**Технический проект: Информационная система для управления проектами в детском технопарке «Кванториум».**

Настоящий документ описывает и обосновывает архитектурные, технологические и проектные решения для создания веб-ориентированной информационной системы для автоматизации управления проектной деятельностью, контроля выполнения задач и организации взаимодействия между руководителями и обучающимися.

**Функциональная структура**

Система реализует модульную архитектуру с четким разделением функциональности:

* модуль аутентификации и авторизации: управление доступом для учеников, руководителей и администраторов;
* модуль управления проектами: создание, редактирование, удаление проектов;
* модуль управления задачами: работа с задачами в виде досок, диаграмм Ганта и календаря;
* модуль командной работы: формирование команд, чаты, интерактивные доски;
* модуль аналитики: статистика выполнения задач, достижения учеников;
* модуль документооборота: прикрепление файлов к задачам.

Такое разделение обеспечивает логическую целостность системы, независимость модулей и возможность масштабирования. Для наглядного представления архитектуры используется диаграмма IDEF0 и ее декомпозиция.

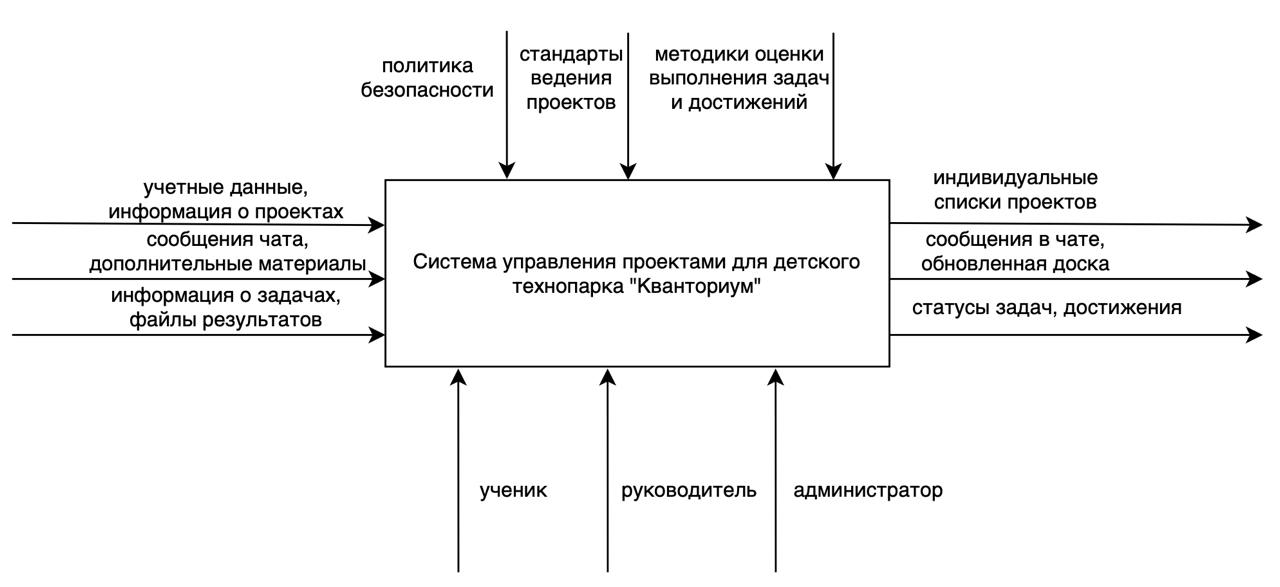


Рисунок 1 – IDEF0

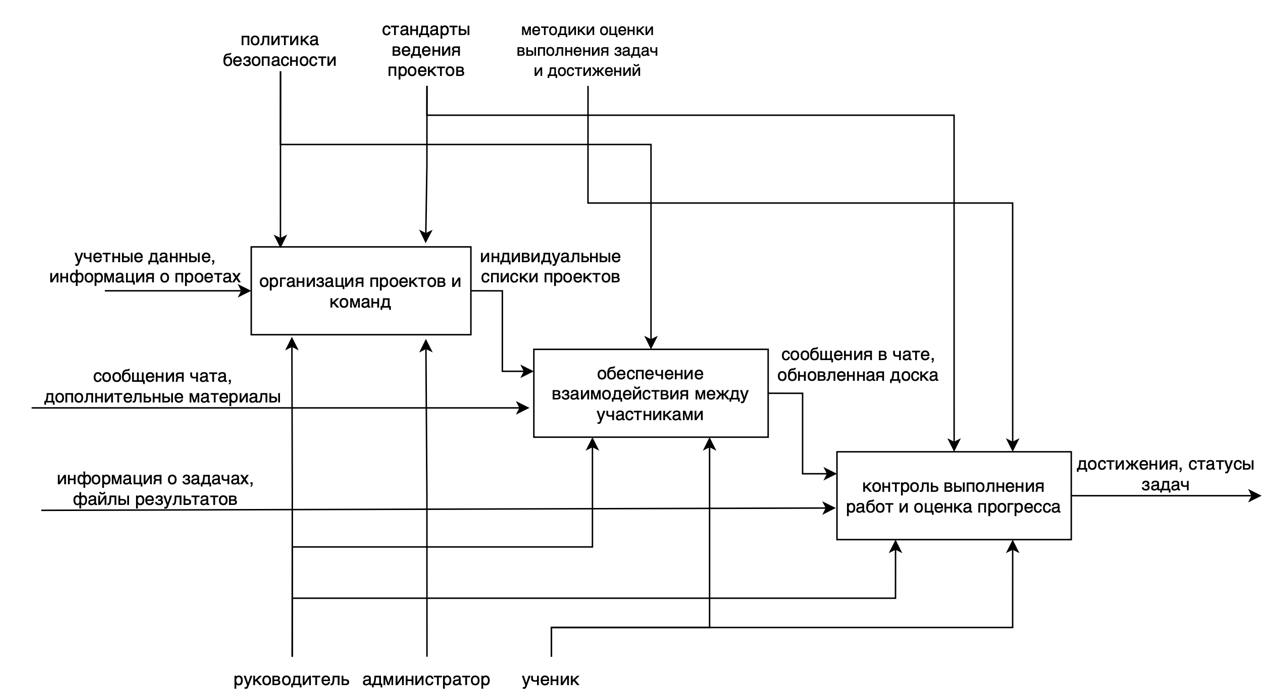


Рисунок 2 - Декомпозиция

**Спецификация процессов**

Поведение системы описывается через основные сценарии использования.

Use Case диаграмма определяет три ключевые роли пользователей: Ученика, Руководителя и Администратора, и описывает их взаимодействие с системой. Все функции начинаются с обязательного прецедента «Авторизоваться», который обеспечивает безопасный доступ в соответствии с ролью.

Ученик работает в рамках назначенных ему проектов: он просматривает список проектов и задач в различных представлениях (доска, диаграмма Ганта, календарь), выполняет задачи, загружая файлы с результатами, и общается с командой и руководителем через встроенный чат. Для мотивации ученик может просматривать свои достижения, которые назначаются руководителем, и изучать дополнительные материалы на интерактивной доске проекта.

Руководитель обладает расширенными правами по организации учебного процесса. Он управляет проектами (создание, редактирование, удаление), формирует проектные команды, назначая в них учеников, и управляет задачами (постановка, контроль, проверка результатов). Для поддержки командной работы руководитель общается в чате, управляет содержимым интерактивной доски, размещая учебные материалы, и фиксирует прогресс учеников, добавляя им достижения за успешное выполнение этапов работы.

Администратор выполняет функции централизованного управления системой. Он регистрирует и управляет учетными записями руководителей, а также осуществляет общее управление проектами и задачами. Ключевой уникальной функцией администратора является мониторинг и анализ активности через просмотр статистики выполнения задач за неделю, что позволяет оценивать общую продуктивность проектных команд в технопарке.

Таким образом, Use Case диаграмма комплексно отображает, как система обеспечивает полный цикл проектной деятельности — от постановки задач и коммуникации до контроля выполнения и анализа результатов, учитывая специфику образовательного процесса в «Кванториуме».

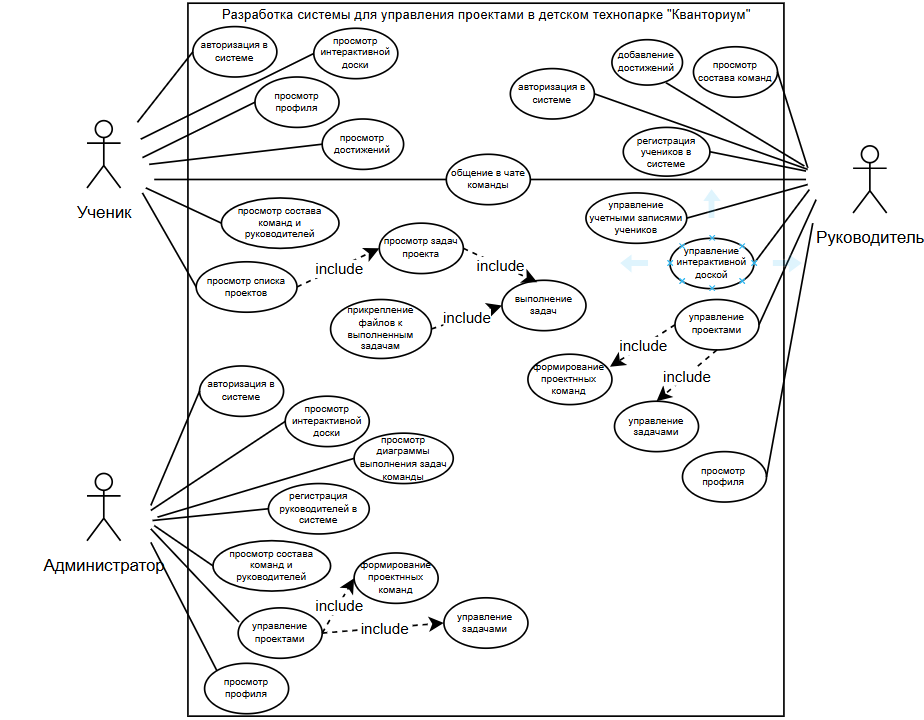


Рисунок 3 - Use Case диаграмма

**Структура хранимой информации**

Данные организованы вокруг сущностей: Проект, Пользователи, Команды, Задачи, Достижения.

Связи между сущностями:

* один проект содержит множество задач и связан с одной командой;
* каждая задача имеет конкретного исполнителя из числа пользователей и может содержать несколько прикрепленных файлов;
* команда объединяет нескольких пользователей для работы над проектом;
* элементы интерактивной доски привязаны к проектам и создаются руководителями;
* сообщения в чатах связаны с конкретными проектами и их авторами.

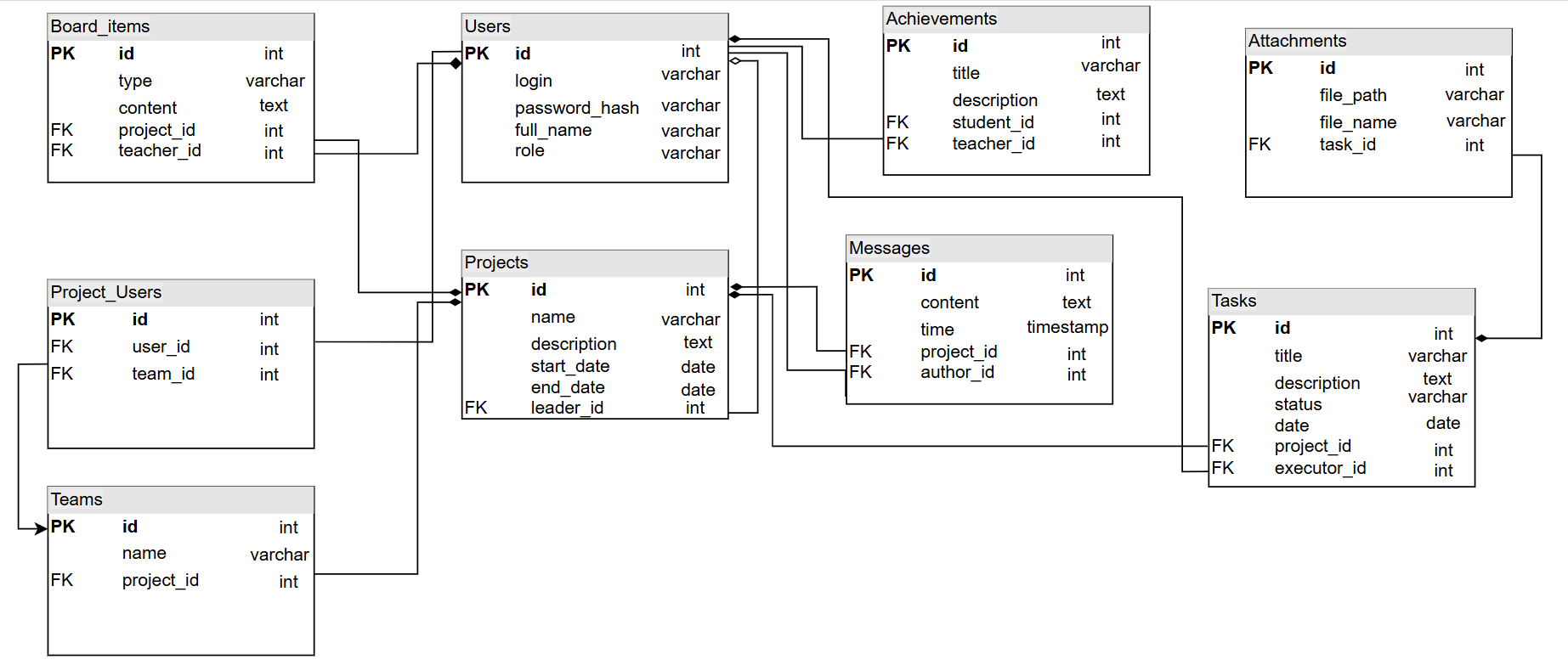


Рисунок 4 - Структура хранения данных

**Архитектурно-структурное решение**

Для реализации системы для управления проектами выбрана трехуровневая клиент-серверная архитектура, которая включает три уровня: уровень представления (веб-клиент), уровень бизнес-логики (сервер приложений) и уровень данных (сервер БД и база данных). Клиентская часть взаимодействует с сервером приложений через HTTP-запросы, а сервер приложений обменивается данными с СУБД через SQL-запросы. Реализация предусматривает использование React для клиентской части, FastAPI для сервера приложений и PostgreSQL для хранения данных.

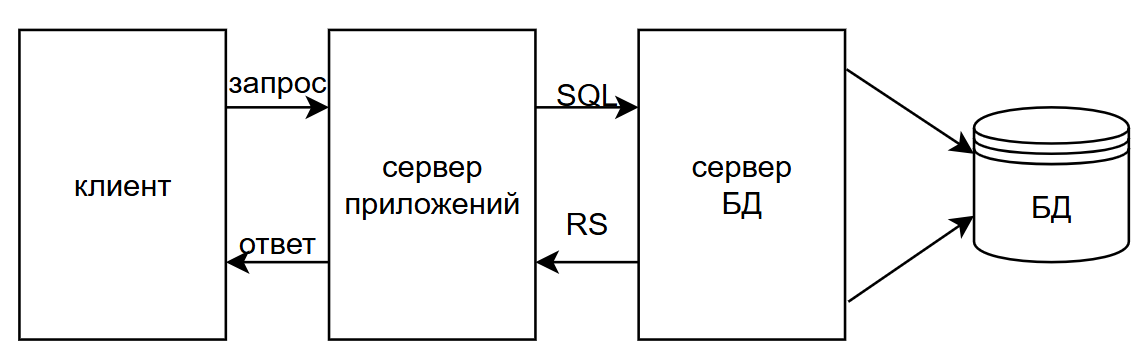


Рисунок 5 - Трехуровневая клиент-серверная архитектура