Елизавета Сергеевна подпись

УТВЕРЖДАЮ:

Директор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Наименование организации

\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись расшифровка

М. П.

2024-2025 уч. год

**Содержание**

Оглавление

[1. Сведения об организации (базе практики) 3](#_Toc194437166)

[2. Разработка и анализ технического задания 5](#_Toc194437167)

[3. Разработка программного модуля по алгоритму в соответствии техническому заданию 12](#_Toc194437168)

[4. Документации на программный модуль 18](#_Toc194437169)

[5. Выполнение отладки программного модуля 32](#_Toc194437170)

[6. Выполнение тестирования программного модуля 34](#_Toc194437171)

[7. Оформление результатов тестирования 36](#_Toc194437172)

[8. Рефакторинг и оптимизации программного кода 38](#_Toc194437173)

[9. Разработка программного модуля для мобильного устройства 41](#_Toc194437174)

[Заключение 46](#_Toc194437175)

[Приложение 1 48](#_Toc194437176)

# **Сведения об организации (базе практики)**

Местом прохождения производственной практики по профессиональному модулю ПМ.01 "Разработка модулей программного обеспечения для компьютерных систем" являлось Общество с ограниченной ответственностью «МастерСофт – Сервис» (сокращенное наименование: ООО «МастерСофт – Сервис»).

**Юридический и фактический адрес организации:** 610017, г. Киров, ул. Маклина, д. 40.

**Краткая характеристика:**

ООО «МастерСофт – Сервис» является аккредитованной IT-компанией, включенной в реестр субъектов малого и среднего предпринимательства (МСП). Основной сферой деятельности компании является разработка программного обеспечения, внедрение и сопровождение систем на платформе 1С:Предприятие, IT-аутсорсинг, предоставление комплексных IT-услуг для бизнеса и т.д.

В ходе практики были предоставлены необходимые ресурсы для выполнения заданий, включая.

Оборудованное рабочее место: Персональный компьютер с установленной операционной системой Windows, необходимыми периферийными устройствами (монитор, клавиатура, мышь) и доступом к локальной сети организации.

Доступ к программному обеспечению:

Платформа "1С:Предприятие 8.3" с возможностью запуска в режиме Конфигуратора для разработки и отладки.

Лицензионные средства разработки и тестирования, используемые в компании.

Стандартный пакет офисных приложений для работы с документацией.

Доступ к информационным ресурсам:

Доступ к тестовой базе данных 1С для разработки, отладки и проверки работоспособности создаваемых модулей без влияния на рабочую систему компании.

Доступ к внутренней базе знаний компании и технической документации.

Доступ к сети Интернет для поиска информации и работы с внешними сервисами.

Консультационная поддержка: Регулярные консультации со стороны руководителя практики от организации, а также возможность обращения за помощью и разъяснениями к другим опытным сотрудникам отдела разработки компании.

**Руководитель практики от организации:** Кузнецова Елизавета Сергеевна

# **Разработка и анализ технического задания**

**1. Техническое Задание.**

**1. Введение**

**1.1. Наименование системы:** Информационная система управления задачами разработки "TaskMaster".

**1.2. Основание для разработки:** Производственная практика студента группы 21П-1 Збруева Антона Владимировича в ООО «МастерСофт – Сервис».

**1.3. Цель разработки:** Создание информационной системы для автоматизации процесса постановки и контроля выполнения задач разработки программного обеспечения, интегрированной с системой управления версиями GitLab и предоставляющей доступ к задачам через мобильное приложение.

**1.4. Задачи разработки:**

* Разработать систему, позволяющую ставить задачи, отслеживать их выполнение.
* Обеспечить возможность интеграции с GitLab для автоматического создания и обновления задач, синхронизации статусов (опционально, по возможности).
* Разработать мобильное приложение для просмотра задач (опционально, по возможности).
* Обеспечить разграничение прав доступа пользователей (Руководитель проекта и Менеджер).
* Реализовать удобный и интуитивно понятный интерфейс.

**2. Требования к системе**

**2.1. Функциональные требования:**

* **2.1.1. Управление проектами:**
  + Создание и редактирование проектов (Руководитель проекта).
  + Удаление проектов (Руководитель проекта).
  + Указание названия, описания, идентификатора проекта в GitLab , руководителя проекта.
  + Возможность указания токена GitLab для доступа к API.
  + Указание статуса проекта.
* **2.1.2. Управление пользователями:**
  + Ведение списка пользователей системы (Руководитель проекта).
  + Назначение пользователям ролей (Руководитель проекта):
    - Руководитель проекта.
    - Менеджер.
  + Указание ФИО, email, телефона.
* **2.1.3. Управление задачами:**
  + Создание задач (Руководитель проекта, Менеджер).
  + Редактирование задач (Руководитель проекта).
  + Удаление задач (Руководитель проекта).
  + Указание проекта, темы, описания, исполнителя, статуса, приоритета, срока выполнения.
  + Автоматическое создание задач в GitLab при создании задачи в 1С .
  + Обновление задач в GitLab при изменении задачи в 1С .
  + Синхронизация статусов задач между 1С и GitLab .
  + Ведение истории изменений статусов задач.
* **2.1.4. Интеграция с GitLab:**
  + Использование GitLab API для создания, обновления и получения информации о задачах.
  + Автоматическое создание задачи в GitLab при создании задачи в 1С.
  + Получение списка проектов GitLab.
  + Реализация общего модуля для взаимодействия с GitLabAPI
  + Реализация регламентного задания для периодической синхронизации данных.
* **2.1.5. Мобильное приложение:**
  + Просмотр списка задач (с возможностью фильтрации по проекту, статусу).
  + Просмотр детальной информации о задаче (включая описание).
* **2.1.6. Отчетность:**
  + Формирование отчетов по проектам и задачам (список задач по проекту, список задач по статусу).

**2.2. Нефункциональные требования:**

* **2.2.1. Надежность:**
  + Система должна обеспечивать надежное хранение данных.
  + Система должна быть устойчива к сбоям.
* **2.2.2. Безопасность:**
  + Доступ к системе должен быть защищен паролем (стандартный механизм 1С).
  + Должно быть реализовано разграничение прав доступа пользователей (Руководитель проекта, Менеджер).
* **2.2.3. Производительность:**
  + Система должна обеспечивать приемлемую скорость работы.
  + Время отклика системы на действия пользователя не должно превышать 3 секунд (ориентировочно).
* **2.2.4. Масштабируемость:**
  + Система должна быть спроектирована с учетом возможного дальнейшего расширения функциональности (добавление новых ролей, интеграции и т.д.).
* **2.2.5. Удобство использования:**
  + Система должна иметь интуитивно понятный интерфейс.
* **2.2.6. Технологии:**
  + Система должна быть разработана на платформе "1С:Предприятие 8.3".
  + Для взаимодействия с GitLab API должен использоваться протокол HTTPS.
  + Мобильное приложение должно быть разработано с использованием средств мобильной разработки 1С.

**3. Состав работ**

1. Анализ требований и проектирование системы.
2. Разработка ER-диаграммы.
3. Разработка структуры базы данных (в терминах 1С).
4. Разработка программных модулей на платформе 1С.
5. Разработка мобильного приложения .
6. Интеграция с GitLab API .
7. Тестирование и отладка системы.
8. Рефакторинг и оптимизация кода.
9. Разработка документации (руководство пользователя).
10. Развертывание системы.

**4. Требования к документации**

* **Руководство пользователя:** Должно содержать описание основных функций системы, сценарии работы с системой (для Руководителя проекта и Менеджера), описание интерфейса.

**5. Порядок контроля и приемки**

1. Промежуточный контроль выполнения работ осуществляется руководителем практики.
2. Приемка системы осуществляется комиссией в составе руководителя практики от колледжа.
3. Критерии приемки:
   * Соответствие системы требованиям технического задания (в реализованном объеме).
   * Отсутствие ошибок в работе системы.
   * Наличие полного комплекта документации.

**6. Термины и определения**

* **GitLab:** Веб-инструмент с открытым исходным кодом для хостинга IT-проектов и совместной разработки, основанный на системе контроля версий Git.
* **API (Application Programming Interface):** Набор способов и правил, по которым различные программы общаются между собой.
* **HTTPS (HyperText Transfer Protocol Secure):** Расширение протокола HTTP, поддерживающее шифрование.
* **ER-диаграмма (Entity-Relationship Diagram):** Диаграмма "сущность-связь", используемая для проектирования баз данных.

**7. Исходные данные**

* Документация по GitLab API.
* Платформа "1С:Предприятие 8.3".
* Средства мобильной разработки 1С.

**Анализ Технического Задания на Разработку Системы "TaskMaster"**

1. **Общая оценка:**

Техническое задание в целом достаточно хорошо структурировано и описывает основной функционал разрабатываемой системы управления задачами. Оно определяет цели, задачи, основные требования (функциональные и нефункциональные), состав работ и другие важные аспекты проекта. Упрощение ролей до "Руководителя проекта" и "Менеджера" делает модель доступа понятной для реализации. Четкое выделение интеграции с GitLab и мобильного приложения как опциональных компонентов является правильным подходом для проекта в рамках производственной практики, позволяя гибко управлять объемом работ.

**2. Сильные стороны ТЗ:**

* **Четко определена цель и задачи:** Понятно, для чего создается система и какие основные проблемы она должна решать.
* **Структурированность:** Документ имеет логичную структуру (Введение, Требования, Состав работ и т.д.), что облегчает его понимание.
* **Разделение требований:** Функциональные и нефункциональные требования выделены в отдельные разделы.
* **Определение ролей и прав:** Достаточно четко разграничены права для двух основных ролей ("Руководитель проекта" и "Менеджер"), что упрощает реализацию механизма доступа.
* **Опциональность сложных компонентов:** Явное указание на опциональность интеграции с GitLab и мобильного приложения позволяет сфокусироваться на ядре системы и добавлять эти компоненты по мере возможности, что реалистично для учебного проекта/практики.
* **Технологические требования:** Указана конкретная платформа разработки (1С:Предприятие 8.3) и связанные технологии (HTTPS, 1С мобильная разработка).
* **Включены важные этапы:** Состав работ включает не только разработку, но и анализ, проектирование, тестирование, рефакторинг и документацию.
* **Детализация некоторых функциональных требований:**
  + Управление задачами (п. 2.1.3): Указано "исполнителя (опционально, если будет интеграция с GitLab)". Неясно, как назначается исполнитель, если интеграции нет. Нужно ли это поле вообще в базовой версии, или оно появляется только при интеграции? Это нужно уточнить.
* **Обработка ошибок интеграции:** В п. 2.1.4 описаны функции интеграции, но не указано, как система должна реагировать на ошибки (например, если GitLab недоступен при попытке создать задачу). Должна ли операция в 1С откатываться, или задача должна создаваться в 1С с пометкой об ошибке отправки? Это важный момент для стабильности.
* **Требования к документации (п. 4):** Указано только "Руководство пользователя". Для отчета по практике часто требуется и техническое описание (структура данных, описание модулей, алгоритмы), что следовало бы отразить.

1. **Полнота и ясность:**

ТЗ достаточно полно описывает основной функционал системы управления задачами. Функциональные требования к интеграции и мобильному приложению описаны на базовом уровне, что соответствует их опциональному статусу. Описание ролей и их основных прав ясное. Нефункциональные требования стандартны и понятны. Некоторые детали (обработка ошибок, конкретный состав отчетов) требуют уточнения.

1. **Реализуемость:**

В рамках производственной практики создание базовой системы управления задачами (управление проектами, пользователями, задачами, история статусов, базовые отчеты) с двумя ролями на платформе 1С является вполне реализуемой задачей. Реализация *полноценной* двусторонней интеграции с GitLab и разработка мобильного приложения могут быть достаточно трудоемкими и зависят от отведенного времени и уровня подготовки, поэтому их опциональный статус оправдан.

1. **Заключение и Рекомендации:**

Данное ТЗ является хорошей основой для разработки системы "TaskMaster" в рамках производственной практики.

# **Разработка программного модуля по алгоритму в соответствии техническому заданию**

На данном этапе производственной практики была выполнена непосредственная разработка программного модуля информационной системы "TaskMaster" на платформе "1С:Предприятие 8.3". Разработка велась в строгом соответствии с ранее утвержденным Техническим Заданием (ТЗ) и разработанными алгоритмами работы основных функций системы.

* 1. **Среда разработки:**

Разработка и отладка программного модуля осуществлялись с использованием конфигуратора платформы "1С:Предприятие 8.3" (8.3.26.1498)

**3.2. Структура Программного Модуля (Объекты Метаданных):**  
В соответствии с проектными решениями и ER-диаграммой, была создана следующая структура объектов метаданных в конфигураторе 1С:

* **Справочники:**
  + Проекты: Для хранения информации о проектах, их параметрах, настройках интеграции с GitLab (ID, Токен) и составе команды (через табличную часть).
  + Пользователи: Для хранения информации о пользователях системы, их контактных данных и назначенных ролях.
* **Перечисления:**
  + РолиВПроекте: Определяет возможные роли пользователей в системе (Руководитель проекта, Менеджер).
  + СтатусыЗадач: Определяет возможные статусы задач (Новая, В работе, На проверке, Выполнена и т.д.).
  + ПриоритетыЗадач: Определяет приоритеты задач (Высокий, Средний, Низкий и т.д.).
  + СтатусыПроектов: Определяет жизненный цикл проекта (Планируется, В разработке, Завершен и т.д.).
* **Документы:**
  + Задача: Основной документ системы, фиксирующий постановку задачи, ее атрибуты (проект, тема, описание, исполнитель, статус, приоритет, срок) и идентификатор задачи в GitLab. Документ является регистратором для регистра сведений ИсторияИзмененийСтатусовЗадач.
* **Регистры сведений:**
  + ИсторияИзмененийСтатусовЗадач: Непериодический регистр, подчиненный регистратору (Документ.Задача), предназначенный для хранения истории смены статусов задач с указанием пользователя и даты изменения.
* **Общие модули:**
  + GitLabAPI (или аналогичное имя): Серверный модуль, инкапсулирующий логику взаимодействия с GitLab API (формирование запросов, отправка, обработка ответов). Вызов функций этого модуля осуществляется из других объектов системы (например, из модуля объекта документа "Задача").
  + РаботаСПользователями (или аналогичное имя, опционально): Модуль для общих функций, связанных с пользователями (например, получение текущего пользователя системы).
* **Роли:**
  + РуководительПроекта: Роль с полными правами на все объекты конфигурации.
  + Менеджер: Роль с ограниченными правами (создание и просмотр задач, без доступа к редактированию/удалению других объектов).
* **Отчеты:**
  + СписокЗадач (и другие отчеты): Реализованы с использованием Системы Компоновки Данных (СКД) для вывода и анализа информации по задачам и проектам.

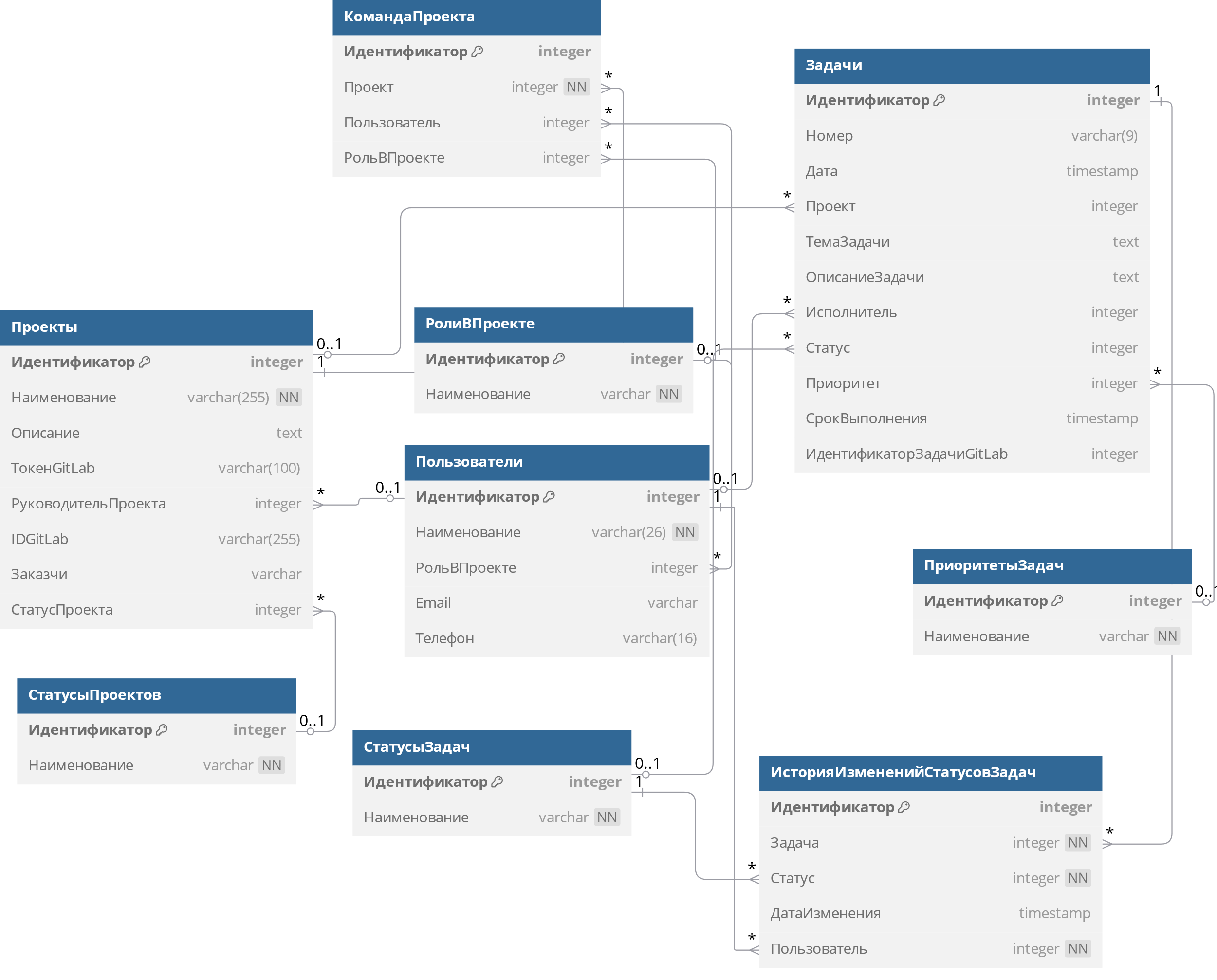


Рисунок 1 – ER-диаграмма базы данных системы "TaskMaster"

На диаграмме представлены основные сущности системы (Проекты, Пользователи, Задачи и т.д.), их атрибуты и связи между ними, реализованные через соответствующие объекты метаданных 1С).

* 1. **Реализация Алгоритмов:**

Ключевые алгоритмы, описанные ранее, были реализованы следующим образом:

* **Алгоритм Создания Новой Задачи:**
  + Реализован через ФормуДокумента документа "Задача", где пользователь вводит данные.
  + Проверка обязательности полей выполняется в обработчике ПередЗаписьюНаСервере формы или в ОбработкеПроверкиЗаполнения модуля объекта.
  + Основная логика (запись в историю, отправка в GitLab, запись ID GitLab) реализована в обработчике ОбработкаПроведения модуля объекта документа "Задача". Вызов функций общего модуля GitLabAPI осуществляется из этого обработчика.
* **Алгоритм изменения статуса задачи:**
  + Пользователь изменяет статус в ФормеДокумента документа "Задача".
  + При проведении документа в обработчике ОбработкаПроведения выполняется запись *новой* строки в регистр ИсторияИзмененийСтатусовЗадач и вызывается функция обновления задачи в GitLab через общий модуль GitLabAPI.
* **Алгоритм синхронизации с GitLab:**
  + Реализован в виде регламентного задания (если требовалось) или обработки для ручного запуска. Логика запроса данных из GitLab, сравнения статусов и обновления документов "Задача" в 1С инкапсулирована в соответствующей процедуре общего модуля GitLabAPI или модуля обработки/регламентного задания.
  1. **Реализация пользовательского интерфейса:**

Для взаимодействия пользователей с системой были разработаны управляемые формы:

* **Формы списков:** Для справочников (Проекты, Пользователи) и документа (Задача) для отображения перечня объектов и навигации.
* **Формы элементов:** Для справочников (Проекты, Пользователи) для ввода и редактирования данных конкретного проекта или пользователя.
* **Форма документа:** Для документа Задача для ввода и редактирования всех реквизитов задачи.
* **Формы отчетов:** Для настройки параметров и вывода результатов отчетов.

Элементы на формах сгруппированы по смыслу для обеспечения интуитивно понятного интерфейса в соответствии с требованиями ТЗ.

* 1. **Реализация разграничения доступа:**

Разграничение доступа реализовано с помощью механизма ролей платформы "1С:Предприятие". Для ролей РуководительПроекта и Менеджер настроены соответствующие права доступа к объектам метаданных (справочникам, документам, регистрам, отчетам), как указано в ТЗ. Руководитель проекта имеет полные права, Менеджер имеет права на создание и просмотр задач, но лишен прав на изменение/удаление других объектов и задач.

* 1. **Соответствие техническому заданию:**

Разработка программного модуля "TaskMaster" производилась в соответствии с функциональными и нефункциональными требованиями, изложенными в Техническом Задании. Реализованы все обязательные функции, связанные с управлением проектами, пользователями и задачами, а также механизм разграничения доступа.

* 1. **Реализация Вариантов Использования (Use Cases):**

Разработанный функционал и пользовательский интерфейс обеспечивают выполнение вариантов использования, определенных на этапе проектирования.

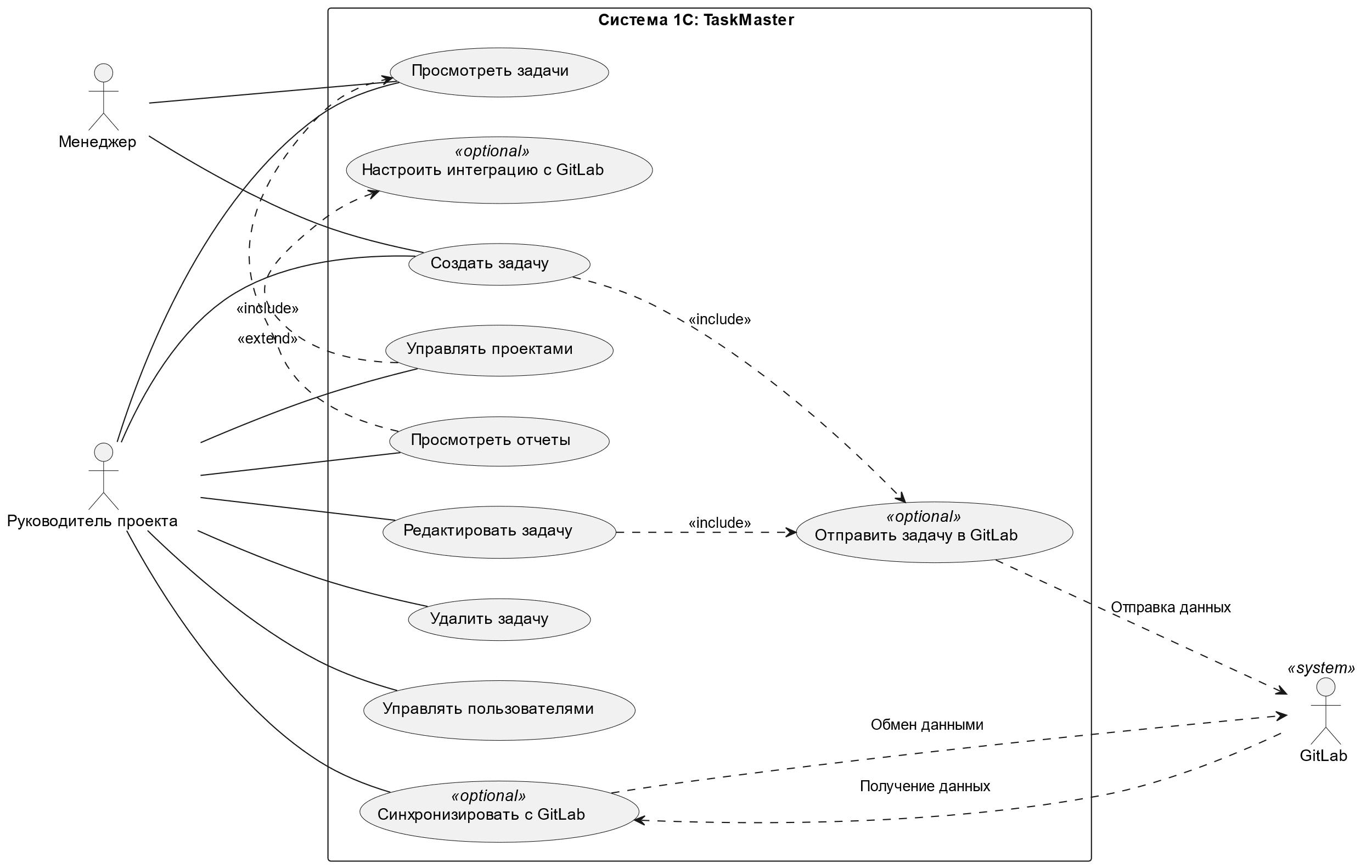
****

Рисунок 2 - Диаграмма вариантов использования

На диаграмме показаны акторы системы ("Менеджер", "Руководитель проекта") и их взаимодействие с основными функциями ("Создать задачу", "Управлять проектами" и т.д.), а также связи между вариантами использования (<<include>>, <<extend>>)).

Например, вариант использования "Создать задачу" реализуется через ФормуДокумента документа "Задача" и логику в его ОбработкеПроведения, доступен ролям "Менеджер" и "Руководитель проекта". Вариант "Управлять проектами" реализуется через формы справочника "Проекты" и доступен только роли "Руководитель проекта".

* 1. **Качество кода:**

При разработке уделялось внимание качеству кода: использовались осмысленные имена переменных и процедур/функций, добавлялись комментарии к сложным участкам кода, применялись стандартные приемы программирования на платформе 1С.

* 1. **Заключение по этапу разработки:**

По итогам данного этапа был разработан программный модуль "TaskMaster", реализующий основную функциональность, описанную в Техническом Задании и спроектированную на предыдущих этапах. Создана структура данных, реализованы основные алгоритмы, настроен пользовательский интерфейс и разграничение доступа. Модуль готов к этапам отладки и тестирования.

# **Документации на программный модуль**

**1. Введение**

**1.1. Назначение документа**

Настоящее руководство пользователя предназначено для ознакомления пользователей с основными функциями и правилами работы в информационной системе управления задачами разработки "TaskMaster".

* 1. **Назначение системы**

Система "TaskMaster" предназначена для автоматизации процесса постановки, учета и контроля выполнения задач в рамках проектов разработки программного обеспечения. Система позволяет управлять проектами, пользователями, задачами, отслеживать историю изменения статусов задач и формировать отчетность. Система интегрирована с платформой GitLab для автоматической передачи информации о задачах.

* 1. **Уровень подготовки пользователя**

Пользователь должен обладать базовыми навыками работы с персональным компьютером под управлением ОС Windows и иметь общее представление о работе с приложениями на платформе "1С:Предприятие". Специальных знаний в области программирования не требуется, за исключением понимания терминологии управления проектами и задачами.

**1.4. Условные обозначения**

* **РП:** Действия, доступные пользователю с ролью "Руководитель проекта".
* **МН:** Действия, доступные пользователю с ролью "Менеджер".
* **[Название кнопки/элемента]:** Обозначение элемента интерфейса (кнопки, поля ввода, пункта меню).

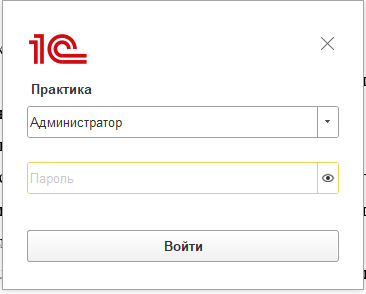
**2. Начало работы**

**2.1. Запуск системы**

Запуск системы "TaskMaster" осуществляется стандартным способом для приложений "1С:Предприятие" через ярлык на рабочем столе или меню "Пуск".

**2.2. Авторизация**

При запуске системы откроется окно выбора информационной базы и окно авторизации.

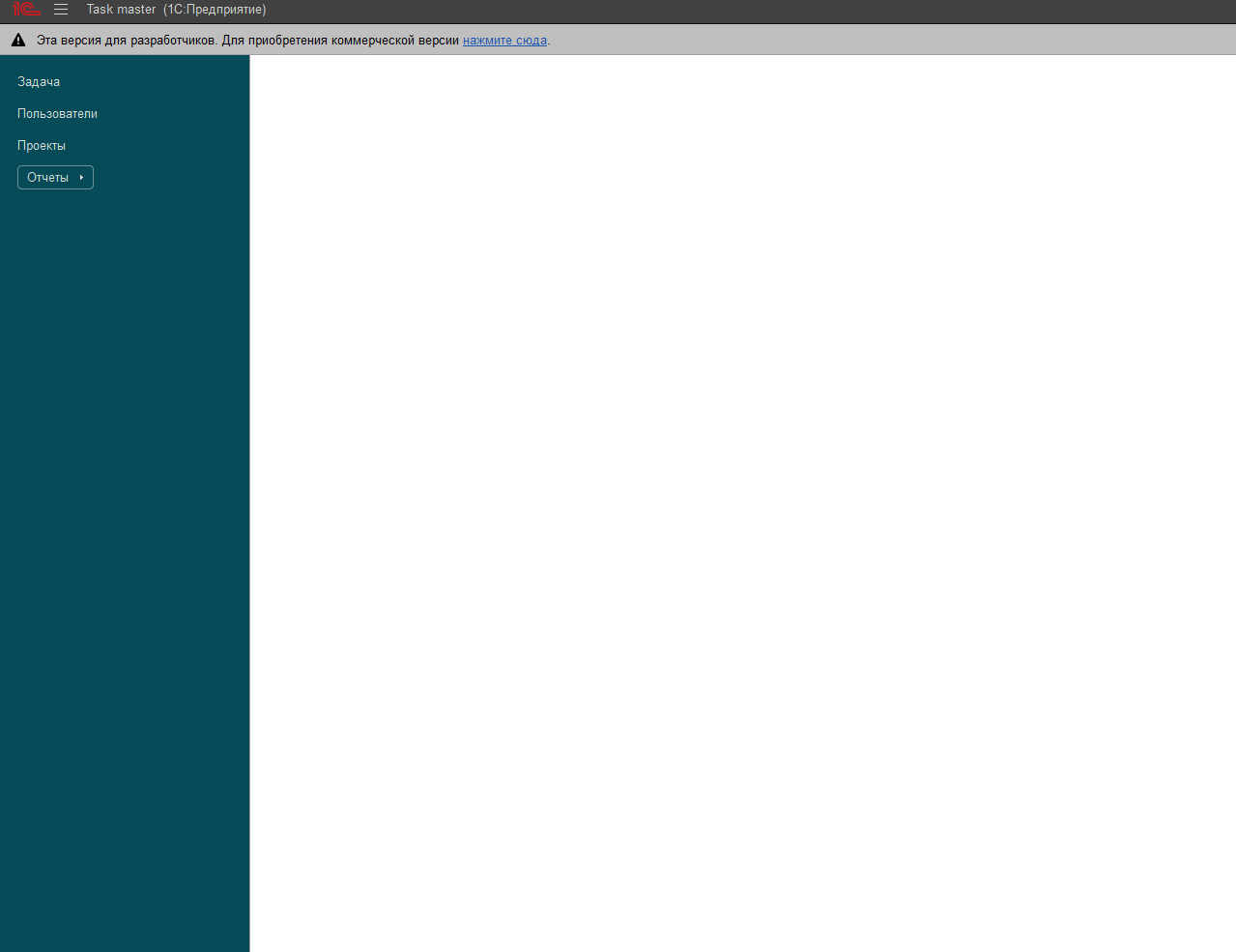
****  
Рисунок 1 – Окно авторизации

Введите свой **Логин** и **Пароль**, предоставленные администратором системы, и нажмите кнопку **[OK]**.

**2.3. Общий вид интерфейса**

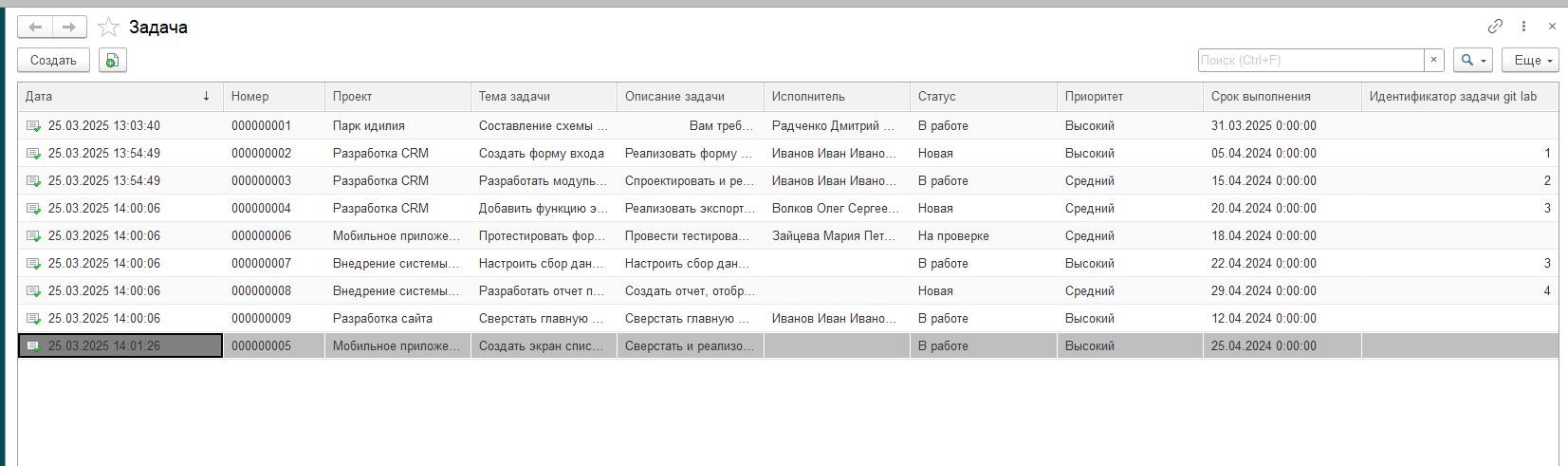
После успешной авторизации откроется основное окно программы. Интерфейс может незначительно отличаться в зависимости от назначенной пользователю роли. Основные элементы интерфейса:

* Панель разделов (слева): для навигации по основным блокам системы (Задачи, Проекты, Пользователи, Отчеты и т.д.).
* Рабочая область: отображает списки, формы или отчеты выбранного раздела.
* Панель инструментов: содержит основные команды для текущего раздела.

****  
Рисунок 2 – Основное окно программы

**3. Работа с задачами**

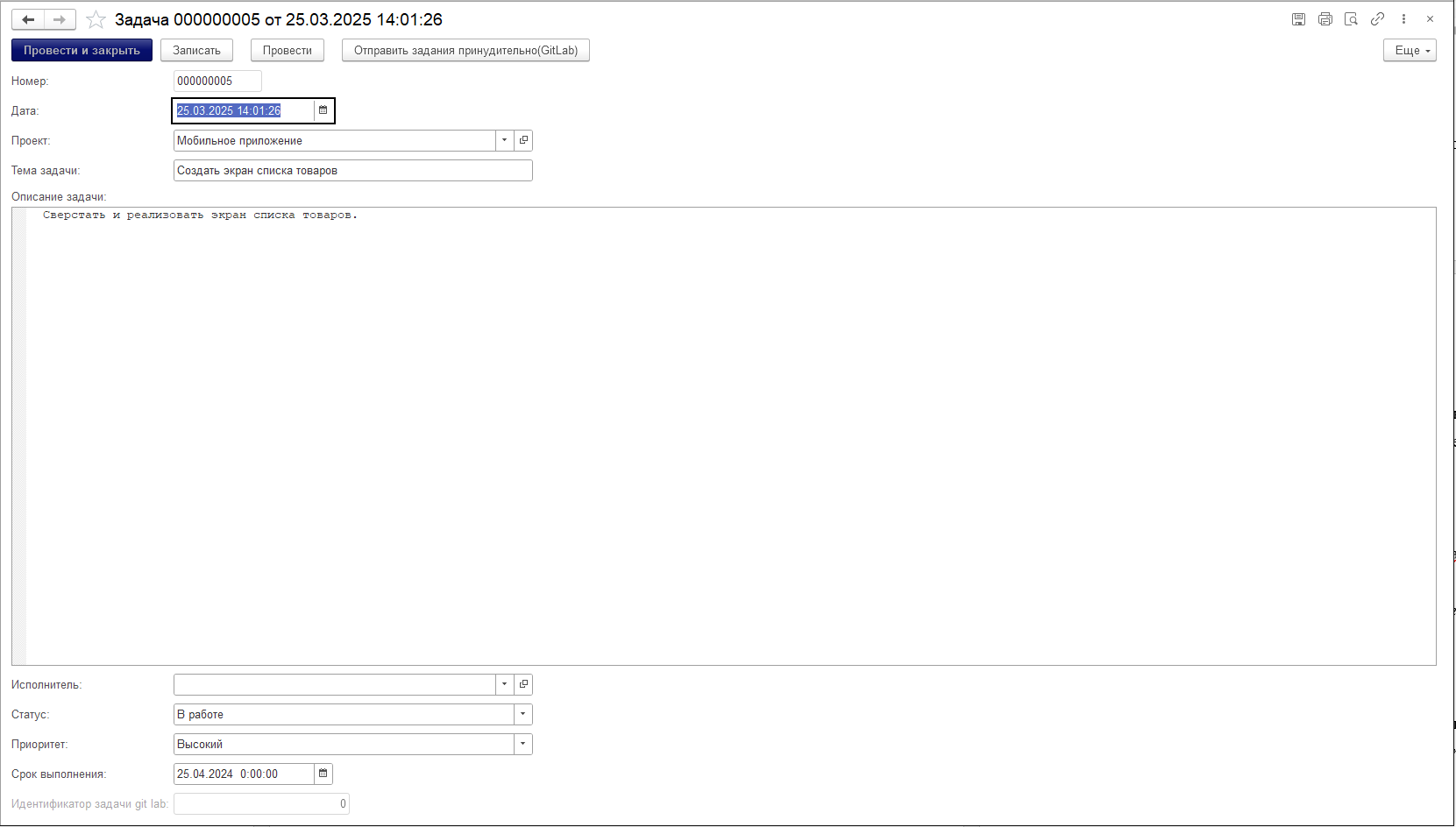
**3.1. Просмотр списка задач (Доступно: РП, МН)**  
Для просмотра списка задач перейдите в раздел **[Название раздела, например, "Задачи" или "Управление задачами"]**. В рабочей области отобразится список всех доступных вам задач.

****  
Рисунок 3 – Список задач

Вы можете использовать стандартные механизмы 1С для поиска, сортировки и отбора задач в списке. Для просмотра детальной информации дважды щелкните по нужной задаче или выберите ее и нажмите **[Enter]** или кнопку **[Изменить]** (доступно только РП).

**3.2. Создание новой задачи (Доступно: РП, МН)**

1. В списке задач нажмите кнопку **[Создать]**.
2. Откроется форма создания новой задачи.

****  
*Рисунок 4 – Форма документа "Задача"*

1. Заполните поля:
   * **[Проект]:** Выберите проект из справочника "Проекты". (Обязательно)
   * **[ТемаЗадачи]:** Введите краткое название задачи. (Обязательно)
   * **[ОписаниеЗадачи]:** Введите подробное описание.
   * **[Исполнитель]:** Выберите пользователя-исполнителя.
   * **[Приоритет]:** Выберите приоритет задачи.
   * **[СрокВыполнения]:** Укажите дату и время срока выполнения.
   * Поле **[Статус]** заполнится автоматически ("Новая").
2. Нажмите кнопку **[Провести и закрыть]**. Задача будет создана в 1С и отправлена в GitLab (если настроена интеграция).

**3.3. Редактирование задачи (Доступно: РП)**

1. В списке задач выберите нужную задачу и откройте ее (двойной клик или кнопка **[Изменить]**).
2. Внесите необходимые изменения в поля формы (Рисунок 4).
3. Нажмите кнопку **[Провести и закрыть]**. Изменения будут сохранены в 1С и отправлены в GitLab (если настроена интеграция и задача уже была отправлена).

**3.4. Изменение статуса задачи (Доступно: РП)**

1. Откройте задачу для редактирования (см. п. 3.3).
2. В поле **[Статус]** выберите новое значение из выпадающего списка.
3. Нажмите кнопку **[Провести и закрыть]**. Изменение статуса будет зафиксировано в истории и отправлено в GitLab.

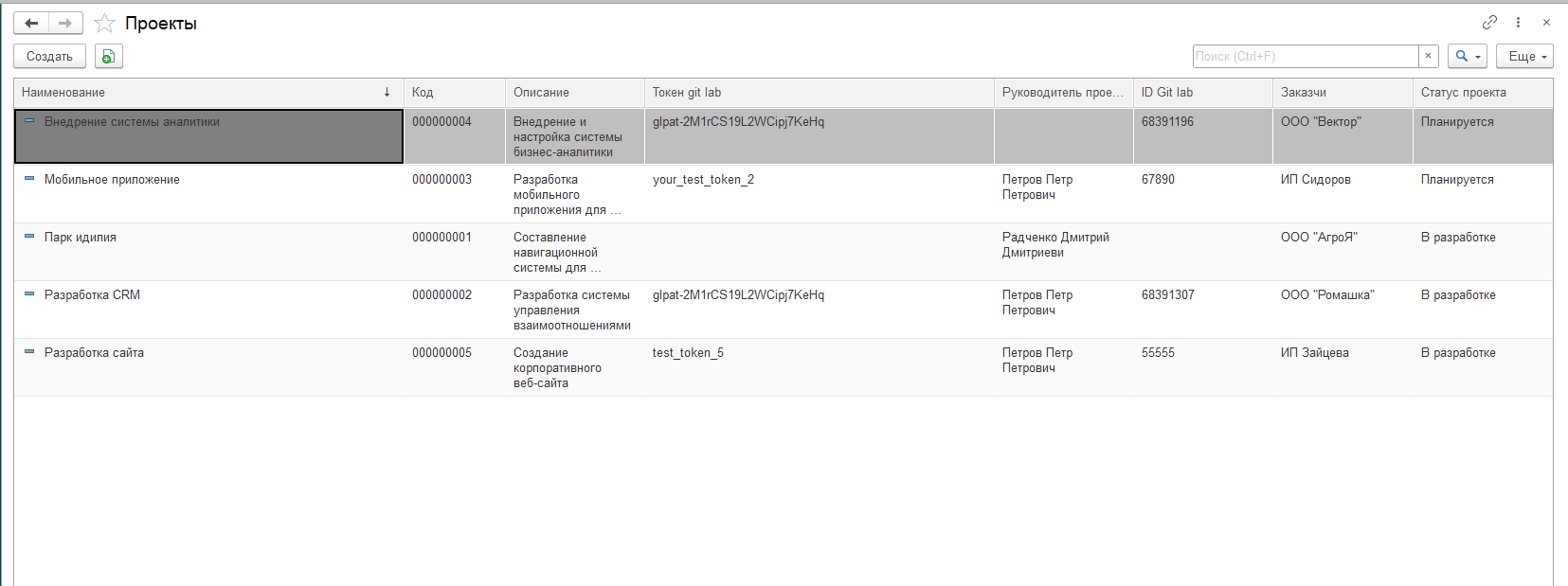
**3.5. Удаление задачи (Пометка на удаление) (Доступно: РП)**

1. В списке задач выделите задачу, которую необходимо удалить.
2. Нажмите кнопку **[Пометить на удаление / Снять пометку]** (может быть в меню "Еще").
3. Подтвердите действие. Задача будет помечена на удаление. Примечание: Физическое удаление помеченных объектов выполняется администратором.

**4. Работа с проектами (Доступно: РП)**

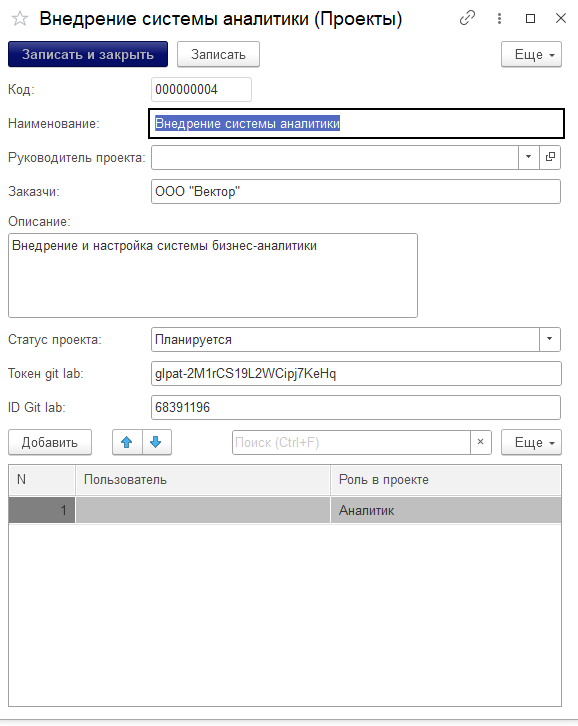
**4.1. Просмотр списка проектов**

Перейдите в раздел **[Название раздела, например, "Проекты"]**. Отобразится список проектов.

****  
Рисунок 5 – Список проектов

**4.2. Создание нового проекта**

1. В списке проектов нажмите кнопку **[Создать]**.
2. Откроется форма создания нового проекта.

****  
Рисунок 6 – Форма справочника "Проект"

1. Заполните поля: **[Наименование]**, **[Описание]**, **[РуководительПроекта]**, **[СтатусПроекта]**, **[Заказчи]** (если требуется).
2. Нажмите **[Записать и закрыть]**.

**4.3. Редактирование проекта**

1. Откройте нужный проект из списка.
2. Внесите изменения в поля формы (Рисунок 6).
3. Нажмите **[Записать и закрыть]**.

**4.4. Управление командой проекта**

1. Откройте проект для редактирования.
2. Перейдите на вкладку **[Команда проекта]** (или найдите соответствующую табличную часть на форме).
3. Для добавления участника нажмите **[Добавить]**, выберите пользователя и его роль в данном проекте.
4. Для удаления участника выделите строку и нажмите **[Удалить]**.
5. Нажмите **[Записать и закрыть]** для сохранения изменений.

**4.5. Настройка интеграции с GitLab**

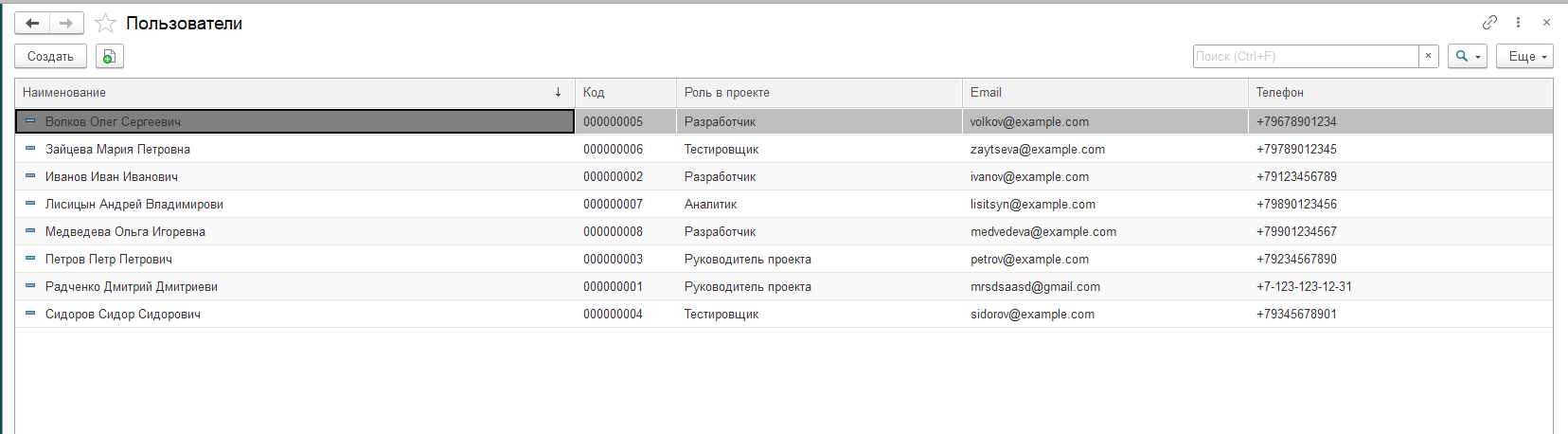
1. Откройте проект для редактирования.
2. Найдите поля **[IDGitLab]** и **[ТокенGitLab]**.
3. Введите корректный ID проекта из GitLab и персональный токен доступа.
4. Нажмите **[Записать и закрыть]**. Внимание: Токен доступа является конфиденциальной информацией!

**4.6. Удаление проекта (Пометка на удаление)**

1. В списке проектов выделите проект.
2. Нажмите **[Пометить на удаление / Снять пометку]**.
3. Подтвердите действие. Примечание: Проект не может быть удален, если с ним связаны задачи.

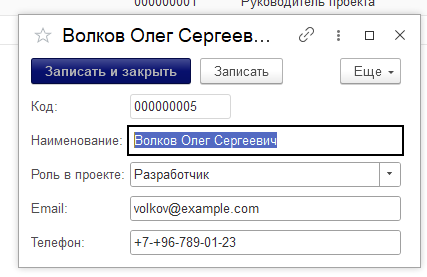
**5. Работа с пользователями (Доступно: РП)**

**5.1. Просмотр списка пользователей**  
Перейдите в раздел **[Название раздела, например, "Администрирование" или "Пользователи"]**. Отобразится список пользователей системы.

****  
Рисунок 7 – Список пользователей

**5.2. Создание нового пользователя**

1. В списке пользователей нажмите **[Создать]**.
2. Откроется форма создания пользователя.

****  
Рисунок 8 – Форма справочника "Пользователь"

1. Заполните поля: **[Наименование (ФИО)]**, **[Email]**, **[Телефон]**.
2. Выберите **[РольВПроекте]** ("Руководитель проекта" или "Менеджер").
3. Нажмите **[Записать и закрыть]**.

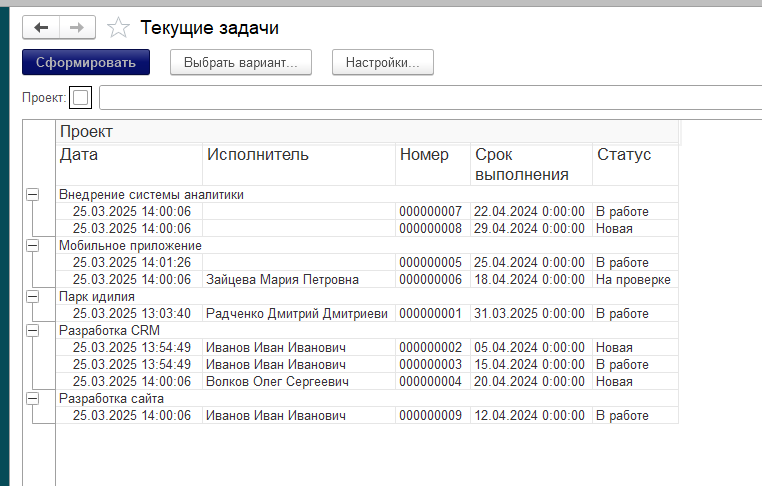
**5.3. Редактирование пользователя**

1. Откройте пользователя из списка.
2. Внесите изменения в поля формы (Рисунок 8).
3. Нажмите **[Записать и закрыть]**.

**6. Работа с отчетами (Доступно: РП)**

**6.1. Формирование отчета "Список задач"**

1. Перейдите в раздел **[Название раздела, например, "Отчеты"]**.
2. Выберите отчет **[Список задач]**.
3. Откроется форма отчета.
4. Нажмите кнопку **[Сформировать]**. Отчет будет отображен в рабочей области.

****  
Рисунок 9 – Пример отчета "Список задач"

**6.2. Настройка отборов и группировок в отчете**

1. В форме сформированного отчета нажмите кнопку **[Настройки]**.
2. Откроется окно настроек варианта отчета (СКД).
3. На вкладке **[Отборы]** можно добавить или изменить условия фильтрации данных (например, по Проекту, Статусу, Исполнителю).
4. На вкладке **[Структура]** можно добавить или изменить группировки строк (например, сгруппировать по Проекту, затем по Статусу).
5. После внесения изменений нажмите **[Завершить редактирование]**, а затем **[Сформировать]** для обновления отчета.

**6.3. Интеграция с GitLab**

Система "TaskMaster" поддерживает интеграцию с сервисом GitLab для автоматической отправки созданных и измененных задач, а также для синхронизации их статусов.

**6.3.1. Настройка интеграции для проекта (Доступно: РП)**

Для того чтобы задачи по конкретному проекту отправлялись в GitLab, необходимо выполнить настройку интеграции в карточке проекта:

1. Откройте справочник "Проекты" (см. п. 4.1).
2. Откройте нужный проект для редактирования.
3. Найдите поля **[IDGitLab]** и **[ТокенGitLab]** (см. Рисунок 6).
4. В поле **[IDGitLab]** введите числовой идентификатор соответствующего проекта в GitLab.
5. В поле **[ТокенGitLab]** введите персональный токен доступа (Personal Access Token), сгенерированный в GitLab и имеющий права на работу с задачами (issues) в указанном проекте. Внимание: Токен доступа является конфиденциальной информацией!
6. Нажмите **[Записать и закрыть]**.

После выполнения этих настроек задачи, создаваемые или изменяемые в данном проекте в 1С, будут автоматически отправляться или обновляться в GitLab при проведении документа "Задача".

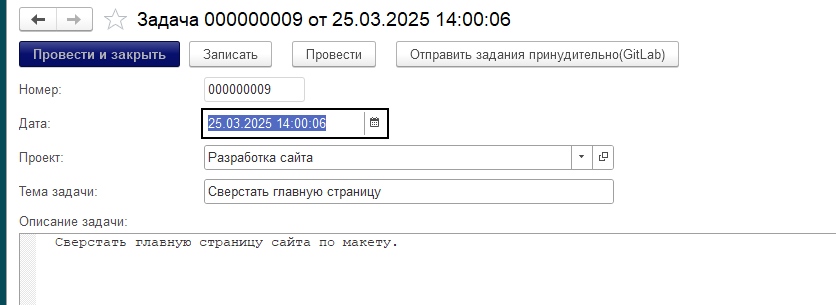
**6.3.2. Отправка задачи в GitLab при создании/изменении**  
При нажатии кнопки **[Провести и закрыть]** в форме документа "Задача" (см. п. 3.2, 3.3, 3.4), если для связанного проекта настроена интеграция (заполнены ID и Токен), система автоматически попытается:

* **Создать** новую задачу в GitLab, если поле **[ИдентификаторЗадачиGitLab]** в документе 1С было пустым. В случае успеха, полученный внутренний ID задачи GitLab (iid) будет записан в это поле.
* **Обновить** существующую задачу в GitLab (например, изменить статус или описание), если поле **[ИдентификаторЗадачиGitLab]** уже заполнено.

В случае ошибок при взаимодействии с GitLab (неверный токен, недоступность сервера, неверный ID проекта и т.д.) проведение документа "Задача" в 1С будет отменено, и пользователю будет выведено соответствующее сообщение об ошибке.

**6.3.3. Принудительная отправка/обновление задачи в GitLab (Доступно: РП)**

В форме документа "Задача" для роли "Руководитель проекта" доступна кнопка **[Отправить в GitLab]**.

****  
Рисунок 10 – Кнопка принудительной отправки задачи в GitLab

Нажатие этой кнопки инициирует немедленную попытку отправки/обновления данных текущей задачи в GitLab, независимо от проведения документа. Это может быть полезно, если:

* Задача была создана ранее без отправки (например, интеграция была настроена позже).
* Произошла ошибка при автоматической отправке во время проведения, и ее нужно повторить после устранения причины.
* Нужно принудительно синхронизировать описание или другие данные.

Система выполнит те же действия по отправке/обновлению, что и при проведении, и сообщит пользователю о результате операции.

**6.3.4. Регламентное задание синхронизации статусов (Автоматическое выполнение)**

В системе настроено (или может быть настроено администратором) регламентное задание **РегламентПоОтправкеЗаданий**, которое периодически (например, каждые 5-10 минут) выполняет следующие действия:

1. Находит задачи в 1С, которые связаны с GitLab (имеют ИдентификаторЗадачиGitLab) и для проектов которых настроена интеграция.
2. Для каждой такой задачи запрашивает у GitLab API ее текущий статус.
3. Сравнивает статус в GitLab со статусом в 1С.
4. Если статусы отличаются, система автоматически обновляет статус задачи в 1С, чтобы он соответствовал статусу в GitLab, и создает соответствующую запись в регистре "ИсторияИзмененийСтатусовЗадач" (с пользователем "Система" или аналогичным).

Это позволяет поддерживать актуальность статусов задач в 1С, даже если они были изменены непосредственно в GitLab разработчиком. Пользователю не требуется выполнять никаких действий для этой синхронизации. Информацию о работе регламентного задания и возможных ошибках можно найти в Журнале регистрации 1С.

**6.3.5. Рабочий процесс разработчика в GitLab**

Важно понимать, что непосредственные исполнители задач (разработчики, тестировщики и т.д.), как правило, **не работают** в системе "TaskMaster" (1С). Их основной рабочий инструмент – это **GitLab**.

После того как задача создана или обновлена в "TaskMaster" и успешно отправлена в GitLab:

1. **Отображение задачи в GitLab:** Задача появляется в списке задач (Issues) соответствующего проекта в GitLab. Часто команды используют доски задач (Issue Boards), настроенные как Kanban-доски, где задачи отображаются в виде карточек в колонках, соответствующих этапам работы (например, "К выполнению / To Do", "В работе / In Progress", "На проверке / Review", "Готово / Done").
2. **Взятие задачи в работу:** Разработчик, готовый приступить к задаче, обычно выполняет следующие действия непосредственно в интерфейсе GitLab:
   * Назначает задачу на себя (Assignee).
   * Перемещает карточку задачи на доске в колонку "В работе / In Progress" (или аналогичную). Это действие в GitLab обычно автоматически изменяет статус или метку (label) задачи.
3. **Завершение работы:** По мере выполнения и завершения работы над задачей, разработчик перемещает ее карточку на доске задач GitLab в соответствующие колонки ("На проверке / Review", "Готово / Done") или закрывает задачу (Close issue).
4. **Синхронизация статуса в "TaskMaster":** Регламентное задание синхронизации (описанное в п. 6.3.4) обнаруживает изменение статуса задачи в GitLab (например, она стала "closed" или получила определенную метку) и автоматически обновляет поле [Статус] в соответствующем документе "Задача" в системе "TaskMaster" (1С).

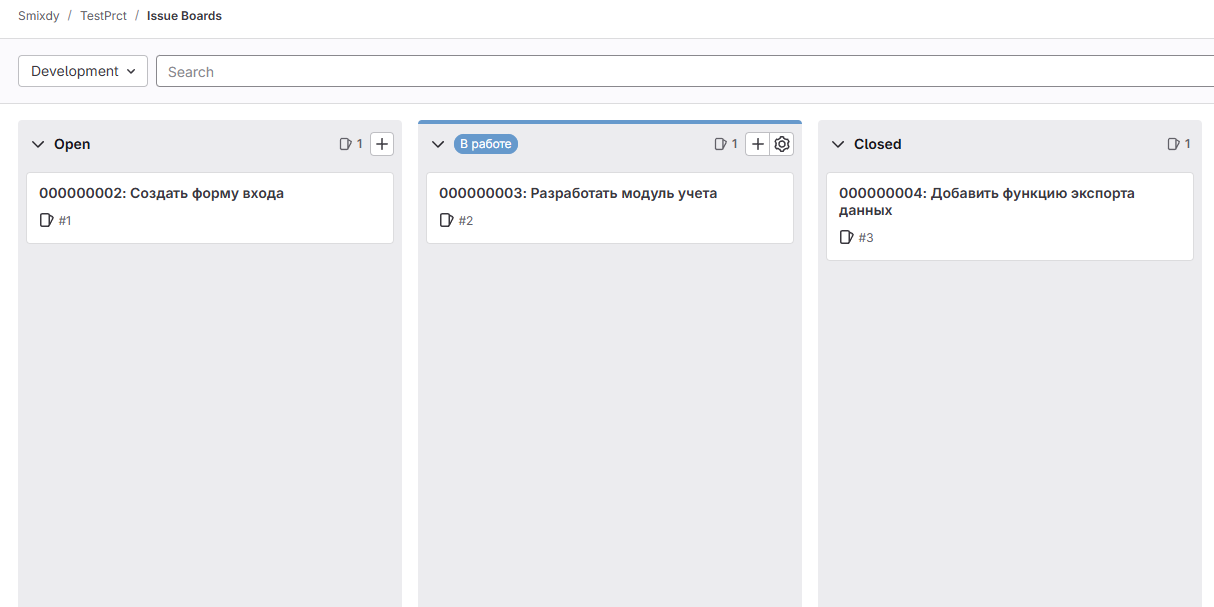


Рисунок 11 – Пример канбан доски разработчика

Таким образом, система "TaskMaster" служит для постановки, первичного учета и контроля задач на уровне менеджеров и руководителей, а GitLab – для непосредственной работы исполнителей и отслеживания технического прогресса. Интеграция обеспечивает синхронизацию ключевой информации (прежде всего, статуса) между этими двумя системами.

**7. Завершение работы**

Для выхода из системы "TaskMaster" закройте основное окно программы стандартным способом (нажать крестик в правом верхнем углу окна).

# **Выполнение отладки программного модуля**

**1. Цель отладки:**

Выявление и устранение ошибок, несоответствий и логических неточностей в разработанном программном модуле "TaskMaster" для обеспечения его корректной и стабильной работы в соответствии с техническим заданием.

1. **Используемые средства:**

Для проведения отладки использовались штатные средства отладки платформы "1С:Предприятие 8.3", включая:

* **Отладчик:** Основной инструмент для пошагового выполнения кода, установки точек останова и анализа состояния программы.
* **Табло:** Для просмотра и изменения значений переменных и выражений в процессе выполнения кода.
* **Стек вызовов:** Для отслеживания последовательности вызовов процедур и функций.
* **Замер производительности:** Для выявления узких мест и неоптимального кода.
* **Журнал регистрации:** Для анализа системных событий и сообщений об ошибках.

1. **Проведенные работы:**

В ходе отладки были выполнены следующие основные действия:

* **Пошаговое выполнение кода:** Проанализирована логика работы ключевых процедур и функций в модулях объектов (Документ.Задача, Справочник.Проекты, Справочник.Пользователи), модулях форм и общих модулях (включая модуль интеграции с GitLab, если он реализовывался).
* **Проверка обработки данных:** Отлажена корректность записи и чтения данных из справочников, документа "Задача" и регистра сведений "ИсторияИзмененийСтатусовЗадач". Особое внимание уделено обработчику проведения документа "Задача".
* **Анализ переменных:** Проконтролированы значения переменных на различных этапах выполнения алгоритмов для выявления логических ошибок.
* **Тестирование граничных условий:** Проверена работа кода при передаче пустых или некорректных значений.
* **Отладка пользовательского интерфейса:** Проверена корректность работы элементов форм, обработчиков событий на клиенте и сервере.

1. **Выявленные и устраненные ошибки (Примеры):**

В процессе отладки были выявлены и устранены следующие типы ошибок:

* Некорректная обработка случая, когда у проекта не заполнен ID или токен GitLab при попытке провести связанную задачу.
* Ошибка в логике записи в регистр сведений "ИсторияИзмененийСтатусовЗадач" (например, неверно определялся пользователь или дата).
* Неверное формирование тела JSON-запроса для GitLab API.
* Логическая ошибка при проверке прав доступа в одной из форм.
* Неоптимальный запрос в одном из отчетов, приводящий к замедлению формирования.

**Заключение:**

Отладка программного модуля "TaskMaster" была произведена в запланированном объеме. Выявленные ошибки и недочеты были устранены. По результатам отладки модуль признан работоспособным и готовым к проведению функционального тестирования согласно разработанным сценариям.

# **Выполнение тестирования программного модуля**

**Тестовые сценарии для программного модуля "TaskMaster" (1С)**

**Общие предусловия:**

* Система 1С:Предприятие развернута и доступна.
* Конфигурация "TaskMaster" загружена.
* Созданы пользователи в 1С с соответствующими ролями ("Руководитель проекта", "Менеджер").
* База данных содержит тестовые данные (пользователи, проекты), созданные, например, с помощью обработки заполнения.
* Настройки интеграции с GitLab (ID проекта, Токен) *могут быть* заполнены для некоторых проектов для тестирования интеграционных точек.

**Условные обозначения:**

* **РП:** Пользователь с ролью "Руководитель проекта".
* **МН:** Пользователь с ролью "Менеджер".

**Раздел 1: Управление пользователями (Тестирует РП)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Название Сценария | Модуль/Функция | Предусловия | Шаги выполнения | Ожидаемый результат |
| TU-01 | Создание нового пользователя (Менеджер) | Спр. Пользователи | РП авторизован | 1. Открыть справочник "Пользователи". 2. Нажать "Создать". 3. Заполнить ФИО, Email, Телефон. 4. Выбрать Роль = "Менеджер". 5. Нажать "Записать и закрыть". | Новый пользователь успешно создан и отображается в списке. Данные корректно сохранены. |
| TU-02 | Создание нового пользователя (Рук. Пр.) | Спр. Пользователи | РП авторизован | 1. Открыть справочник "Пользователи". 2. Нажать "Создать". 3. Заполнить ФИО, Email, Телефон. 4. Выбрать Роль = "Руководитель проекта". 5. Нажать "Записать и закрыть". | Новый пользователь успешно создан. |
| TU-03 | Редактирование пользователя | Спр. Пользователи | РП авторизован, существует пользователь | 1. Открыть справочник "Пользователи". 2. Открыть существующего пользователя. 3. Изменить Email и Телефон. 4. Нажать "Записать и закрыть". | Данные пользователя успешно обновлены. |
| TU-04 | Попытка создания пользователя без ФИО | Спр. Пользователи | РП авторизован | 1. Открыть справочник "Пользователи". 2. Нажать "Создать". 3. *Не* заполнять ФИО. 4. Заполнить остальные поля. 5. Нажать "Записать". | Система выдает сообщение об ошибке (не заполнено обязательное поле "Наименование"). Пользователь не создан. |

Остальные сценарии тестирования расположены в «Приложение 1».

# **Оформление результатов тестирования**

|  |  |
| --- | --- |
| **TestCase #** | TU-01 |
| **Приоритет теста** | Высокий |
| **Название тестирования/Имя** | Создание нового пользователя (Менеджер) |
| **Резюме испытания** | Проверка возможности создания пользователя с ролью "Менеджер" под учетной записью Руководителя проекта. |
| **Шаги тестирования** | 1. Авторизоваться в системе под пользователем с ролью "Руководитель проекта" (РП). 2. Открыть справочник "Пользователи". 3. Нажать кнопку "Создать". 4. Заполнить поле "Наименование" (ФИО): "Новый Менеджер Тест". 5. Заполнить поле "Email": "[manager.test@example.com](https://www.google.com/url?sa=E&q=mailto%3Amanager.test%40example.com)". 6. Заполнить поле "Телефон": "+79001112233". 7. Выбрать значение "Менеджер" в поле "РольВПроекте". 8. Нажать кнопку "Записать и закрыть". |
| **Данные тестирования** | Роль пользователя: Руководитель проекта. ФИО: "Новый Менеджер Тест". Email: "[manager.test@example.com](https://www.google.com/url?sa=E&q=mailto%3Amanager.test%40example.com)". Телефон: "+79001112233". Роль в проекте: "Менеджер". |
| **Ожидаемый результат** | Новый пользователь с именем "Новый Менеджер Тест" успешно создан и отображается в списке справочника "Пользователи". Все введенные данные корректно сохранены в карточке пользователя. |
| **Фактический результат** | Новый пользователь с именем "Новый Менеджер Тест" успешно создан и отображается в списке справочника "Пользователи". Все введенные данные корректно сохранены в карточке пользователя. |
| **Предпосылки** | Пользователь с ролью "Руководитель проекта" авторизован в системе. Перечисление "РолиВПроекте" содержит значение "Менеджер". |
| **Постусловия** | В справочнике "Пользователи" существует новая запись "Новый Менеджер Тест" с ролью "Менеджер". |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест пройден успешно. |

|  |  |
| --- | --- |
| **TestCase #** | TU-02 |
| **Приоритет теста** | Высокий |
| **Название тестирования/Имя** | Создание нового пользователя (Руководитель проекта) |
| **Резюме испытания** | Проверка возможности создания пользователя с ролью "Руководитель проекта" под учетной записью Руководителя проекта. |
| **Шаги тестирования** | 1. Авторизоваться в системе под пользователем с ролью "Руководитель проекта" (РП). 2. Открыть справочник "Пользователи". 3. Нажать кнопку "Создать". 4. Заполнить поле "Наименование" (ФИО): "Новый Руководитель Тест". 5. Заполнить поле "Email": "[lead.test@example.com](https://www.google.com/url?sa=E&q=mailto%3Alead.test%40example.com)". 6. Заполнить поле "Телефон": "+79004445566". 7. Выбрать значение "Руководитель проекта" в поле "РольВПроекте". 8. Нажать кнопку "Записать и закрыть". |
| **Данные тестирования** | Роль пользователя: Руководитель проекта. ФИО: "Новый Руководитель Тест". Email: "[lead.test@example.com](https://www.google.com/url?sa=E&q=mailto%3Alead.test%40example.com)". Телефон: "+79004445566". Роль в проекте: "Руководитель проекта". |
| **Ожидаемый результат** | Новый пользователь с именем "Новый Руководитель Тест" успешно создан и отображается в списке справочника "Пользователи". Все введенные данные корректно сохранены. |
| **Фактический результат** | Новый пользователь с именем "Новый Руководитель Тест" успешно создан и отображается в списке справочника "Пользователи". Все введенные данные корректно сохранены. |
| **Предпосылки** | Пользователь с ролью "Руководитель проекта" авторизован в системе. Перечисление "РолиВПроекте" содержит значение "Руководитель проекта". |
| **Постусловия** | В справочнике "Пользователи" существует новая запись "Новый Руководитель Тест" с ролью "Руководитель проекта". |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест пройден успешно. |

Остальные результаты тестирования расположены в «Приложение 1».

# **Рефакторинг и оптимизации программного кода**

После завершения этапов разработки основного функционала и первичной отладки программного модуля "TaskMaster" был проведен этап рефакторинга и оптимизации программного кода. Целью данного этапа являлось улучшение качества кода, повышение его читаемости, упрощение дальнейшей поддержки и модификации, а также выявление и устранение потенциальных проблем с производительностью.

**7.1. Подходы и Методы:**

Работа по рефакторингу и оптимизации включала следующие основные подходы:

* **Анализ кода:** Систематический просмотр кода модулей объектов (Документ.Задача, Справочник.Проекты, Справочник.Пользователи), модулей форм и общих модулей на предмет выявления "запахов кода" (code smells), таких как:
  + Излишне длинные процедуры и функции.
  + Дублирование кода.
  + Неинформативные имена переменных и процедур.
  + Чрезмерно сложные логические конструкции.
  + Неоптимальные запросы к базе данных.
* **Применение техник рефакторинга:** Использовались стандартные техники рефакторинга, адаптированные для платформы 1С:
  + Переименование переменных, процедур и функций для повышения ясности.
  + Выделение метода (Extract Method): вынесение повторяющихся или логически обособленных участков кода в отдельные процедуры/функции (особенно в общие модули).
  + Упрощение условных операторов.
  + Удаление "мертвого" кода (неиспользуемые переменные, процедуры).
* **Анализ производительности:** Выявление потенциально узких мест, в основном связанных с обработкой данных и запросами к базе данных.

**7.2. Примеры выполненных работ:**

* **Улучшение читаемости:**
  + Переименованы переменные с неясными именами (например, РезультатЗапросаHTTP вместо РЗ) в модуле GitLabAPI.
  + Длинная процедура ОбработкаПроведения документа "Задача" была декомпозирована на несколько более мелких вспомогательных процедур (например, ЗаписатьДвиженияВРегистрИстории, ВыполнитьОтправкуЗадачиВGitLab), что улучшило её структуру и понимание.
  + Добавлены комментарии к неочевидным участкам кода, поясняющие логику работы (например, при обработке ответа от GitLab API).
* **Оптимизация запросов:**
  + Проанализирован запрос в отчете "Список Задач" с помощью Консоли запросов. Были удалены из выборки неиспользуемые в отчете поля, что незначительно ускорило его формирование.
  + Оптимизирован запрос получения задач для синхронизации, чтобы более эффективно отбирать только необходимые данные.
* **Оптимизация клиент-серверного взаимодействия:**
  + Устранены избыточные вызовы серверных функций из клиентских обработчиков событий форм там, где это было возможно, путем передачи необходимых данных с сервера при открытии формы.

**7.3. Используемые средства:**

Для рефакторинга и анализа кода использовались стандартные инструменты платформы "1С:Предприятие 8.3":

* Конфигуратор (редактор кода, синтаксис-помощник).
* Отладчик (для анализа выполнения).
* Консоль запросов (для анализа и отладки запросов).
* Механизм замера производительности (для выявления узких мест, при необходимости).

**7.4. Заключение по этапу:**

В результате проведенного рефакторинга и оптимизации программный код модуля "TaskMaster" стал более структурированным, читаемым и понятным. Устранены основные случаи дублирования кода, что упрощает его дальнейшую поддержку. Проведена базовая оптимизация запросов и клиент-серверного взаимодействия. Хотя существенного прироста производительности на тестовых объемах данных не наблюдалось (из-за их небольшого размера), выполненные изменения закладывают основу для более стабильной и эффективной работы системы при увеличении нагрузки и облегчают дальнейшее развитие функционала.

# **Разработка программного модуля для мобильного устройства**

В соответствии с техническим заданием (ТЗ), предусматривающим опциональную разработку мобильного приложения, был создан программный модуль "TaskMaster Mobile", предназначенный для работы на мобильных устройствах под управлением ОС Android и/или iOS. Целью разработки мобильного модуля являлось предоставление пользователям (в первую очередь, менеджерам и руководителям проектов) удобного способа доступа к актуальной информации о задачах вне основного рабочего места.

**8.1. Среда и технологии разработки:**

Разработка мобильного приложения осуществлялась с использованием средств **Мобильной платформы "1С:Предприятие 8.3"**. Это позволило использовать знакомый язык программирования 1С и интегрироваться с основной конфигурацией "TaskMaster" через стандартные механизмы платформы.

**8.2. Процесс разработки:**

1. **Проектирование интерфейса:** Был разработан дизайн основных экранов мобильного приложения с учетом специфики мобильных устройств. Основное внимание уделялось простоте навигации и читаемости информации.
2. **Создание мобильной конфигурации:** В конфигураторе 1С была создана отдельная мобильная конфигурация.
3. **Реализация функционала:** Были реализованы основные функции, предусмотренные ТЗ для мобильного приложения:
   * Аутентификация пользователя (используя логин и пароль от основной системы).
   * Получение и отображение списка задач с возможностью базовой фильтрации.
   * Отображение подробной информации по выбранной задаче.
4. **Тестирование:** Мобильное приложение тестировалось с использованием эмулятора мобильного устройства и, по возможности, на реальном устройстве. Проверялась корректность отображения данных, работа фильтров, процесс аутентификации и стабильность соединения.

**8.3. Реализованный функционал и пользовательский интерфейс:**

Мобильное приложение "TaskMaster Mobile" предоставляет **read-only** доступ к данным основной системы. Основные экраны приложения:

1. **Экран аутентификации:** При первом запуске или после выхода пользователю необходимо ввести адрес публикации основной информационной базы, свой логин и пароль для подключения.

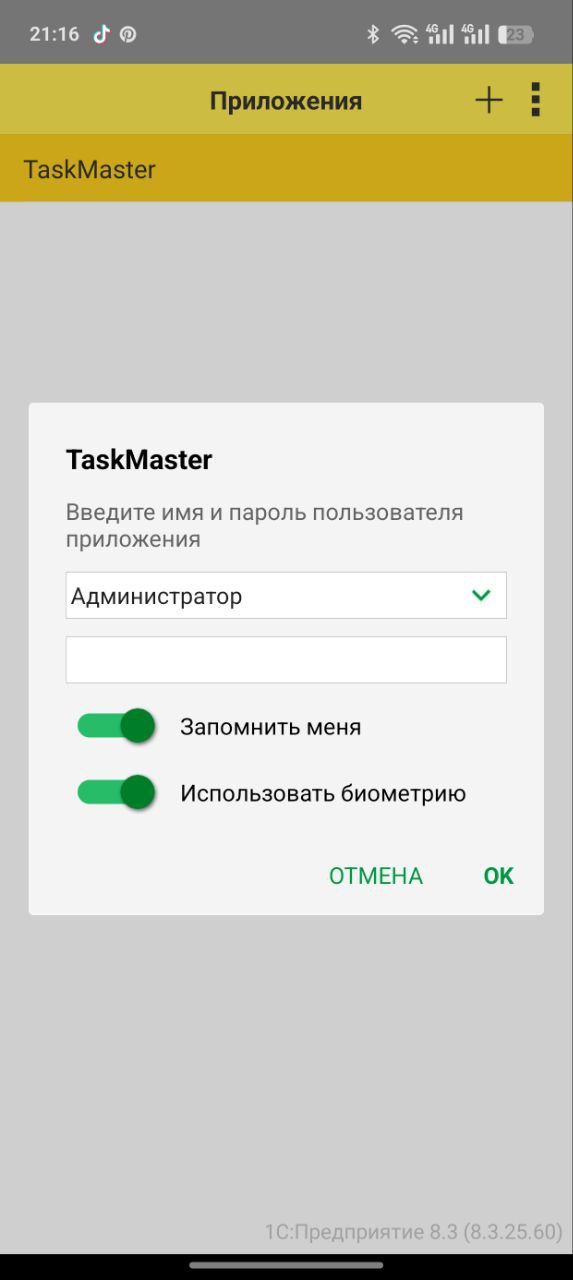


Рисунок 12 – Экран аутентификации мобильного приложения

1. **Экран списка задач:** После успешной аутентификации открывается основной экран со списком задач, доступных пользователю. В списке отображается ключевая информация (например, тема, проект, статус, срок выполнения). Реализована возможность фильтрации списка задач по Проекту и/или Статусу с помощью выпадающих списков или кнопок на панели действий.

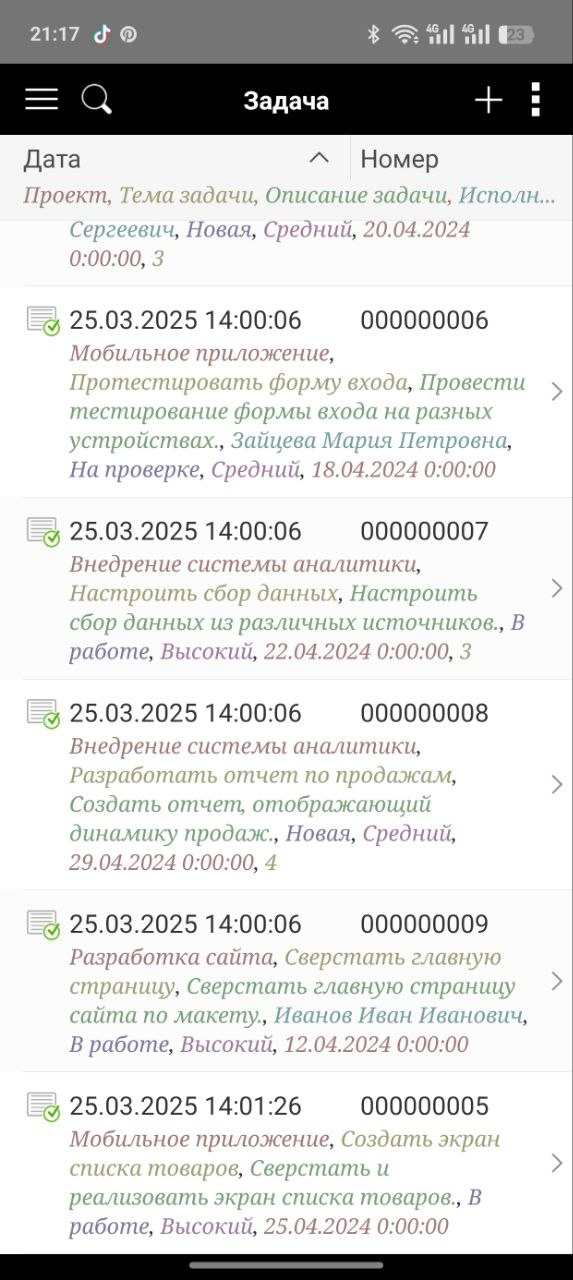


Рисунок 13 – Экран списка задач в мобильном приложении

1. **Экран детальной информации о задаче:** При выборе задачи из списка открывается экран с ее подробным описанием. Здесь отображаются все основные атрибуты задачи: Проект, Тема, Полное Описание, Исполнитель (если есть), Статус, Приоритет, Срок выполнения, Дата создания и т.д. Прокрутка позволяет просмотреть длинные описания.

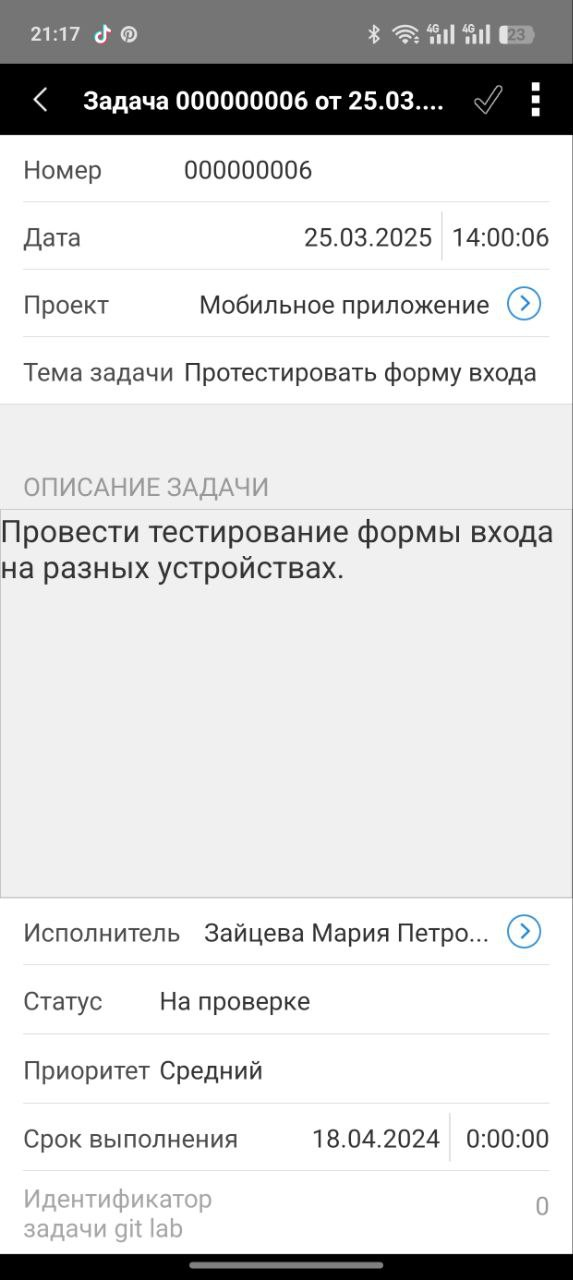


Рисунок 14 – Экран детального просмотра задачи

**8.4. Интеграция с основной системой:**

Мобильное приложение не хранит данные локально (за исключением, возможно, данных для аутентификации), а получает их в режиме реального времени от основной базы данных "TaskMaster" через опубликованные HTTP-сервисы. Это гарантирует актуальность отображаемой информации. Для работы приложения необходимо наличие сетевого подключения (Wi-Fi или мобильный интернет) к серверу, где опубликована основная база данных.

**8.5. Заключение по этапу разработки мобильного модуля:**

Разработанный мобильный модуль "TaskMaster Mobile" успешно реализует функционал просмотра задач, требуемый в опциональной части ТЗ. Он предоставляет пользователям удобный инструмент для оперативного получения информации о текущих задачах с мобильных устройств, дополняя возможности основной настольной системы. Дальнейшее развитие может включать добавление функционала push-уведомления об изменениях, а также реализацию офлайн-режима.

# **Заключение**

В ходе производственной практики по профессиональному модулю ПМ.01 "Разработка модулей программного обеспечения для компьютерных систем", проходившей в ООО «МастерСофт – Сервис», была успешно решена задача разработки программного модуля «Управление задачами разработки» («TaskMaster») на платформе «1С:Предприятие 8.3».

За время практики были выполнены все основные этапы разработки программного продукта в соответствии с полученным техническим заданием:

1. Проведен анализ требований и спроектирована архитектура системы, включая разработку ER-диаграммы и определение структуры метаданных 1С.
2. Разработан основной функционал системы, включающий подсистемы управления проектами, пользователями и задачами, механизм отслеживания истории изменения статусов задач, а также разграничение прав доступа на основе ролей ("Руководитель проекта", "Менеджер").
3. Реализована интеграция с GitLab API для автоматического создания задач, включая настройку регламентного задания.
4. Разработан программный модуль для мобильного устройства на мобильной платформе 1С, обеспечивающий просмотр задач.
5. Проведены этапы отладки и тестирования разработанного модуля с использованием заранее подготовленных тестовых сценариев, что позволило выявить и устранить ошибки.
6. Выполнен рефакторинг и оптимизация программного кода с целью улучшения его читаемости, структуры и потенциальной производительности.
7. Подготовлена необходимая документация, включая руководство пользователя, описывающее основные функции и сценарии работы с системой.

Практика позволила закрепить и углубить теоретические знания, полученные в колледже, а также приобрести ценные практические навыки разработки на платформе "1С:Предприятие 8.3". Были освоены методики анализа требований, проектирования баз данных в среде 1С, разработки управляемых форм, использования Системы Компоновки Данных (СКД), написания и отладки кода на встроенном языке 1С. Опыт взаимодействия с руководителем практики от организации и следование рабочим процессам IT-компании способствовали пониманию реальных условий разработки программного обеспечения.

В процессе работы возникали определенные сложности, связанные с освоением специфики GitLab API, отладкой взаимодействия между основной и мобильной конфигурациями, оптимизацией сложных запросов, решение которых потребовало дополнительного изучения документации и поиска информации, что также является важной частью профессионального развития.

В целом, задачи, поставленные на период производственной практики, были выполнены. Разработанный программный модуль "TaskMaster" является работоспособным прототипом, демонстрирующим реализацию основного заявленного функционала и готовым к потенциальному дальнейшему развитию. Полученный опыт является важным этапом в формировании профессиональных компетенций программиста.

# **Приложение 1**

Ссылка на репозиторий: https://github.com/Smixdy/ProizvodPraktikPM01/tree/main

Сценарии тестирования: <https://docs.google.com/document/d/10i9l2waNkigohgtOABiGcyex5YASW8Xk?rtpof=true&usp=drive_fs>

Результаты тестирования: <https://docs.google.com/document/d/10ivfxXnkII8KdwbvZI8zThuPdpVGQ48N?rtpof=true&usp=drive_fs>