

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ДИАГРАММА ГАНТА ДЛЯ ПЛАНИРОВАНИЯ ГРАФИКА  
КОМАНДНЫХ РАБОТ**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 411 группы

направления 02.03.02 — Фундаментальная информатика и информационные  
технологии

факультета КНиИТ

Живодерова Дмитрия Александровича

Научный руководитель

зав. каф. тех. прог. \_\_\_\_\_

к. ф.-м. н., доцент

И. А. Батраева

Заведующий кафедрой

к. ф.-м. н., доцент \_\_\_\_\_

С. В. Миронов

Саратов 2026

## ВВЕДЕНИЕ

Планирование графика командных работ является практической задачей, возникающей при разработке программных продуктов, выполнении учебных проектов и организации небольших исследовательских работ. Даже небольшой проект становится менее управляемым, если сведения о задачах, сроках, исполнителях и текущем состоянии хранятся в переписке, списках или разрозненных таблицах. В такой ситуации участники видят отдельные поручения, но не всегда понимают взаимное расположение работ во времени, загрузку ресурсов и последствия переноса сроков.

Актуальность темы выпускной квалификационной работы определяется потребностью в простом, наглядном и технически понятном инструменте календарного планирования. Для малых команд и учебных проектов профессиональные системы управления проектами часто оказываются избыточными, а электронные таблицы при изменении плана требуют ручной корректировки диапазонов, формул и подписей. Поэтому практический интерес представляет веб-приложение, позволяющее структурированно описывать задачи, строить диаграмму Ганта и интерактивно редактировать календарный план.

Диаграмма Ганта представляет задачи в виде горизонтальных отрезков на временной шкале, где положение и длина блока отражают сроки выполнения работы. Такое представление позволяет оценивать последовательные и параллельные процессы, видеть пересечения сроков, состояние этапов и распределение работ по ресурсам. В данной работе диаграмма Ганта рассматривается также как объект программной реализации, для которого необходимо определить модель данных, построить временную шкалу, рассчитать координаты задач, обработать действия пользователя и сохранить изменения.

Цель работы — разработать веб-приложение для построения и редактирования диаграммы Ганта, предназначенное для планирования графика командных работ в малых и учебных проектах.

Для достижения поставленной цели рассмотрены основы планирования работ и назначение диаграммы Ганта, определён состав данных для её построения, сформулированы требования к приложению, выполнен обзор существующих решений, спроектированы структура данных, архитектура и интерфейс, реализованы клиентская и серверная части, алгоритм визуализации, демонстрационный пример и проверка основных пользовательских сценари-

ев.

Объектом исследования являются процессы календарного планирования работ в малых и учебных проектах. Предметом исследования являются подходы к представлению, обработке и визуализации данных диаграммы Ганта в веб-приложении.

При выполнении работы использовались методы анализа литературы и существующих решений, проектирования структуры данных и пользовательского интерфейса, прототипирования, программной реализации, календарных расчётов и тестирования. Клиентская часть приложения реализована на React с использованием TypeScript и Vite, оформление выполнено средствами Tailwind CSS. Для серверной обработки данных и работы с демонстрационной базой использованы Python, FastAPI, SQLAlchemy, Pydantic и SQLite.

Материалами исследования послужили научные и учебные источники по вопросам календарного планирования и визуализации проектных работ, существующие веб-сервисы для управления задачами, а также демонстрационный набор данных, содержащий сведения о задачах, сроках их выполнения, исполнителях, статусах и взаимном расположении работ во времени.

Структура выпускной квалификационной работы включает введение, две главы, заключение, список использованных источников и приложения. Первая глава «Теоретические основы и анализ аналогов диаграмм Ганта» посвящена вопросам планирования работ, особенностям диаграммы Ганта, данным для её построения, требованиям к современному веб-приложению и обзору существующих решений. Во второй главе «Проектирование и реализация» рассматриваются постановка задачи, проектирование структуры данных, архитектуры и интерфейса приложения, реализация клиентской и серверной частей, алгоритм построения диаграммы Ганта, демонстрационный пример и результаты тестирования.

Практическая значимость работы состоит в том, что разработанное приложение может использоваться как учебный пример реализации диаграммы Ганта и как основа для дальнейшего развития планировщика задач. В приложении показан путь данных от модели и серверного хранения до визуального отображения на временной шкале, а также реализованы изменение масштаба, перенос задач, изменение длительности, импорт и экспорт расписания, локальное сохранение и выявление конфликтов.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Первая глава посвящена теоретическим основам планирования работ, анализу диаграммы Ганта и сравнению существующих решений. Планирование рассматривается как процесс определения состава действий, сроков, последовательности этапов, исполнителей и ресурсов, необходимых для достижения результата. Для программных проектов это особенно важно, поскольку анализ требований, проектирование, реализация, тестирование и подготовка документации взаимосвязаны и могут выполняться параллельно. Без структурированного плана участники теряют общее представление о сроках, зависимостях и распределении работ.

В работе разграничиваются понятия планирования и управления задачами. Планирование определяет, какие работы и в какие сроки необходимо выполнить, а управление задачами связано с сопровождением плана: постановкой задач, назначением исполнителей, отслеживанием статусов, выявлением задержек и корректировкой сроков. Основной единицей планирования является задача, имеющая содержание, срок выполнения, ответственного исполнителя и критерий завершения. Связанные задачи могут объединяться в этапы проекта.

Диаграмма Ганта рассматривается как инструмент календарного планирования, отображающий задачи в виде горизонтальных блоков на временной шкале. Она позволяет оценивать длительность работ, видеть последовательные и параллельные процессы, пересечения сроков и текущее состояние проекта. В отличие от обычной таблицы, диаграмма наглядно показывает календарное расположение задач.

Для построения диаграммы необходимы структурированные данные. Минимальный набор включает название задачи, дату начала и дату окончания. Для практического использования также требуются идентификатор, статус, прогресс выполнения и связь с ресурсом. В расширенной модели могут учитываться приоритет, этап проекта, зависимости и контрольные точки. Ресурс в приложении соответствует строке диаграммы и может обозначать исполнителя, команду, направление работы или область проекта.

Качество визуализации зависит от полноты, согласованности и актуальности данных. Обязательные поля должны быть заполнены, даты не должны противоречить друг другу, а сроки и статусы необходимо своевременно обнов-

лять. Поэтому проверка данных должна выполняться как в пользовательском интерфейсе, так и на сервере.

На основе анализа предметной области сформулированы требования к веб-приложению, представленные в таблице 1. Система должна строить временную шкалу, поддерживать разные масштабы отображения, позволять создавать и редактировать задачи и ресурсы, переносить задачи, изменять их длительность, сохранять состояние, выполнять импорт и экспорт данных, выявлять конфликты и взаимодействовать с серверной частью. Нефункциональные требования связаны с читаемостью интерфейса, устойчивостью к ошибочным данным, производительностью и возможностью дальнейшего расширения.

Таблица 1 – Требования к веб-приложению для построения диаграммы Ганта

Требование	Содержание требования
Построение временной шкалы	Отображение календарного периода проекта с возможностью визуального размещения задач по датам начала и окончания.
Поддержка масштаба	Переключение масштаба диаграммы для просмотра плана на уровне дней, недель или месяцев.
Работа с задачами	Создание, редактирование и удаление задач с указанием названия, сроков, статуса, прогресса и связанного ресурса.
Работа с ресурсами	Добавление и изменение ресурсов, соответствующих строкам диаграммы и используемых для группировки задач.
Интерактивное редактирование	Перенос задач по времени и между строками, а также изменение длительности задачи с помощью интерфейса.
Сохранение данных	Сохранение текущего состояния расписания в локальном хранилище и передача данных на серверную часть.
Импорт и экспорт	Загрузка и выгрузка расписания в формате JSON для переноса данных и повторного использования плана.
Проверка корректности	Контроль обязательных полей, допустимых дат, статусов, прогресса выполнения и согласованности временных интервалов.
Выявление конфликтов	Обнаружение пересечений сроков задач, назначенных на один ресурс, и визуальное выделение конфликтующих элементов.
Взаимодействие с сервером	Выполнение операций создания, получения, изменения и удаления проектов и задач через REST API.
Удобство интерфейса	Обеспечение читаемости диаграммы, понятного расположения элементов управления и поддержки горизонтальной прокрутки.
Устойчивость к ошибкам	Предотвращение некорректных действий пользователя и вывод понятных сообщений при ошибочных входных данных.
Возможность расширения	Сохранение структуры приложения, допускающей последующее добавление зависимостей, приоритетов и контрольных точек.

В первой главе также рассмотрены существующие решения для календарного планирования. Профессиональные системы управления проектами поддерживают работу со сроками, ресурсами, зависимостями и отчётностью, но для небольших учебных проектов могут быть избыточными. Специализированные сервисы построения диаграмм Ганта предоставляют готовые средства визуализации, однако не раскрывают внутреннюю логику формирования шкалы и расчёта положения задач. Инструменты управления задачами удобны для работы с карточками и статусами, но не всегда ориентированы на календарное представление. Электронные таблицы доступны, однако при изменении плана требуют ручной корректировки визуальной части.

По результатам анализа сделан вывод о целесообразности разработки собственного веб-приложения. Его задача состоит не в конкуренции с профессиональными системами, а в демонстрации основных принципов построения диаграммы Ганта, организации данных и взаимодействия интерфейса с серверной частью.

Вторая глава посвящена проектированию и реализации веб-приложения для построения диаграммы Ганта. Оно ориентировано на малые и учебные проекты, где требуется быстро описать работы, распределить их по ресурсам, увидеть календарный план и внести изменения без ручного перестроения таблиц. Система визуализирует план и обеспечивает обработку структурированных данных.

Клиентская часть реализована на React с использованием TypeScript и Vite, оформление выполнено средствами Tailwind CSS. Серверная часть разработана на Python в виде REST API на FastAPI. Для описания моделей хранения применяется SQLAlchemy, для проверки входных данных — Pydantic, а в качестве демонстрационной базы данных используется SQLite.

В процессе проектирования определена структура данных приложения. Центральной сущностью является задача, содержащая идентификатор, название, связь с ресурсом, даты начала и окончания, статус, прогресс выполнения и дополнительные параметры редактирования. Ресурс определяет вертикальное положение задачи на диаграмме. В состоянии клиентской части также хранятся выбранный масштаб, видимый период, настройки временной шкалы и выбранная задача.

Серверная модель предназначена для хранения проектов и связанных

с ними задач. Проект выступает контейнером календарного плана, а задача содержит сведения о работе, сроках, статусе и прогрессе выполнения. Разделение клиентской и серверной моделей связано с тем, что интерфейс отвечает за визуализацию и интерактивное редактирование, а серверная часть — за хранение и проверку данных.

Архитектура приложения построена как взаимодействие интерфейса, централизованного состояния, вычислительных функций и серверного API. Основные компоненты архитектуры приведены в таблице 2. Пользовательские действия изменяют данные, после чего диаграмма автоматически перестраивается. Локальное сохранение позволяет восстанавливать состояние после обновления страницы, а серверная часть предоставляет операции создания, получения, изменения и удаления проектов и задач.

Таблица 2 – Основные компоненты архитектуры веб-приложения

Компонент	Назначение	Реализация в приложении
Пользовательский интерфейс	Отображение диаграммы и элементов управления	React-компоненты, панель действий, временная шкала, строки ресурсов и область диаграммы.
Централизованное состояние	Хранение текущих данных и параметров отображения	Список задач и ресурсов, выбранный масштаб, видимый период и выбранная задача.
Вычислительные функции	Преобразование календарных данных в визуальное представление	Расчёт диапазона дат, ширины единицы времени, смещения и размера блоков задач.
Механизм редактирования	Обработка пользовательских действий с задачами	Создание, изменение, перенос задач, смена ресурса и пересчёт длительности.
Локальное хранение	Восстановление состояния после обновления страницы	Сохранение расписания в браузере и загрузка сохранённых данных при повторном открытии приложения.
Серверное API	Обмен данными между клиентом и серверной частью	REST API на FastAPI для создания, получения, изменения и удаления проектов и задач.
База данных	Постоянное хранение проектов и задач	SQLite с моделями SQLAlchemy для демонстрационного хранения календарного плана.
Проверка данных	Предотвращение некорректного состояния расписания	Валидация входных данных на клиенте и сервере с использованием Pydantic-моделей.

При проектировании интерфейса учитывалось, что диаграмма Ганта содержит значительный объём визуальной информации. Поэтому экран разделён на панель действий, настройки масштаба, временную шкалу, строки ресурсов и рабочую область диаграммы. Пользователь может добавлять и

редактировать задачи, изменять масштаб, импортировать и экспортировать расписание, переходить к текущей дате и анализировать пересечения сроков. Для длинных периодов предусмотрена горизонтальная прокрутка.

Работа с данными реализована на нескольких уровнях. Импорт и экспорт расписания выполняются в формате JSON. При импорте проверяются структура файла, обязательные поля, типы идентификаторов, корректность дат, статусов и прогресса выполнения. При экспорте текущее состояние задач и ресурсов преобразуется в файл, пригодный для последующей загрузки. Локальное хранение используется для сохранения пользовательского плана в браузере, а серверная часть дополняет этот механизм хранением данных в SQLite.

Основой визуализации является алгоритм построения диаграммы Ганта. На вход алгоритма поступают задачи, ресурсы, выбранный масштаб и отображаемый период. Затем определяется календарный диапазон, формируется временная шкала, рассчитывается ширина единицы времени, а для каждой задачи вычисляются горизонтальное смещение и ширина блока. После этого задачи распределяются по строкам ресурсов и отображаются в соответствующих позициях.

При изменении масштаба пересчитывается только визуальное представление, а исходные календарные данные остаются неизменными. При переносе задачи по горизонтали изменяются даты начала и окончания, при переносе между строками изменяется связанный ресурс, а при перемещении границы блока пересчитывается длительность. Дополнительные проверки не допускают появления некорректных временных интервалов.

Для выявления конфликтов сравниваются интервалы задач, относящихся к одному ресурсу. Если периоды выполнения пересекаются, соответствующие задачи отмечаются как конфликтующие и выделяются в интерфейсе. Такой механизм не заменяет полноценную оптимизацию расписания, но позволяет быстро обнаружить возможную перегрузку ресурса.

Демонстрационный пример построен на проекте разработки веб-приложения диаграммы Ганта. Он включает анализ требований, проектирование структуры данных, реализацию интерфейса, подключение серверной части и тестирование. Для задач заданы сроки, статусы и прогресс выполнения, что позволяет показать размещение блоков на временной шкале и текущее состояние работ.

Таблица 3 – Результаты проверки пользовательских сценариев

Проверяемый сценарий	Ожидаемый результат
Построение диаграммы	Задачи отображаются на временной шкале в соответствии с датами начала, окончания и выбранными ресурсами.
Изменение масштаба	При переключении масштаба пересчитывается визуальное представление, а исходные календарные данные остаются неизменными.
Перенос задачи по времени	После перемещения задачи изменяются даты начала и окончания, а блок отображается в новой позиции.
Перенос задачи между ресурсами	При перемещении задачи в другую строку изменяется связанный ресурс и сохраняются сроки выполнения.
Изменение длительности	При изменении границы блока пересчитывается дата окончания или начала без образования некорректного интервала.
Импорт расписания	JSON-файл загружается только при корректной структуре, допустимых типах данных, датах, статусах и прогрессе.
Экспорт расписания	Текущие задачи и ресурсы сохраняются в JSON-файл, пригодный для последующей загрузки в приложение.
Сохранение состояния	После обновления страницы пользовательский план восстанавливается из локального хранилища браузера.
Проверка конфликтов	Пересекающиеся задачи одного ресурса обнаруживаются и выделяются в интерфейсе как конфликтующие.
Работа серверного API	Операции создания, получения, изменения и удаления проектов и задач выполняются через серверную часть.

В ходе тестирования были проверены основные пользовательские сценарии, представленные в таблице 3. Полученные результаты показали корректную работу функций построения и редактирования диаграммы, механизмов импорта и экспорта данных, локального сохранения, серверной валидации и выявления конфликтов. Проверка также подтвердила согласованность между изменениями данных и их визуальным отображением на временной шкале.

Итогом второй главы стала работоспособная версия веб-приложения, позволяющая управлять задачами и ресурсами, отображать календарный план, изменять сроки, выявлять пересечения и использовать локальное или серверное хранение данных. Основное значение разработки состоит в прозрачности реализации: в работе показано, как данные задач преобразуются в элементы диаграммы Ганта и как действия пользователя отражаются в состоянии приложения.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была рассмотрена и практически реализована задача построения диаграммы Ганта для планирования графика командных работ. Актуальность темы связана с тем, что малым и учебным проектам требуется простой и наглядный инструмент, позволяющий представлять сроки выполнения работ, распределять задачи по ресурсам и быстро изменять календарный план без использования сложных корпоративных систем управления проектами.

В теоретической части были рассмотрены понятия планирования работ и управления задачами, особенности диаграммы Ганта и состав данных, необходимых для её построения. Показано, что диаграмма Ганта удобна для визуального анализа длительности задач, последовательных и параллельных процессов, пересечений сроков и общего хода проекта. Также были сформулированы требования к веб-приложению для календарного планирования и выполнен обзор существующих решений. Проведённый анализ показал, что профессиональные платформы и специализированные сервисы обладают широкими функциональными возможностями, однако для учебной разработки часто оказываются избыточными либо не позволяют раскрыть внутренние принципы построения диаграммы и обработки данных.

В практической части разработано веб-приложение для построения и редактирования диаграммы Ганта. Были спроектированы структура данных, архитектура приложения и пользовательский интерфейс. Клиентская часть реализована на React, TypeScript и Vite, стилизация выполнена с использованием Tailwind CSS, а централизованное состояние организовано через Zustand. Серверная часть реализована на Python с использованием FastAPI, SQLAlchemy, Pydantic и SQLite. Она обеспечивает хранение проектов и задач, базовую проверку данных и взаимодействие с клиентской частью через REST API.

Отдельное внимание было уделено алгоритму построения диаграммы Ганта. Он включает определение видимого календарного периода, формирование временной шкалы, расчёт горизонтального смещения и ширины блоков задач, распределение задач по строкам ресурсов и выявление пересекающихся интервалов. В результате структурированные данные о задачах преобразуются в наглядное визуальное представление, которое автоматически

обновляется при изменении сроков, ресурсов или масштаба отображения.

Демонстрационный пример и проверка основных пользовательских сценариев показали корректную работу приложения. Поддерживаются изменение масштаба временной шкалы, перенос задач, изменение их длительности, импорт и экспорт данных в формате JSON, сохранение состояния в localStorage, проверка данных на стороне сервера и визуальная подсветка конфликтов. Это подтверждает согласованность между данными приложения, пользовательскими действиями и отображением календарного плана.

Таким образом, цель выпускной квалификационной работы достигнута. Разработанное приложение позволяет отображать задачи на временной шкале, анализировать сроки выполнения работ, редактировать календарный план, выявлять конфликты и демонстрировать взаимодействие клиентской и серверной частей веб-системы. Поставленные задачи, связанные с анализом предметной области, проектированием, программной реализацией и проверкой основных функций, были выполнены.

Практическая значимость результата состоит в том, что приложение может использоваться как учебный пример реализации диаграммы Ганта, средство демонстрации принципов календарного планирования и основа для дальнейшего развития более функциональной системы управления задачами. Прозрачная структура проекта позволяет проследить путь данных от модели и серверного хранения до расчёта координат и визуального отображения на временной шкале.

Дальнейшее развитие проекта может быть связано с добавлением авторизации пользователей, поддержкой совместной работы, расширенным управлением зависимостями между задачами, хранением ресурсов на стороне сервера, уведомлениями, аналитикой по срокам и загрузке исполнителей, а также расширением сценариев импорта и экспорта. Выполненная работа показывает, что современные веб-технологии позволяют создать понятный интерактивный инструмент визуального календарного планирования, подходящий для учебных и малых проектов.