Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное
учреждение высшего образования
Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Кафедра Математического обеспечения и суперкомпьютерных технологий

УЧЕБНЫЙ КУРС

«Проектирование и архитектура программных систем»

для подготовки по направлению «Программная инженерия»

ФУНКЦИОНАЛЬНАЯ СПЕЦИФИКАЦИЯ

Нижний Новгород
2025

Содержание

Содержание 2
1 История проекта 3
2 Цели дизайна 4
2.1 Требования пользователя 4
2.2 Системные требования 5
2.3 Сценарии использования 6
3 Исключенные возможности и неподдерживаемые сценарии 10
4 Предположения и зависимости 11
5 Проект решения 12
5.1 Концептуальный проект 12
5.2 Логический проект 13
5.3 Физический проект 14
6 Требования к инсталляции и деинсталляции 16

1 История проекта

Проект инициирован как учебный в рамках курса «Проектирование и архитектура программных систем». Ключевые этапы и решения:

- Сентябрь 2025: Проведен анализ предметной области, сформулированы необходимость и видение проекта. Принято решение о разработке веб-приложения для управления проектами и задачами в распределенных командах.
- Сентябрь 2025: Определена архитектура «клиент-сервер» с использованием стека технологий React (Frontend) и Python/FastAPI (Backend). В качестве СУБД выбрана PostgreSQL для упрощения развертывания.
- Октябрь 2025: Утверждены рамки проекта (Scope). Принято решение исключить из первого релиза встроенный мессенджер и личные справочники исполнителей для соблюдения сроков проекта (до декабря 2025).
- Октябрь 2025: Разработана и утверждена настоящая функциональная спецификация. Разработка документации и дизайна
- Ноябрь 2025: Завершена разработка документации. Начало разработки frontend и backend.
- Конец ноября 2025: Завершена разработка frontend и backend. Начало тестирования и написания документации по спецификации тестов
- Декабрь 2025: Завершение тестирования и написания документации. Создание презентации. Защита проекта.

2 Цели дизайна

Цель дизайна — создать интуитивно понятное, надежное и функциональное веб-приложение для управления задачами и проектами, которое удовлетворяет требованиям пользователей (Администраторов, Менеджеров и Исполнителей) и может быть легко развернуто на сервере заказчика.

2.1 Требования пользователя

1. Для Администратора:
    - регистрировать новых сотрудников с указанием: ФИО, роли, должности, рабочего графика, даты рождения, контактного телефона, электронной почты,
    - добавлять в систему новые офисы, отделы, графики работы, статусы задач,
    - выполнять резервное копирование базы данных,
    - управлять техническими настройками системы.
2. Для Менеджера:
    - создавать, редактировать и удалять проекты и релизы,
    - создавать задачи в рамках проектов, назначать исполнителей, устанавливать сроки выполнения,
    - просматривать список всех задач в проекте с возможностью сортировки по статусу и дате,
    - создавать события («Встречи») в календаре и назначать на них исполнителей,
    - получать системные уведомления при изменении статуса задачи исполнителем (например, при завершении),
    - просматривать статус присутствия коллег (на работе/не на работе/в отпуске).
3. Для Исполнителя:
    - просматривать назначенные ему задачи и проекты,
    - изменять статус назначенной задачи по установленному workflow: «К выполнению» → «В работе» → «На проверке» → «Отложен» → «Отменен» → «Завершена»,
    - оставлять текстовые комментарии к задачам,

- просматривать календарь с назначенными ему встречами и отпуском.

2.2 Системные требования

1. Требования к серверу:
    - процессор (CPU) 4-8 ядер (Intel Xeon E-серии или AMD EPYC),
    - оперативная память (RAM) 16-32 ГБ DDR4 ECC,
    - накопители (Storage) 2 x 512 ГБ NVMe SSD в RAID 1,
    - сетевая карта 1 Гбит/с с гарантированной полосой,
    - блок питания Двойной, с резервированием (A+B),
    - платформа Выделенный сервер (Dedicated) или мощный Виртуальный сервер (VDS).
2. Системные требования для устройств Менеджеров и Исполнителей:
    - MS Edge версии не ниже 124,
    - доступ в интернет,
    - корпоративный VPN.

2.3 Сценарии использования

Ниже приведена диаграмма, описывающая основные сценарии использования (рисунок 1).

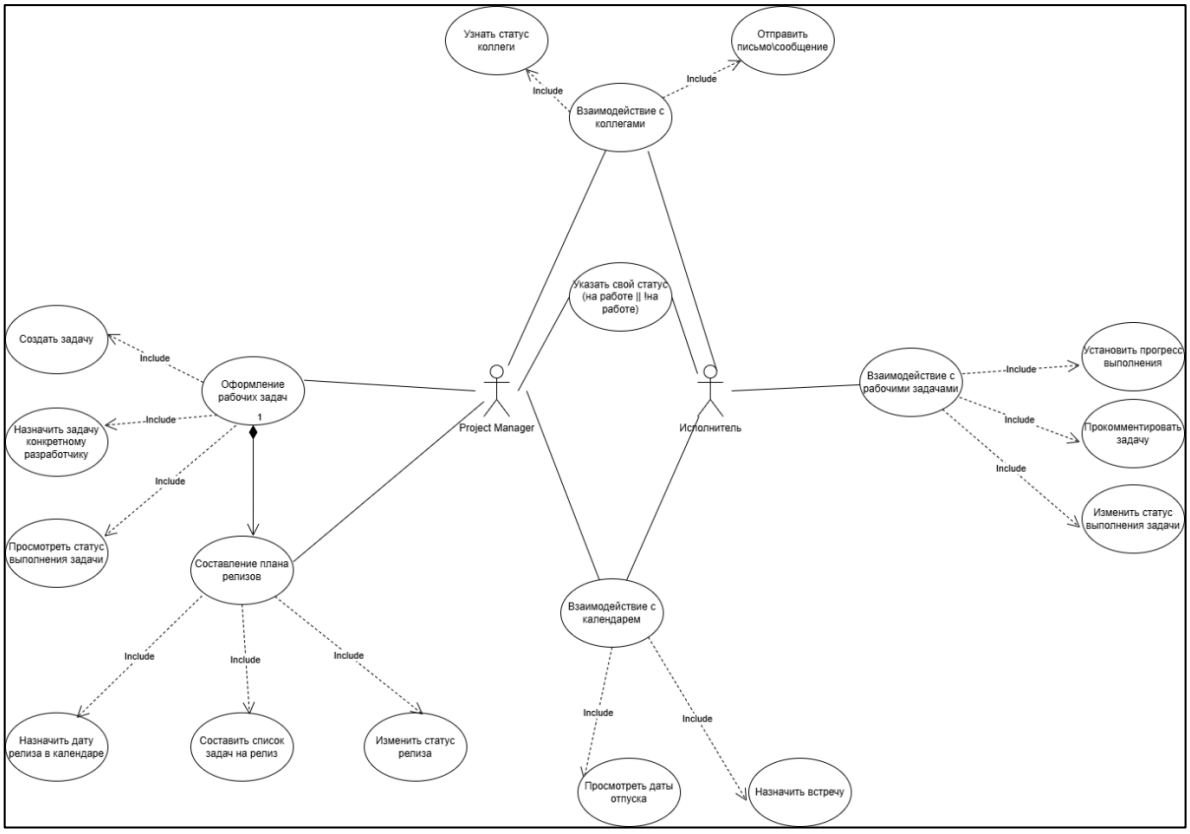

Рисунок 1 – Use-case диаграмма

Основные варианты использования:

1. Управление учетными записями

   Суть: регистрация, вход в систему и разграничение прав доступа по ролям.

   - Регистрация новых пользователей с выбором роли (Администратор/Менеджер/Исполнитель).
   - Аутентификация по логину и паролю.
   - Автоматическое перенаправление на соответствующий интерфейс в зависимости от роли.

2. Работа с календарем

   Суть: планирование и отслеживание мероприятий и отпусков.

   - Создание встреч с указанием участников, времени и описания.

- Просмотр расписания мероприятий команды.
- Отслеживание графика отпусков сотрудников.

3. Мониторинг статусов сотрудников

   Суть: отслеживание доступности членов команды.

   - Визуальное отображение статусов присутствия.
   - Быстрая оценка кто доступен для работы и коммуникации.

4. Управление проектами (для Менеджера)

   Суть: полный цикл создания и ведения проектов.

   - Создание новых проектов с описанием и настройками.
   - Редактирование параметров существующих проектов.
   - Обзор всех задач проекта с отслеживанием прогресса.

5. Управление задачами (для Менеджера)

   Суть: создание, назначение и контроль выполнения задач.

   - Создание задач с назначением исполнителей и сроков.
   - Публикация и редактирование задач.
   - Удаление задач с подтверждением.

6. Планирование релизов (для Менеджера)

   Суть: организация и контроль выпусков продукта.

   - Создание событий "Релиз" в календаре.
   - Формирование списка задач для конкретного релиза.
   - Отслеживание статуса релиза (К выполнению/В работе/На ревью/Отложен/Отменен/Завершен).

7. Получение уведомлений (для Менеджера)

   Суть: автоматическое информирование о ключевых событиях.

   - Всплывающие уведомления о завершении задач исполнителями.
   - Накопительный "Центр уведомлений" для просмотра истории.

8. Работа с задачами (для Исполнителя)

Суть: выполнение и отслеживание назначенных задач.

- Изменение статуса задачи по workflow: "К выполнению" → "В работе" → "На ревью" → "Завершена".
- Каждый статус отражает определенный этап работы над задачей.

9. Комментирование задач (для Исполнителя)

Суть: обсуждение деталей и прогресса по задачам.

- Добавление текстовых комментариев к задачам.
- Просмотр истории обсуждений с указанием авторов и дат.

Activity диаграмма (рисунок 2):

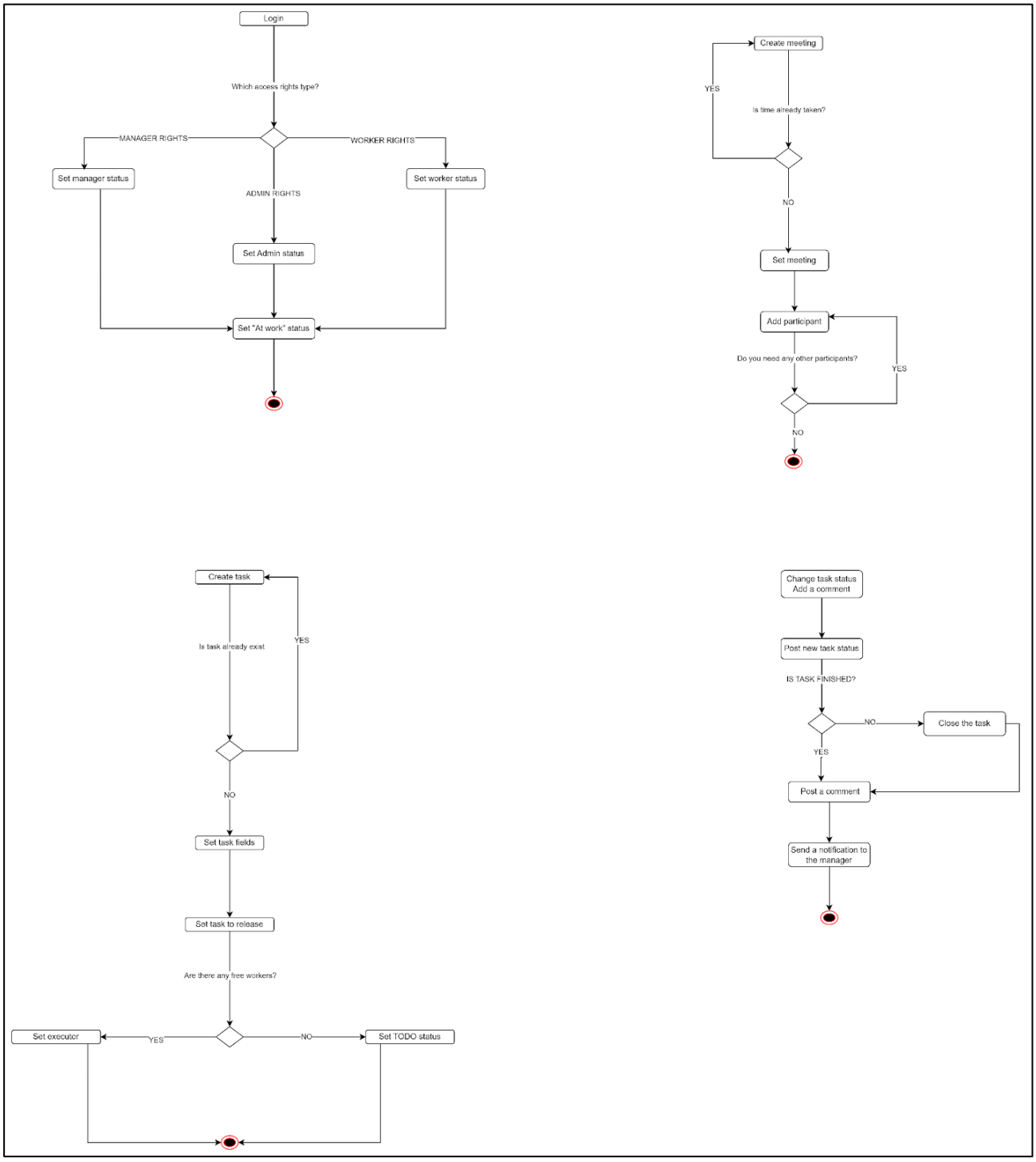

Рисунок 2 – Activity диаграмма

3 Исключенные возможности и неподдерживаемые сценарии

- Встроенный мессенджер для коммуникации. Причина: повышает сложность и объем работ, выходит за рамки основного фокуса проекта (управление задачами). Компромисс в пользу соблюдения сроков. Может быть реализован в будущих версиях.
- Личные справочники/заметки исполнителей. Причина: не является критичной функцией для основного workflow управления проектами. Компромисс в пользу соблюдения сроков и ограничения объема функциональности.
- Установка персональных дедлайнов для задач в календаре (Исполнитель). Причина: не является обязательной для основного workflow, повышает сложность синхронизации. Контроль сроков остается за менеджером.
- Просмотр единого списка всех задач across всех проектов (Менеджер). Причина: базовая навигация по проектам признана достаточной. Реализация глобального отчета требует дополнительной сложности.
- Активация web-приложения через покупку лицензии (ключа активации). Причина: запуск и тестирование проекта на ранних этапах важнее внедрения сложной системы монетизации. Компромисс в пользу ускорения выхода MVP (минимально жизнеспособного продукта) и упрощения первоначального процесса регистрации пользователей.

4 Предположения и зависимости

1. Предположения:
   - аппаратные возможности сервера заказчика достаточны для работы приложения в рамках ожидаемой нагрузки (до 50 concurrent пользователей),
   - пользователи обладают базовыми навыками работы с веб-браузерами,
   - заказчик обеспечит стабильное интернет-соединение и доступ к корпоративному VPN.
2. Зависимости:
   - проект зависит от стабильности и поддержки выбранных технологий: React, FastAPI, PostgreSQL, Docker,
   - сроки реализации проекта жестко привязаны к учебному плану (завершение до конца декабря 2025 года).

5 Проект решения

5.1 Концептуальный проект

Основная идея решения:

Создается простое и понятное веб-приложение, которое помогает командам работать вместе над проектами и задачами. приложение объединяет в одном месте управление задачами, календарь встреч и отслеживание прогресса работы.

1. Архитектура решения:
   - приложение строится по клиент-серверной модели,
   - клиентская часть - веб-интерфейс для Администратора, Менеджеров и Исполнителей,
   - серверная часть обеспечивает логику, хранение данных и API.
2. Ключевые проектные решения:
   - разделение интерфейсов по ролям: Администратор, Менеджер и Исполнитель,
   - использование единого календаря для встреч и отпусков,
   - статусная модель задач: «К выполнению» → «В работе» → «На проверке» → «Отложен» → «Отменен» → «Завершена»,
   - уведомления для Менеджеров при изменении статуса задач.

Как это будет выглядеть на практике (пример сценария работы) (рисунок 3):

1. Администратор регистрирует пользователей в системе с указанием роли.
2. Менеджер создает проект "Разработка нового функционала".
3. Добавляет задачи: "Создать дизайн", "Написать код", "Протестировать".
4. Назначает исполнителей и сроки выполнения.
5. Исполнитель видит свои задачи, начинает работу, меняет статусы.
6. Менеджер получает уведомления о прогрессе.
7. Оба видят в календаре запланированные встречи по проекту.

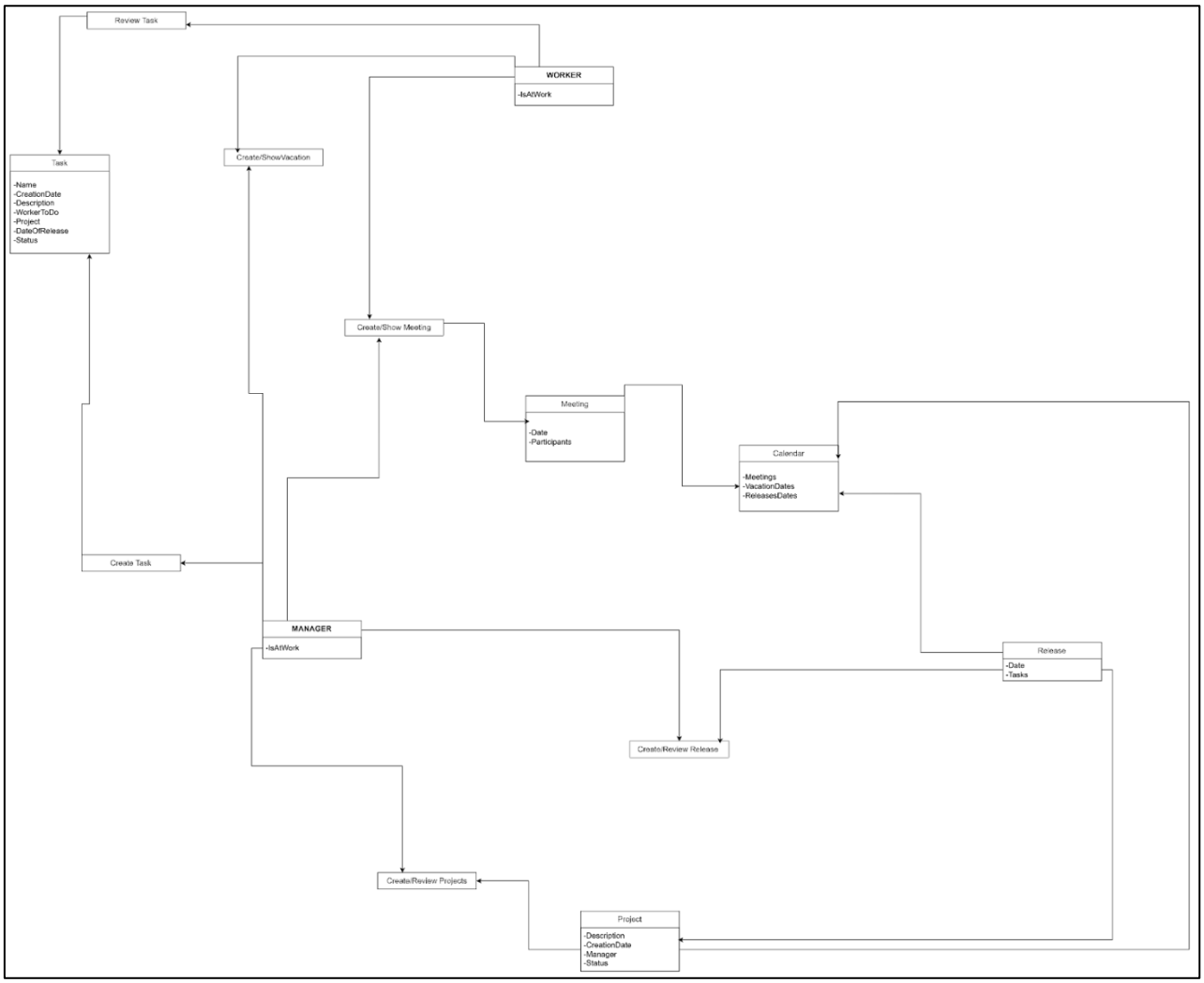

Рисунок 3 – Концептуальная схема

5.2 Логический проект

Ниже представлена диаграмма классов (рисунок 4).

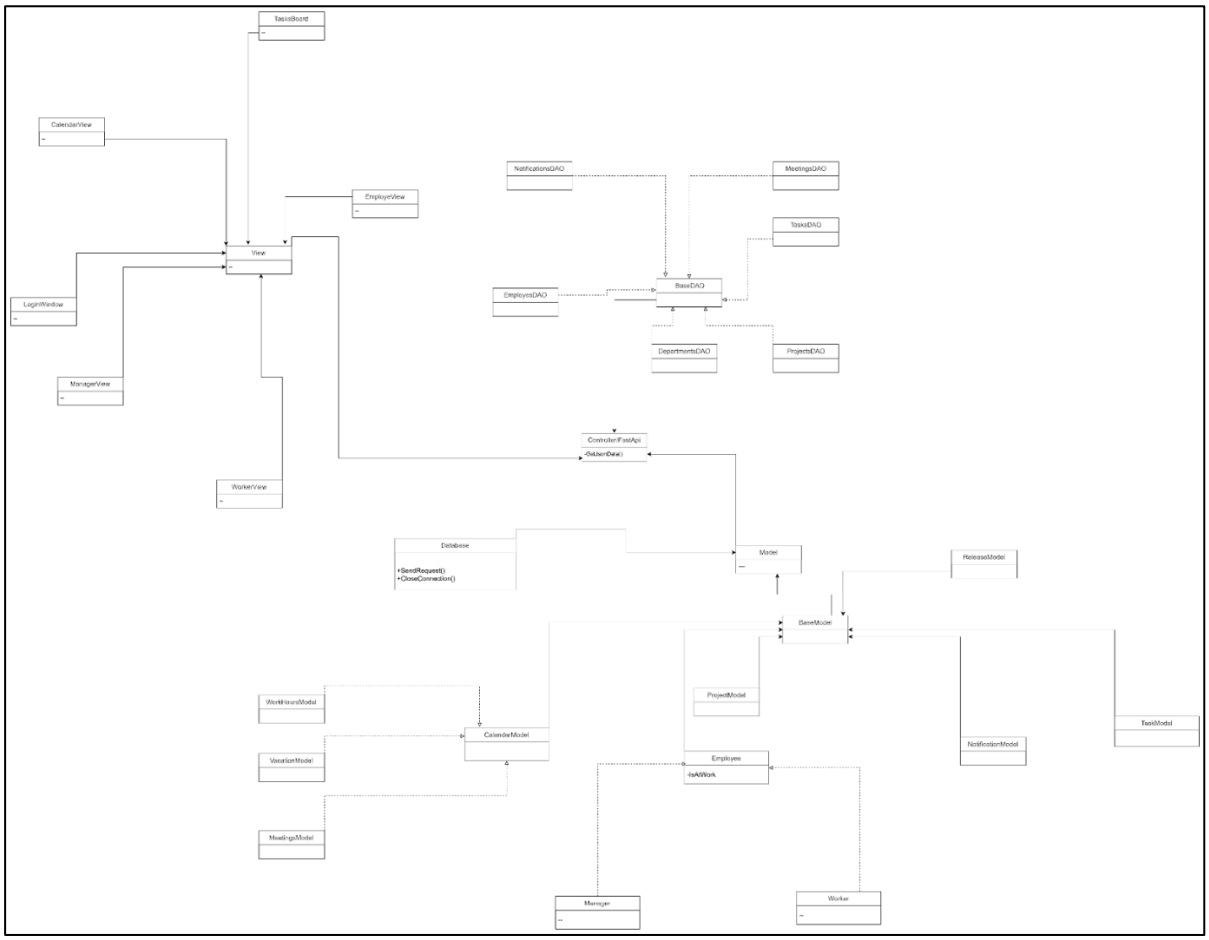

Рисунок 4 – Диаграмма классов

Подробнее можно рассмотреть **здесь**.

5.3 Физический проект

Технологический стек:

- frontend: React, Fetch API , React Router (маршрутизация),
- backend: Python, FastAPI (REST API), JWT (аутентификация), SQLAlchemy (ORM), Alembic (миграции),
- база данных: PostgreSQL,
- контейнеризация: Docker, Docker Compose.

Компоненты системы:

1. Клиентское приложение (SPA):
   - модули: аутентификация, проекты, задачи, календарь, уведомления,

- адаптивный интерфейс для работы в MS Edge версии 124+.

2. Серверное приложение (REST API)
   - модули: пользователи, проекты, задачи, календарь, комментарии,
   - автодокументация через Swagger/OpenAPI.
3. База данных
   - таблицы: departments, employeedepartments, employeemeetings, employees, files, meetings, notifications, projects, releases, roles, schedules, statuses, taskcomments, taskfiles, tasks, vacations, workhours,
   - схема соответствует логической модели.
4. Требования к развертыванию:
   - сервер: 4-8 ядер CPU, 16-32 ГБ RAM, 2×512 ГБ NVMe SSD (RAID 1),
   - ОС: Linux (рекомендуется Ubuntu 20.04+),
   - доступен через корпоративный VPN.

6 Требования к инсталляции и деинсталляции

Инсталляция:

1. Убедиться, что сервер соответствует системным требованиям.
2. Установить Docker и Docker Compose.
3. Склонировать репозиторий с исходным кодом.
4. Настроить переменные окружения (параметры БД).
5. Запустить команду: «docker-compose up -d».
6. Приложение будет доступно по указанному IP-адресу.

Деинсталляция:

1. Остановить контейнеры: «docker-compose down».
2. Удалить образы и тома (опционально): «docker system prune -a –volumes».
3. Удалить файлы проекта с сервера.

Требования к процессам:

1. Резервное копирование БД перед обновлением или удалением.
2. Наличие лицензионного ключа для активации.
3. Доступ к корпоративному VPN для установки и настройки.