

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»

КОНЦЕПЦИЯ ПРОЕКТА

Нижний Новгород

2025

Содержание

Содержание	2
1 Необходимость проекта	3
1.1 Обоснование необходимости.....	3
1.2 Видение проекта.....	3
1.3 Анализ выгод.....	3
2 Концепция решения.....	4
2.1 Цели и Задачи.....	4
2.2 Предположения и Ограничения	5
2.3 Анализ использования.....	5
2.4 Требования.....	6
2.5 Системные требования.....	7
3 Рамки	8
3.1 Функциональность решения	8
3.2 За рамками решения	12
3.3 Критерии одобрения решения	13
4 Стратегии дизайна решения	14
4.1 Стратегия архитектурного дизайна.....	14
4.2 Стратегия технологического дизайна.....	14

1 Необходимость проекта

1.1 Обоснование необходимости

Приложение поможет командам, работающим удаленно, в организации рабочего процесса. Оно упростит процесс назначения задач и контроля их выполнения. Так же приложение позволит членам команды, менеджерам и исполнителям, распределять свои задачи во времени. Помимо этого, есть необходимость в централизации общения менеджеров и исполнителей по вопросам рабочих задач.

1.2 Видение проекта

Web-приложение для менеджеров и членов команд, которое позволит оптимизировать процесс распределения и планирования выполнения задач между сотрудниками. Срок разработки до 31.12.2025.

1.3 Анализ выгод

1. Централизация. Все общение членов команды по поводу рабочих задач и проектов будет в одном приложении, что поможет уменьшить количество используемых сервисов для взаимодействия команд. Как следствие, отсутствие избыточности в функциональности.
2. Отечественная разработка. Приложение, разработанное на территории России, не попадает под санкции, а значит, не будет поводов к запрету использования.
3. Прозрачность выполнения задач. Менеджер может отслеживать, какой статус у каждой задачи и кто является исполнителем. Исполнитель же видит, что и к какому сроку он должен сделать.
4. Опыт. Во время работы над проектом разработчики получают дополнительный опыт в программировании и совершенствуют свои навыки. Так же они смогут использовать готовое приложение в своей работе, находить недостатки и сразу их устранять.

2 Концепция решения

2.1 Цели и Задачи

Разработка web-приложения с тремя графическими интерфейсами: Администратора, Менеджера и Исполнителя.

1. Интерфейс Администратора позволит:
 - регистрировать новых пользователей с указанием роли,
 - добавлять в систему новые офисы, отделы.
2. Интерфейс Менеджера позволит:
 - создавать релизы,
 - создавать проекты в релизах,
 - создавать задачи в проектах,
 - назначать на задачи Исполнителей и сроки выполнения,
 - просматривать список задач в порядке создания задач или по статусу выполнения,
 - редактировать задачи и удалять их,
 - создавать «Встречи» в календаре, добавлять Исполнителей и время,
 - оставлять комментарии на странице задачи.
3. Интерфейс Исполнителя позволит:
 - просматривать список всех задач или предназначенных только для этого Исполнителя, фильтровать список по исполнителям, датам,
 - изменять статус задачи,
 - оставлять комментарии на странице задачи.
4. Должны быть реализованы уведомления для Менеджеров об изменении статуса задач Исполнителями.

2.2 Предположения и Ограничения

Аппаратные возможности сервера, на который будет устанавливаться приложение.

2.3 Анализ использования

2.3.1 Пользователи

- **Исполнитель:** сотрудник команды, исполнитель задач;
- **Менеджер:** руководитель команды, выдающий задачи и устанавливающий сроки выполнения задач;
- **Администратор:** сотрудник, отвечающий за техническое обслуживание системы, управление базой данных и регистрацию пользователей.

2.3.2 Сценарии использования

Ниже представлена use-case диаграмма (рисунок 1).

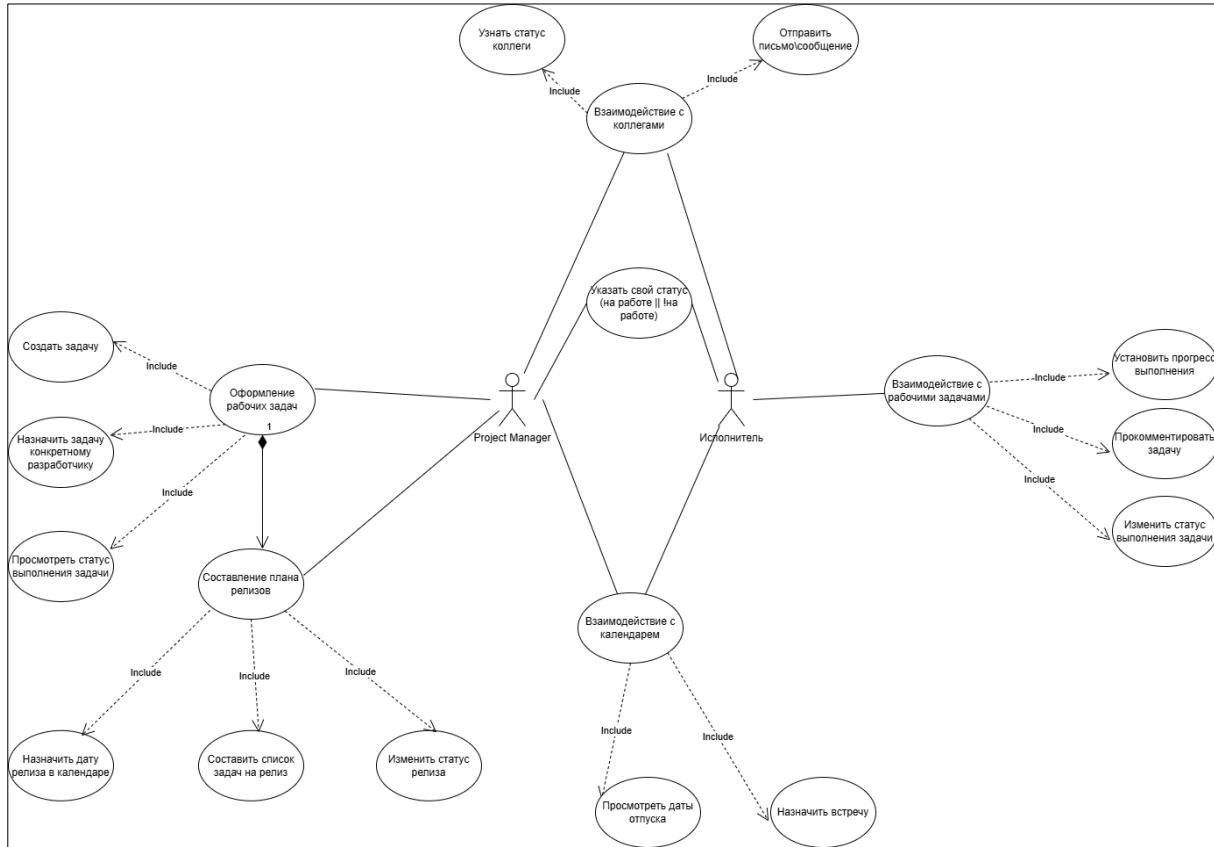


Рисунок 1 – Use-case диаграмма

2.4 Требования

2.4.1 Требования пользователей

Администратор может:

- регистрировать новых сотрудников с указанием: ФИО, роли, должности, рабочего графика, даты рождения, контактного телефона, электронной почты,
- добавлять в систему новые офисы, отделы и редактировать их,
- выполнять резервное копирование базы данных,
- управлять техническими настройками системы.

Менеджер может:

- создавать проект, в котором будут создаваться задачи,
- создавать задачи: указать название, Исполнителя, краткое описание и срок выполнения задачи,

- создавать задачу-созвон: она сразу добавляется в календарь Исполнителей,
- просматривать все задачи по проекту на одной странице, сортировать по статусу выполнения и дате создания,
- создавать уведомление об изменении задач,
- получать уведомления о задачах, которые Исполнители перевели в статус «Завершена».

Исполнитель может:

- оставлять комментарий под задачей,
- изменять статус задачи: «К выполнению» → «В работе» → «На проверке» → «Отложен» → «Отменен» → «Завершена».

2.5 Системные требования

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).

Системные требования для устройств Администраторов, Менеджеров и Исполнителей:

- MS Edge версии не ниже 124,
- доступ в Интернет,
- корпоративный VPN.

3 Рамки

3.1 Функциональность решения

1. Общая функциональность

1.1 Управление пользователями и аутентификация:

1.1.1 вход (получение доступа к системе):

1.1.1.1 ввод логина и пароля пользователем,

1.1.1.2 проверка соответствия в базе данных системой,

1.1.1.3 при успешной проверке создается сессия и пользователь перенаправляется на его персональный дашборд в соответствии с его ролью,

1.1.1.4 при неудаче отображается сообщение о неверном вводе логина или пароля;

1.1.2 разграничение прав и интерфейсов по ролям:

Администратор/Менеджер/Исполнитель (обеспечение различного функционала и интерфейсов).

1.2 Взаимодействие с календарем (только для Менеджеров и Администраторов):

1.2.1 создание мероприятий (встреч) с указанием даты, времени, участников-Исполнителей:

1.2.1.1 нажатие пользователем на “Создать встречу”,

1.2.1.2 заполнение пользователем формы (название встречи, дата и время начала/окончания, список участников (Исполнителей, менеджеров), описание (опционально)),

1.2.1.3 рассылка системного уведомления участникам;

1.2.2 просмотр назначенных мероприятий:

1.2.2.1 отображение в календаре в виде блока с названием и временем,

1.2.2.2 при клике на мероприятие открывается полная информация: название, время, участники, описание;

1.2.3 просмотр графика отпусков:

1.2.3.1 отображение отпуска в календаре в виде специального выделения;

1.3 Взаимодействие с коллегами (кроме Администратора):

1.3.1 просмотр статуса коллеги (на работе/не на работе/в отпуске):

1.3.1.1 отображение рядом с каждым именем пользователя цветового индикатора, означающего его статус.

2. Функциональность для Менеджера:

2.1 Управление проектами:

2.1.1 создание:

2.1.1.1 заполнение пользователем формы с полями: название, описание,

2.1.1.2 заполнение системой полей: дата создания, Менеджер (владелец), статус (К выполнению),

2.1.1.3 создание системой списка задач (пустого);

2.1.2 редактирование:

2.1.2.1 возможность изменения названия, описания, статуса проекта;

2.1.3 просмотр проектов:

2.1.3.1 возможность просмотра общей информации проекта (описания),

2.1.3.2 возможность просмотра списка всех связанных задач и их статусов.

2.2 Управление задачами (задача имеет название, описание, статус, исполнителя, проект):

2.2.1 создание:

2.2.1.1 заполнение пользователем формы с полями: название, описание (оциально), Исполнитель (официально), проект, дедлайн,

2.2.1.2 заполнение системой полей: дата создания,

2.2.1.3 проверка уникальности названия задания в рамках проекта;

2.2.2 назначение Исполнителя:

2.2.2.1 выбор пользователя с ролью Исполнитель из выпадающего списка;

2.2.3 редактирование:

2.2.3.1 возможность изменения всех полей задачи, кроме даты создания;

2.2.4 просмотр статуса выполнения:

2.2.4.1 отображение статуса задачи в общем списке задач и при просмотре информации о задаче;

2.2.5 удаление задач:

2.2.5.1 удаление задач из системы с подтверждением действия.

2.3 Управление релизами:

2.3.1 назначение даты в календаре:

2.3.1.1 создание пользователем события в календаре типа "Релиз",

2.3.1.2 добавление системой назначенного события в календарь всем участникам проекта;

2.3.2 составление списка проектов на релиз,

2.3.2.1 создание пользователем списка проектов на релиз;

2.3.3 изменение статуса релиза:

2.3.3.1 выбор нового статуса из выпадающего списка («К выполнению», «В работе», «На ревью», «Отложен», «Отменен», «Завершен»).

2.4 Система уведомлений:

2.4.1 внутрисистемные уведомления для Менеджера при изменении статуса задачи Исполнителем:

2.4.1.1 накапливание уведомлений в «Центре уведомлений», где их можно посмотреть.

3. Функциональность для Исполнителя

3.1 Управление статусом задач:

3.1.1 изменение статуса назначеннной задачи:

- 3.1.1.1 «К выполнению» - задача создана Менеджером, но Исполнитель еще не начал работу,
- 3.1.1.2 «В работе» - Исполнитель приступил к выполнению,
- 3.1.1.3 «На ревью» - работа завершена, требуется проверка Менеджером,
- 3.1.1.4 «Отложен» - выполнение задачи временно приостановлено по различным причинам
- 3.1.1.5 «Отменен» - задача более не актуальна и больше не будет выполняться
- 3.1.1.6 «Завершена» - Менеджер принял работу.

3.2 Комментарии к задачам:

3.2.1 возможность добавлять текстовые комментарии к задаче:

- 3.2.1.1 написание пользователем комментария в поле ввода для текста и нажатие на кнопку "Добавить комментарий",
- 3.2.1.2 отображение комментария, имени автора и даты системой в информации о задаче.

4. Функциональность для Администратора

4.1 Управление сотрудниками:

4.1.1 регистрация новых сотрудников

- 4.1.1.1 ввод данных в форму для регистрации пользователем (ФИО, пароль, должность (выбор роли из списка: Менеджер/Исполнитель), рабочий график, дата рождения, контактный телефон, электронная почта),
- 4.1.1.2 проверка уникальности логина системой,
- 4.1.1.3 активация системой учетной записи и вход в систему при положительном ответе на запрос об уникальности;

4.2 Управление справочниками:

4.2.1 добавление и редактирование офисов, отделов;

4.3 Техническое обслуживание:

4.3.1 резервное копирование базы данных.

3.2 За рамками решения

1. Встроенный мессенджер для коммуникации. Причина: повышает сложность и объем работ, выходит за рамки основного фокуса проекта (управление задачами). Может быть реализован в будущих версиях. Компромисс в пользу соблюдения сроков (Время в "Треугольнике компромиссов").
2. Личные справочники/заметки исполнителей. Причина: не является критичной функцией для основного workflow управления проектами. Компромисс в пользу соблюдения сроков и ограничения объема функциональности (Время и Содержание в "Треугольнике компромиссов").
3. Установка персональных дедлайнов для задач в календаре (Исполнитель). Причина: не является обязательной для основного workflow управления проектами, где сроки контролируются менеджером. Повышает сложность синхронизации данных между модулями задач и календаря. Компромисс в пользу соблюдения сроков и ограничения объема функциональности (Время и Содержание в "Треугольнике компромиссов").
4. Возможность для Менеджера просматривать единый список всех задач across всех проектов. Причина: базовая навигация по системе уже предусматривает просмотр задач в контексте отдельных проектов, что достаточно для контроля workflow. Создание единого глобального отчета требует дополнительной сложности в разработке интерфейсов, фильтрации данных и пагинации, увеличивая объем работ. Компромисс в пользу соблюдения сроков и ограничения объема функциональности (Время и Содержание в "Треугольнике компромиссов").
5. 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.

3.3 Критерии одобрения решения

3.3.1 Основные критерии:

1. Все ключевые функции из раздела "Функциональность решения" реализованы и работоспособны.
2. Приложение успешно проходит базовое тестирование на создание проекта, постановку задачи, ее выполнение исполнителем и получение уведомления менеджером.
3. Интерфейс приложения интуитивно понятен для пользователей всех трех ролей.

3.3.2 Дополнительные критерии:

1. При удалении проекта автоматически удаляются все связанные с ним задачи и комментарии. Назначение несуществующего исполнителя или проекта невозможно.
2. Приложение выдерживает последовательные действия 3+ пользователей одновременно без критических ошибок или потери данных.
3. Исполнитель не может удалять или редактировать задачи других исполнителей. Менеджер не имеет возможности изменять статусы задач (кроме случаев отката или коррекции).
4. Решение включает минимально необходимую документацию для развертывания (README.md с инструкцией по запуску через docker-compose) и базовое описание API (сгенерированное FastAPI).
5. Отклик интерфейса на действия пользователя (открытие раздела, сохранение изменений) составляет не более 400 мс при скорости интернета от 100 Мбит/с.
6. Приложение может быть развернуто локально (с использованием Docker).

4 Стратегии дизайна решения

4.1 Стратегия архитектурного дизайна

Ниже представлена UML-диаграмма архитектуры (рисунок 2).

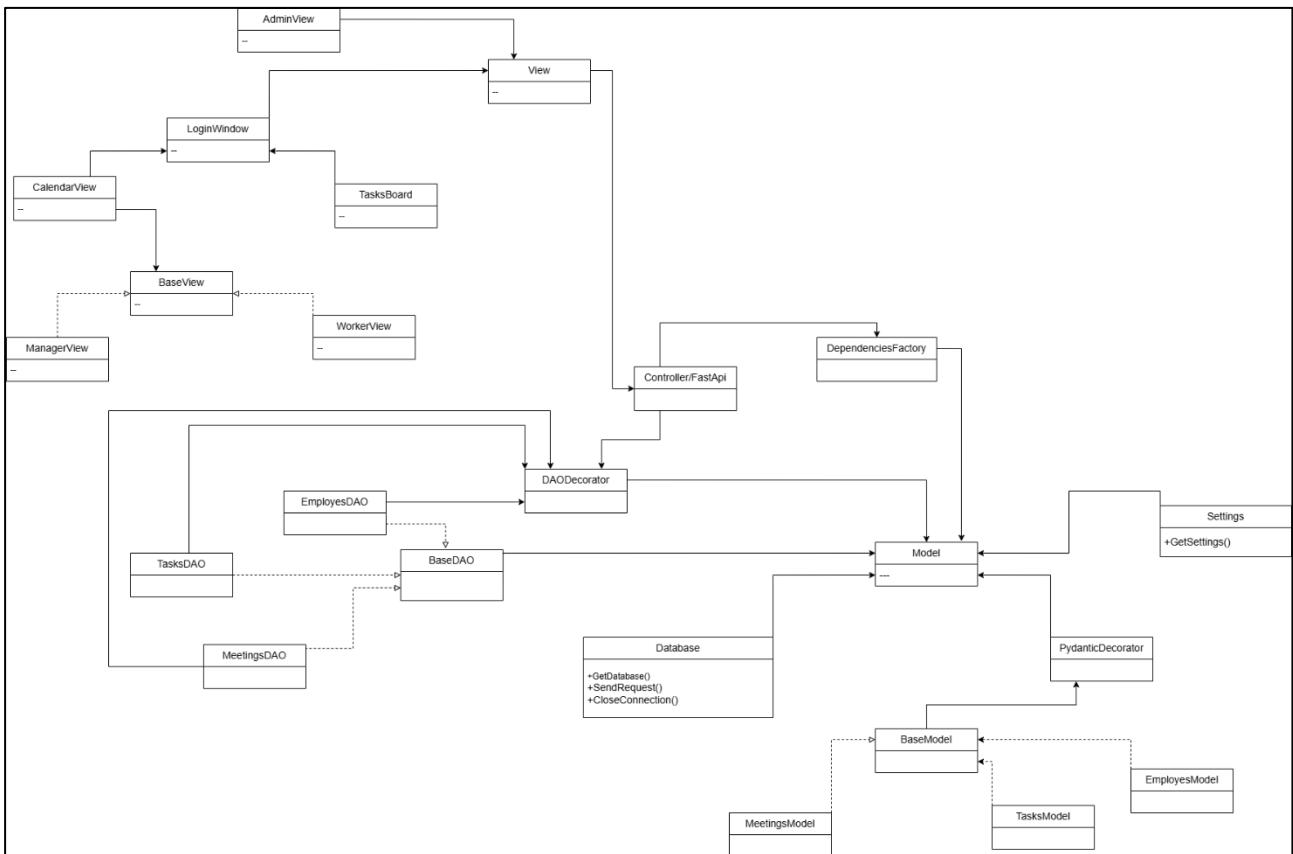


Рисунок 2 - UML-диаграмма архитектуры

[Ссылка на UML-диаграмму](#)

4.2 Стратегия технологического дизайна

4.2.1 Frontend:

1. React: современная, популярная библиотека для построения пользовательских интерфейсов с большим комьюнити и количеством готовых решений (UI-библиотек).
2. HTTP-клиент: Fetch API для взаимодействия с бэкенд-API.
3. Маршрутизация: React Router для организации навигации в SPA без перезагрузки страницы.

4.2.2 Backend:

1. Python + FastAPI: выбор обусловлен высокой производительностью FastAPI, простотой написания кода, автоматической генерацией документации OpenAPI и сильной типизацией, что снижает количество ошибок.
2. Аутентификация: JWT (JSON Web Tokens) для stateless-аутентификации.
3. ORM: SQLAlchemy в качестве абстракции для работы с базой данных, что обеспечивает безопасность типов данных и простоту миграций.
4. Миграции: Alembic для управления версиями схемы базы данных.

4.2.3 База данных:

PostgreSQL: надежная, полнофункциональная реляционная СУБД с поддержкой транзакций, внешних ключей и сложных запросов. Выбор обусловлен масштабируемостью, стабильностью работы в production-среде и соответствием требованиям по нагрузке при росте числа пользователей.

4.2.4 Развёртывание:

Docker, Docker Compose: позволит упаковать приложение в контейнеры, что гарантирует идентичность окружения при развертывании на любой машине и упрощает процесс поставки для компаний. Контейнеризация также нивелирует потенциальные проблемы с совместимостью сред между разработкой и продакшеном.