

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра информационных систем управления

АЙАНА Илья Хайлемикаэлевич

**СИСТЕМА УПРАВЛЕНИЯ ЗАДАЧАМИ И РАБОЧИМ
ВРЕМЕНЕМ СОТРУДНИКОВ НА ПРЕДПРИЯТИИ**

Дипломная работа

Научный руководитель:
старший преподаватель
Орешко И.Г.

Допущена к защите

" ____ " _____ 2025 г.

Заведующий кафедрой информационных
систем управления

д-р техн. наук, доцент А. М. Недзьведь

Минск, 2025

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	6
ВВЕДЕНИЕ	7
ГЛАВА 1 ПРОБЛЕМА УПРАВЛЕНИЯ ПРОЕКТАМИ	9
1.1 Обзор литературы	9
1.2 Обзор и анализ существующих инструментов управления проек- тами	19
1.3 Постановка задачи	25
1.4 Выводы	26
ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ	27
2.1 Профили пользователей	27
2.2 Профиль задач	28
2.3 Профиль среды	29
2.4 Объектная модель	31
2.5 Выводы	33
ГЛАВА 3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	34
3.1 Архитектура системы	34
3.2 Реализация архитектуры	35
3.3 Методика применения системы	41
3.4 Выводы	41
ГЛАВА 4 ПРИМЕР ИСПОЛЬЗОВАНИЯ СИСТЕМЫ	42
4.1 Постановка задачи	42
4.2 Решение на основе методики	42
4.3 Выводы	46
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	48
ПРИЛОЖЕНИЕ А	48

РЕФЕРАТ

Дипломная работа, 50 страниц., 13 таблиц, 19 иллюстраций, 11 источников, 1 приложение.

Ключевые слова: УПРАВЛЕНИЕ ЗАДАЧАМИ, РАБОЧЕЕ ВРЕМЯ, ЭФФЕКТИВНОСТЬ, ОРГАНИЗАЦИЯ, АВТОМАТИЗАЦИЯ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.

Объектом исследования является система управления задачами, направленная на оптимизацию распределения рабочего времени и повышение эффективности работы сотрудников на предприятии.

Предметом исследования являются методы и технологии управления задачами и рабочим временем сотрудников на предприятии.

Целью исследования является разработка системы, способной автоматизировать управление задачами и контролировать рабочее время сотрудников, выявлять узкие места в процессах, а также повышать общую продуктивность команды.

Методами исследования являются методы анализа существующих методик и программных решений в области управления задачами и рабочего времени. В качестве основы для разработки программной части применены современные языки программирования и инструменты, такие как Django, PostgreSQL и др.

Полученные результаты и их новизна: выявлены ключевые проблемы, связанные с управлением задачами и учетом рабочего времени сотрудников. Разработана модель системы управления задачами, а также программные компоненты для планирования, мониторинга и анализа рабочего времени. Проведено тестирование предложенных решений.

Достоверность материалов и результатов дипломной работы: использованные материалы и результаты дипломной работы являются достоверными. Работа выполнена самостоятельно.

Областью возможного практического применения является эффективное управление задачами и рабочим временем сотрудников на предприятиях, что способствует повышению производительности, улучшению организационной структуры и снижению временных затрат на выполнение задач.

РЭФЕРАТ

Дыпломная праца, 50 старонак, 13 табліц, 19 ілюстрацый, 11 крыніц, 1 дадатак.

Ключавыя словы: КІРАВАННЕ ЗАДАЧАМІ, РАБОЧЫ ЧАС, ЭФЕКТЫЎНАСЦЬ, АРГАНІЗАЦЫЯ, АЎТАМАТЫЗАЦЫЯ, ПРАГРАМНАЕ ЗАБЯСПЕЧАННЕ.

Аб'ектам даследавання з'яўляецца сістэма кіравання задачамі, накіраваная на аптымізацыю размеркавання рабочага часу і павышэнне эфектыўнасці працы супрацоўнікаў на прадпрыемстве.

Прадметам даследавання з'яўляюцца метады і тэхналогіі кіравання задачамі і рабочым часам супрацоўнікаў на прадпрыемстве.

Мэтай даследавання з'яўляецца распрацоўка сістэмы, якая можа аўтаматызаваць кіраванне задачамі і кантраляваць рабочы час супрацоўнікаў, выяўляць вузкія месцы ў працэсах, а таксама павышаць агульную прадукцыйнасць каманды.

Метадамі даследавання з'яўляюцца метады аналізу існуючых метадык і праграмных рашэнняў у галіне кіравання задачамі і рабочым часам. У якасці асновы для распрацоўкі праграмнай часткі ўжыты сучасныя мовы праграмавання і інструменты, такія як Django, PostgreSQL і інш.

Атрыманыя вынікі і іх навізна: выяўлены ключавыя праблемы, звязаныя з кіраваннем задачамі і ўлікам рабочага часу супрацоўнікаў. Распрацавана мадэль сістэмы кіравання задачамі, а таксама праграмныя кампаненты для планавання, маніторынгу і аналізу рабочага часу. Праведзена тэставанне прапанаваных рашэнняў.

Даставернасць матэрыялаў і вынікаў дыпломнай працы: выкарыстаныя матэрыялы і вынікі дыпломнай працы з'яўляюцца даставернымі. Праца выканана самастойна.

Вобласцю магчымага практычнага прымянення з'яўляецца эфектыўнае кіраванне задачамі і рабочым часам супрацоўнікаў на прадпрыемствах, што спрыяе павышэнню прадукцыйнасці, паляпшэнню арганізацыйнай структуры і зніжэнню часавых выдаткаў на выкананне задач.

ABSTRACT

Diploma work, 50 pages, 13 tables, 19 illustrations, 11 sources, 1 appendix.

Keywords: ORGANIZATION, TASK MANAGEMENT, EFFICIENCY, WORK TIME, SOFTWARE, AUTOMATION.

The object of the research is a task management system aimed at optimizing the distribution of working time and improving the efficiency of employees in an enterprise.

The subject of the research is the methods and technologies of task management and working time control for employees in an enterprise.

The purpose of the research is to develop a system capable of automating task management and monitoring employees' working time, identifying bottlenecks in processes, and increasing the overall productivity of the team.

Methods of research include analysis of existing methodologies and software in the field of task and time management. Modern programming languages and tools such as Django, PostgreSQL, etc., were used as the basis for the software development.

The results of the work and their novelty: key problems related to tracking working time and task management were identified. A model of the system was developed, along with software components for planning, monitoring, and analyzing working time. Testing of the proposed solutions was conducted.

Authenticity of the materials and results of the diploma work: the materials used and the results of the diploma work are authentic. The work was completed independently.

Recommendations on the usage: effective management of tasks and working time of employees in enterprises, which contributes to increased productivity, improved organizational structure, and reduced time costs for task completion.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

PMI (Project management institute) – Институт управления проектами.

CPM (Critical Path Method) – Метод критического пути.

CCPM (Critical Chain Path Method) – Метод критической цепи.

PMLC (Project Management Lifecycle) – Жизненный цикл управления проектами.

PMBoK (Project Management Body of Knowledge) – Свод знаний по управлению проектами.

KPI (key performance indicators) – Ключевые показатели эффективности.

ORM (Object-Relational Mapping) – Объектно-реляционное отображение.

ВВЕДЕНИЕ

Современный мир характеризуется высокой конкуренцией и стремительными изменениями бизнес-среды. Компании постоянно появляются и исчезают. Успешные организации отличаются рядом факторов, которые им позволяют держаться на плаву. Одним из таких аспектов является грамотное управление проектами, задачами, временем и другими ресурсами. В условиях жесткой конкуренции и необходимости повышения прозрачности и подотчетности возрастает спрос на операционную эффективность и достижение реальных результатов.

Добиться высокой эффективности и результативности можно, внедряя проверенные практики, которые помогают оптимально распределять ресурсы организации.

Данные показывают, что 85–90% проектов либо выходят за сроки, либо превышают бюджет, либо не дотягивают до нужного качества [1]. Почему так происходит? Основные причины:

- проект начинают без чёткого понимания его ценности для бизнеса;
- цели формулируют размыто, и разные участники понимают их по-своему;
- коммуникация между заинтересованными сторонами налажена плохо;
- ожидаемые результаты не имеют конкретных измеримых критериев;
- качество работы проверяют с большими перерывами или слишком поздно;
- сроки и бюджет оценивают слишком оптимистично;
- роли в команде распределены нечётко — кто за что отвечает;
- ресурсы планируют «по ходу дела», без должной координации.

Исследования, проведенные Джоном Робертсом и Джонатаном Фурлонгером [2], посвящённые проектам в сфере информационных систем, показывают, что применение детализированной методологии управления проектами повышает производительность на 20–30%. Более того, системный подход к управлению проектами помогает:

- ясно определить границы проекта — что входит в работу, а что нет;
- согласовать цели между всеми участниками;
- заранее понять, какие ресурсы понадобятся;
- сделать работу команды более прозрачной;
- держать в фокусе конечную пользу для бизнеса.

Цель работы состоит в разработке системы управления задачами и рабочим временем, которая эффективно решает выше озвученные проблемы.

Задачи работы:

- анализ существующих методологий и инструментов управления проектом с целью выявления ключевых возможностей и особенностей будущей системы;
- проектирование системы, с целью конкретизации образа системы, которую мы хотим получить;
- разработка прототипа системы, который соответствует функциональным и нефункциональным требованиям;
- демонстрация применения системы на практическом примере.

ГЛАВА 1

ПРОБЛЕМА УПРАВЛЕНИЯ ПРОЕКТАМИ

1.1 Обзор литературы

Понятием, которое стоит во главе сферы управления проектами, является собственно проект. Если следовать определению, которое дает Институт управления проектами (IPM) в своем руководстве РМВоК, то проект — это «временное предприятие, направленное на создание уникального продукта, услуги или результата». Проект обладает следующими характеристиками:

- каждый проект уникален. Да, у разных проектов могут быть общие элементы, например общие материалы или проектами могут заниматься одна и та же команда. Однако по своим главным особенностям объект остается уникальным (например, строительство офисного здания является уникальным проектом, ибо каждое такое здание обладает уникальным местоположением, окружающей средой и т.д.);

- проект – это временное предприятие, то есть он имеет какие-то временные рамки, начало и конец. То, что проект является временным не означает, что проект рассчитан на короткое время. Проект заканчивается тогда, когда достигнуты поставленные цели (или же эти самые цели не могут быть достигнуты), или же потребность в проекте как таковом отпала, или же человеческие и материальные ресурсы были исчерпаны, или же финансирование на осуществление проекта исчерпано и т.д.;

- проект рассчитан конкретно измеримый результат. Причем успех этого результата можно оценить по заранее согласованным критериям.

В любом проекте существует ряд ограничений, которые борются за внимание менеджера проекта. Это стоимость, объем, качество, риск, ресурсы и время [4]. Эти шесть пунктов часто сокращенно называют «тройным ограничением» - время, стоимость и объем, - изображаемым в виде треугольника, чтобы визуализировать работу над проектом и увидеть взаимосвязь между объемом/качеством, графиком/временем и стоимостью/ресурсами (рисунок 1.1).

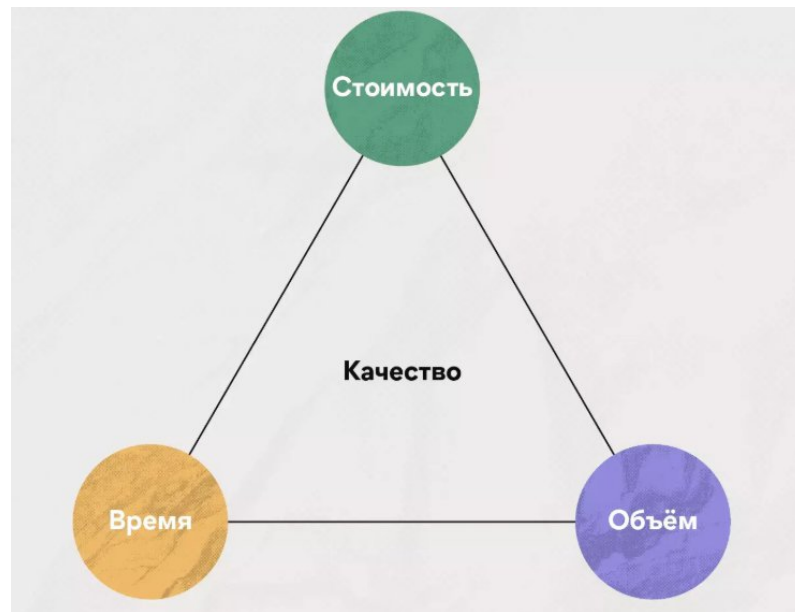


Рисунок 1.1 – Треугольник управления проектом

В этом треугольнике каждая сторона представляет собой одно из ограничений (или связанных с ним ограничений), при этом любые изменения одной стороны приводят к изменению других сторон. Лучшие проекты имеют идеально сбалансированный треугольник. Поддерживать этот баланс сложно, поскольку проекты подвержены изменениям.

Объем проекта - это его размер с точки зрения качества, детализации и масштаба результатов. По мере увеличения масштаба проекта для его завершения, естественно, требуется больше времени и денег. Элементы объема проекта могут включать в себя: сложность проекта, качество продукции, количество готовой продукции, уровень детализации, мощность (например, количество одновременных пользователей, которое может поддерживать приложение) и т.д.

В контексте проектного треугольника стоимость не ограничивается буквальными долларовыми суммами. Часто называемая «ресурсами», эта точка треугольника охватывает все инструменты, оборудование и поддержку, необходимые для завершения проекта. Элементы стоимости могут включать в себя: финансовый бюджет, количество членов команды, оборудование и оснащение, ключевые возможности и т.д.

Под временем подразумевается время, необходимое для завершения проекта. Это самый часто упускаемый из виду аспект разработки проекта, что отражается в пропущенных сроках и неполной готовности. Ко времени можно отнести: количество часов, отработанных в рамках проекта, общую хронологию проекта, время на планирование и выработку стратегии, количество этапов проекта,

внутренние календари и ориентиры и т.д.

Для эффективного управления проекта, руководителю нужно держать эти три переменные в виду и иметь о них четкое представление, чтобы он мог своевременно отвечать на изменения в рамках всего проекта. Например, если нужно увеличить объём, нужно будет также увеличить стоимость или время.

К сожалению, дело управления проектами связано с рядом трудностей.

Одной из таких проблем является плохое планирование. Хорошее планирование включает проработку рисков, которые могут быть связаны с деятельностью проекта. Такие риски есть у каждого крупного проекта, а посему к ним стоит подготовиться заранее. К сожалению, менеджерам не всегда могут тщательно проработать план. Часто это связано с давлением со стороны людей, занимающих вышестоящие должности. На выходе имеем непроработанный план, который сотрудники могут даже проигнорировать, так как считают это тратой времени. Это может привести к проблемам, особенно на последних этапах разработки, так как все цели и задачи часто связаны друг с другом, например одна задача не может быть выполнена без выполнения другой.

Также существует проблема, когда план слишком детализированный, так как менеджеры часто не имеют полного представления о том, сколько та или иная задача может занять. Члены команды могли бы составить план сами, но им может не хватать знаний о том, как это сделать, или же им не хватает полномочий, ибо они могут планировать только свою часть работы.

Еще одна трудность, с которой часто сталкиваешься, это нечеткие цели и задачи. Они считаются таковыми, если они недостаточно определены в начале проекта, или же если требования к продукту были некорректно проинтерпретированы. Также описание проекта может и не учитывать факторы, которые выходят за рамки его непосредственной реализации. То есть команда может понимать, какие функции должен выполнять продукт, но в то же время может и не учитывать другие аспекты, например интеграцию в заданную среду, удобство пользователя и т.д.

Помимо того, что цели могут быть нечетки, они могут еще меняться в процессе разработки. Это может быть связано, например, с расширением масштабов проекта или же с разрастанием функционала, когда добавляют слишком необязательных функций, из-за чего продукт становится перегруженным. Это часто связано с желанием сделать продукт более полезным.

Также при реализации проекта часто недооценивают время и ресурсы, которые необходимы для завершения проекта, так как менеджеры не учитывают возможные трудности, которые могут возникнуть во время разработки. Еще бывает, когда для ускорения проекта в команду добавляют больше людей для повышения производительности, однако на это приводит к обратному исходу. Увеличивается сложность взаимодействия между членами команды, снижается ответственность, а следовательно и мотивация и т.д. Это так называемый эффект Рингельмана.

Одной из главных проблем в сфере управления проектами также является недостаток вовлеченности клиентов в процесс разработки. Клиент должен активно участвовать в реализации продукта, потому что в противном случае он рискует получить не то, что хотел.

Не стоит забывать и о плохой коммуникации в команде. Каждый член команды сосредоточен на своих задачах, а не на общей картине. Это может привести к отклонению от целей, недооценке вклада отдельных членов команды и т.д.

Все это может вылиться в то, что члены команды не успевают завершить проект в срок или не выполняют требования клиента. Это приводит к задержкам и перерасходу средств. Итоговый продукт может не реализовывать все функциональные и нефункциональные требования.

Проекты, а следовательно и область управления проектами, существовали еще с давних времен. Построить пирамиду или Китайскую стену без какого-либо плана довольно затруднительно. Однако современное представление о проектах сформировалось только ближе к концу 19 века, когда люди стали использовать всевозможные графики и диаграммы.

Вот возьмем так называемую гармонограмму, которая широко применялась с 1986 года и считается одним из первых известных графических отображений системы контроля потоками работ. Операции отображаются на ней с помощью полосок бумаги, которые обозначают последовательно идущие задачи в процессе. К этим полоскам бумажные ярлыки для измерения времени, которое затрачивается на выполнение задачи.

Гармонограммы, к сожалению, особо распространены не были, так как работы автора Карола Адамецкого были опубликованы только на русском и польском. Повсеместно же стали использоваться так называемые диаграммы Ганта, изобретенные в 1910 г. Эти диаграммы имеют много общего с ранее упомянуты-

ми гармонограммами, однако рабочие процессы тут представлены по-другому: горизонтальные полосы используются для отображения времени выполнения задачи, ее фаз и зависимостей, сами же задачи размещены по вертикали.

Настоящий прорыв произошел в 1950-х годах, когда в США появились два революционных метода:

- Метод критического пути, суть которого состоит в нахождении критического пути. Критический путь – это самая длинная последовательность работ, которые требуется выполнить, чтобы закончить проект в срок;

- PERT — метод оценки и анализа проектов. Был разработан по заказу ВМС США для управления для проекта создания ракетной системы «Поларис». Самой известной частью данного метода является диаграмма, показывающая взаимосвязь работ и событий. Диаграмма состоит из блоков. Блок воплощает собой конкретную задачу с порядковым номером исполнения. Между этими блоками устанавливают зависимость стрелками. Начало и завершение проекта связаны многочисленными путями с разной длиной. Путь с наибольшей длиной будет являться критическим.

С появлением компьютеров инструменты проектного управления стали цифровыми. Одним из первых является Artemis, разработанный в 1977 г. Однако только в 1980-х годах, благодаря появлению персональных компьютеров и запуску Microsoft Project в 1984 году, программное обеспечение для управления проектами стало по-настоящему популярным. Особенностью Microsoft Project являлся дружелюбный интерфейс, а также интеграция с другими приложениями Microsoft Office.

Главной явлением в 1990-ых годах стало массовое распространение интернета, что изменило то, как руководители подходят к планировке проектов, составлению расписаний, отслеживанию выполнения задач и т.д. Конечно же появились и веб-инструменты для управления проектами, например Basecam, запущенный в 1999.

В 2000-ых революцию в области управления проектами произвели так называемые гибкие методологии. Традиционные подходы к разработке имеют свои особенности, будь то тщательное планирование проектов или жесткие механизмы контроля. Они подходят, если ожидаемый результат очевиден и понятен, что не всегда так, особенно в области по созданию программного обеспечения. Гибкие методологии же решают эту проблему с их итеративностью, прототи-

пированием, а также активным вовлечением заказчика в этап разработки.

Наконец, 2010-ые года. Если в 1990-ых был интернет, а в 2000-ых гибкие методологии, то в этом десятилетии подобным открытием стали облачные технологии, благодаря которой команды могут работать где угодно и как угодно.

Основой успешного управления проектами является жизненный цикл проекта, который включает пять взаимосвязанных этапов: инициацию, планирование, исполнение, мониторинг и контроль, а также завершение [6]. Каждый из этих этапов имеет свои ключевые задачи и результаты, обеспечивая последовательное продвижение проекта от идеи до реализации (рисунок 1.2).

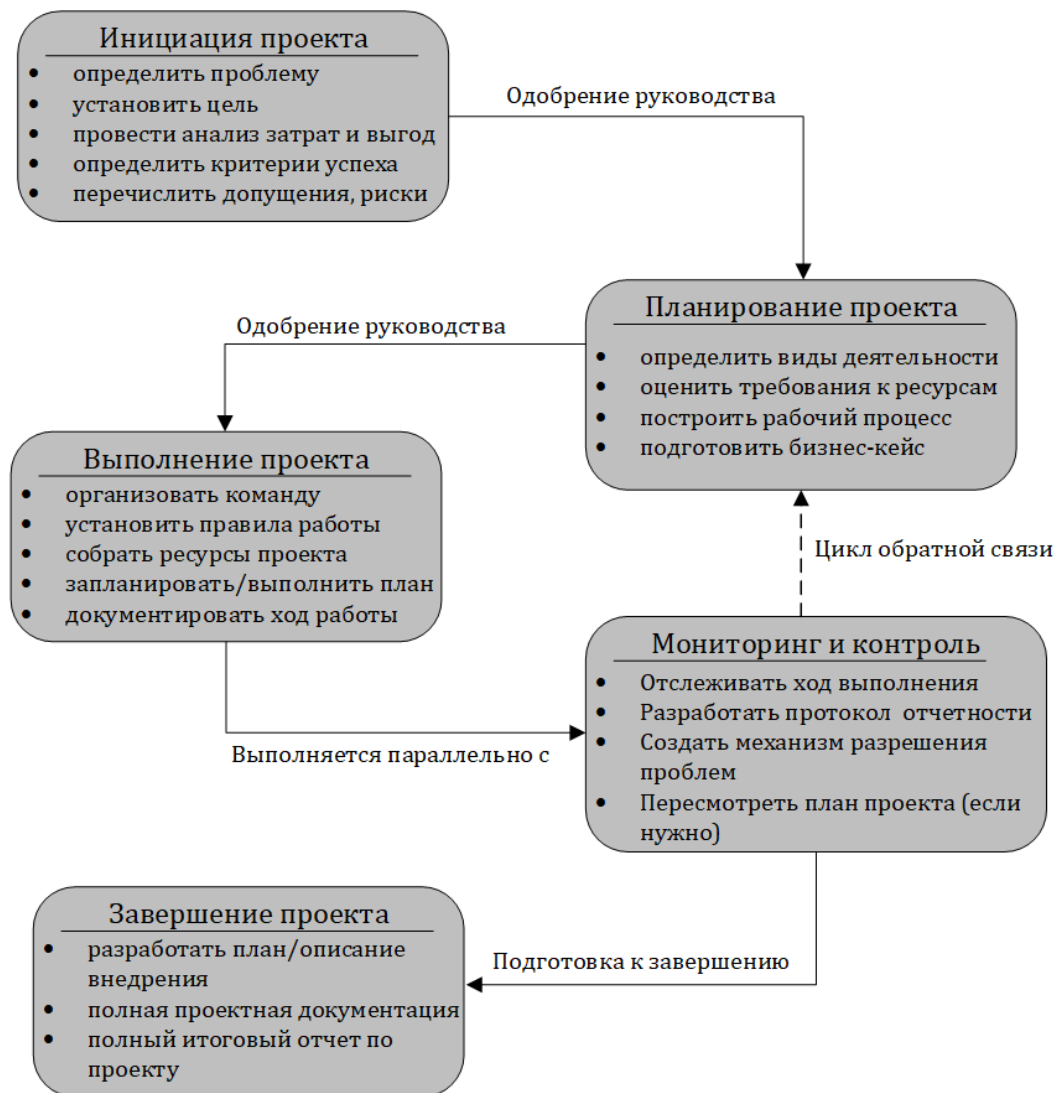


Рисунок 1.2 – Жизненный цикл проекта

Полный жизненный цикл проекта начинается с фазы инициации, целью которой является определение масштаба проекта. Результатом фазы инициации является документ, называемый бизнес-кейсом, в котором описывается проблема, которую будет решать проект, цель и задачи проекта, понесенные затраты

и получаемые выгоды, способы оценки успеха, а также потенциальные риски и препятствия, с которыми можно столкнуться [7].

После того как проектное предложение получило одобрение руководства организации, начинается этап планирования. Определяются основные результаты, а также начинает формироваться команда проекта. Хотя большинство действий на этом этапе выполняются одним или несколькими людьми, которые составят ядро проектной команды, общепринятой практикой является проведение официальной сессии планирования для всех заинтересованных сторон, которые будут влиять на проект или будут затронуты им. Результатом планирования является подробный план проекта, содержащий описание каждой деятельности проекта, ресурсы, необходимые для ее выполнения, график проекта, различные этапы получения промежуточных результатов, а также даты проведения приемочных испытаний и окончательной сдачи продуктов проекта.

На этапе выполнения команде проекта предстоит выполнить задачи и подготовить результаты, описанные в проектной документации. Команда должна уделять особое внимание производству и качеству результатов работы команды, а также следить за тем, чтобы работа выполнялась с должным уровнем дисциплины и строгости, чтобы снизить риски проекта и соответствовать стандартам качества.

Под мониторингом и контролем понимается совокупность процессов, которые происходят на протяжении всего проекта, чтобы обеспечить постоянный пересмотр задач проекта для поддержания качества. Процессы мониторинга и контроля включают все задачи и метрики, необходимые для обеспечения того, чтобы утвержденный проект осуществлялся с минимальным риском, в рамках объема, в срок и в рамках бюджета. Эти процессы включают в себя сравнение фактических показателей с запланированными и принятие корректирующих мер для достижения желаемого результата при наличии значительных расхождений.

На этапе завершения проекта продукт принимается и передается в организацию, либо принимается решение о приостановке или отмене проекта. Также на данном этапе происходит составление итогового отчета по проекту, в котором обобщаются все отчеты о ходе проекта, результаты приемки тестов и краткое описание извлеченных уроков.

Для эффективного управления проектами существуют различные методо-

логии. Менеджеры следуют этим стандартам, когда задаются вопросами: как оценивать сроки, как разбить проект на подзадачи, как организовать взаимодействие между сотрудниками, как оценивать результаты и т.д. Еще методологии предполагают использование тех или иных инструментов управления проектами [8].

Среди типов методологий можно выделить традиционные и гибкие методологии.

Традиционные методологии управления проектами основаны на последовательном выполнении задач. Это, например, каскадная модель, также известная как Waterfall. В ней мы представляем проект в виде последовательности этапов, при этом пропускать этапы или возвращаться к прошлым фазам нельзя. Исправление всевозможных ошибок происходит только под конец на этапе тестирования. Этот метод хорошо подходит, когда результат предсказуем и понятен, однако в реальности это не всегда так. Например, если речь идет о программном обеспечении, то клиенту придется до мельчайших деталей объяснить, какой функционал должен быть в приложении, что под силу далеко не каждому, ибо у заказчика может и не быть целостного представления о том, что он хочет получить. Также из-за последовательной натуры каскадной модели клиент может оценить результат только в самом конце, в процесс разработки он не вовлекается. Из-за этого заказчик может в конце и не получить того, на что он рассчитывает.

Стоит отметить, что речь выше о "чистой" каскадной модели. Также имеются и версии waterfall с элементами гибкой методологии, то есть с прототипированием, циклами обратной связи и т.д.

Все недостатки, характерные для традиционных подходов, отсутствуют в гибких методологиях, или же методологиях Agile. Эти стандарты основываются на следующих ценностях:

- люди и взаимодействие важнее процессов и инструментов;
- работающий продукт важнее исчерпывающей документации;
- сотрудничество с заказчиком важнее согласования условий контракта;
- готовность к изменениям важнее следования первоначальному плану.

Разницу между традиционными и гибкими методологиями хорошо демонстрирует рисунок 1.3.

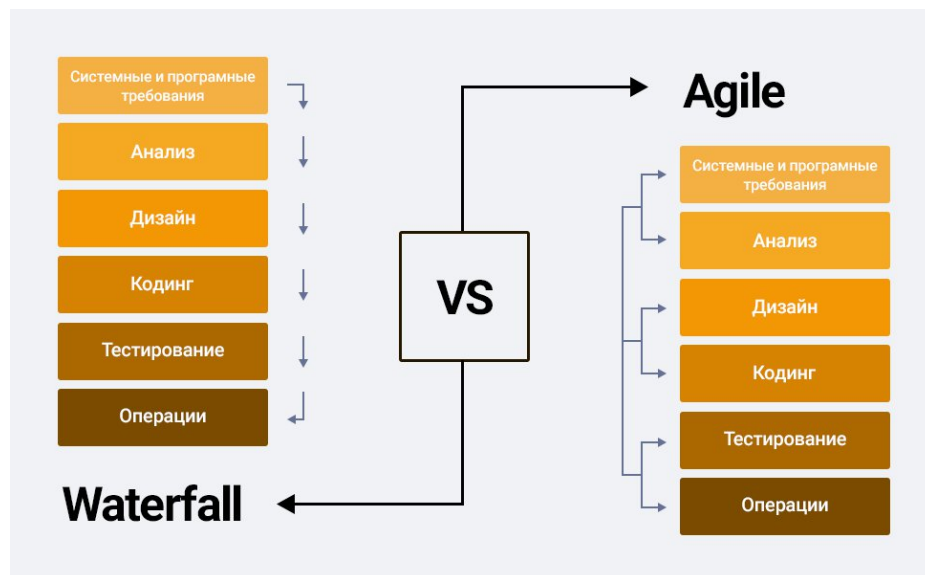


Рисунок 1.3 – Традиционные и гибкие методологии

Гибкие методологии следуют двенадцати принципам, среди которых я бы выделил работающий продукт как основной показатель прогресса, частый выпуск этого самого продукта, с периодичностью от пары недель до пары месяцев, сотрудничество между разработчиками и заказчиком на протяжении всего цикла разработки, а также приветствование изменения требований даже на позднем этапе разработки.

Среди самых распространённых Agile-методик можно выделить:

- Scrum. Scrum является одной из самой популярной методологий, потому что ее относительно просто внедрить. Кроме того, она решает многие проблемы, с которыми в прошлом боролись разработчики программного обеспечения, будь то циклы разработки, негибкие планы проектов и смещение сроков производства. В Scrum небольшую команду возглавляет Scrum-мастер, основная задача которого - устранить все препятствия на пути к эффективной работе. Команда работает короткими циклами по две недели, называемыми «спринтами», при этом члены команды встречаются ежедневно, чтобы обсудить свою работу и все препятствия, которые необходимо устранить. Эта методология позволяет быстро разрабатывать и тестировать, особенно в небольших командах.

- Kanban, еще одна популярная методология. Проектные команды создают визуальные представления своих задач, часто используя стикеры и доски (или онлайн-доски Kanban), перемещая заметки или задачи по заранее определенным этапам, чтобы видеть прогресс по мере его достижения и определить, где могут возникнуть препятствия;

– экстремальное программирование (XP). Прежде всего стоит отметить, что данная методология используется только в сфере разработки программного обеспечения, в то время, как Kanban и Scrum можно адаптировать и под другие области. XP Хорошо подходит для продуктов, требования к которым постоянно меняются и дополняются. Среди особенностей данной методологии я бы выделил частые релизы версий, пользовательские тесты, парное программирование, непрерывная интеграция кода, рефакторинг и коллективное владение кодом, то есть каждый может вносить правки в любой сегмент кода;

– метод разработки динамических систем (DSDM). Как и для многих других гибких методологий, для него характерна итеративность, активное вовлечение заказчика в разработку и т.д. В то же время данный метод выделяется тем, что контроль над сроками и бюджетом проекта осуществляются гораздо строже.

На рисунке 1.4 приведено сравнение методологий Agile в разных областях [9]:

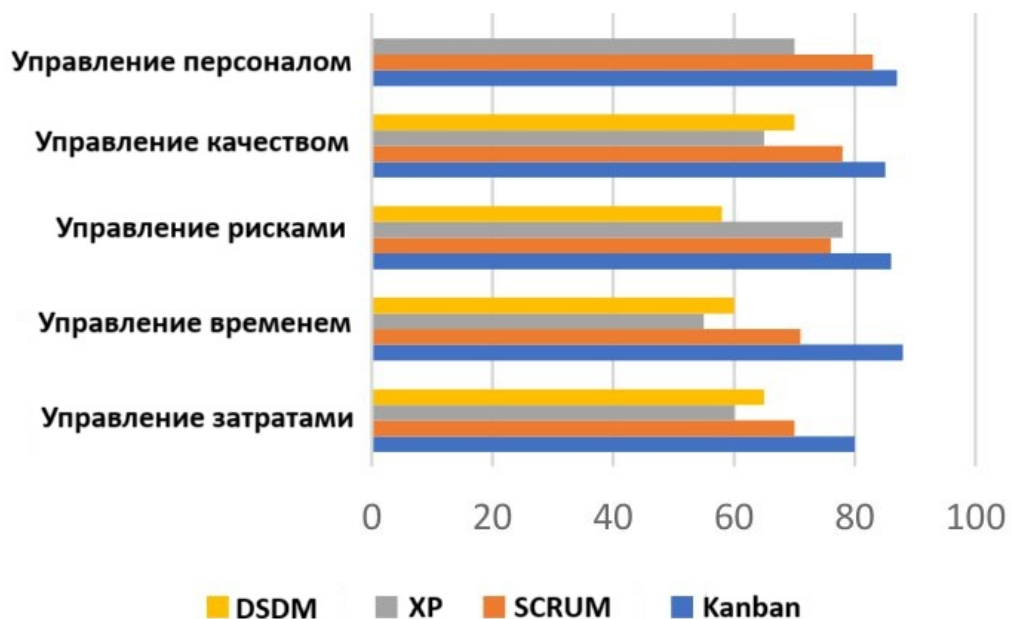


Рисунок 1.4 – Сравнение методологий Agile

Канбан демонстрирует высокие результаты: 80% в управлении затратами (Scrum — 75%, XP — 60%, DSDM — 65%), 88% во времени (Scrum — 85%, XP — 60%, DSDM — 70%), 86% в управлении рисками (Scrum — 84%, XP — 78%, DSDM — 71%), 85% в качестве (Scrum — 83%, XP — 65%, DSDM — 75%) и 87% в HR (как и Scrum, тогда как XP — 70%, DSDM — 75%), подтверждая его лидерство по ключевым критериям.

1.2 Обзор и анализ существующих инструментов управления проектами

1.2.1 Ключевые функции и критерии выбора инструментов

Каждый инструмент управления проектами обладает уникальными особенностями и функциональными возможностями, которые делают его полезным в конкретных ситуациях. Ключевыми возможностями в таких инструментах являются [10,11]:



Рисунок 1.5 – Ключевые возможности инструмента управления проектами

– совместная работа и коммуникация — одна из основных функций, которую должен выполнять инструмент управления проектами – это обеспечение хорошей коммуникации с сотрудниками. Быстрый и эффективный поток коммуникаций позволяет быстрее обмениваться информацией и мнениями между сотрудниками, что повышает эффективность работы. Это может быть реализовано в инструменте в виде чатов, возможности проведения совещаний, работы в командах и т.д.;

– мониторинг выполнения задач — в компаниях, ориентированных на достижение целей, возможность отслеживать прогресс продукта необходима. Таким образом, члены команды могут делиться своими успехами в определенной части проекта, что позволяет участникам иметь лучший обзор развития всего проекта;

- анализ затрат — поскольку проекты регулярно превышают установленную целевую стоимость, необходимо, чтобы инструмент позволял получить представление обо всех затратах. Если менеджеры могут заранее предсказать, сколько будет стоить та или иная часть проекта, мы сможем более точно предсказать целевую общую стоимость;

- планирование — с помощью инструментов управления проектами важно планировать проекты и составлять расписание встреч. Планирование проекта помогает определить ключевые задачи для успешного завершения проекта;

- автоматическая отчетность — дополнительным преимуществом инструмента управления проектами является возможность автоматического создания отчетов, что позволяет менеджерам иметь высокоуровневый обзор состояния проекта. Отчеты также являются ключевой частью успешного управления проектами, поскольку с их помощью можно правильно реагировать на представленную информацию.

Основываясь на существующих исследованиях, я пришёл к выводу, что наиболее важным качеством в управлении проектами является хорошая коммуникация в команде и хорошее отслеживание состояния проекта.

Я разделил функции инструментов на 7 категорий: адаптивность и интеграция, коммуникация и сотрудничество, планирование и проектирование проекта, отслеживание прогресса, управление затратами, формирование отчетов и аналитика, а также средства визуализации.

1.2.2 Сравнительный анализ популярных решений

Для обзора и сравнительного анализа были выбраны следующие инструменты:

- Asana — инструмент, который хорошо подходит для небольших команд, типа агентств, маркетинговых отделов или веб-студий. Среди особенностей данной системы можно выделить гибкую настройку задач, панель мониторинга, которая позволяет в реальном времени отслеживать важные данные по проекту, вехи — ключевые контрольные точки, которые обозначают достижение промежуточного результата и т.д.;

- Trello — популярный планировщик, пользуется спросом у творческих команд и небольших коллективов. Ключевые возможности включают автоматизацию действий с карточками при определенных условиях, шаблоны под самые

разные отделы и цели, интеграцию с почтой и визуализацию данных в "Панели";

- Wrike – ориентирован на большие команды и масштабные проекты. Тут можно выделить элементы искусственного интеллекта, готовые шаблоны проектов, возможность редактировать сразу несколько задач, конструктор отчетов, графики рабочей загрузки и параллельное редактирование документов;

- ClickUp – относительно новый инструмент управления проектами. Поддерживает чат в реальном времени, настройку интерфейса под себя, проектные шаблоны, установку приоритетов к задачам и т.д.;

- Раумо – это облачное ПО для управления проектами и задачами. Подходит небольшим командам. Лучше всего проявляет себя в области управления расходами;

Что касается коммуникации и сотрудничества, то были выделены четыре ключевые функции: обмен файлами между сотрудниками, возможность общения в чатах, отправка/получение и обмен электронными письмами через интегрированный интерфейс, а также возможность упоминать сотрудников. Результаты сравнения отображены на таблице 1.1.:

Таблица 1.1 – Сравнение характеристик коммуникации и сотрудничества

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Обмен файлами	5	5	5	3	5
Чат	5	1	3	1	5
Электронная почта	5	5	5	1	5
Упоминания	5	5	3	5	5
Суммарный балл	20	16	16	10	20

В свойствах планирования и проектирования были выделены три ключевые функции. Они включают в себя добавление, изменение и удаление любых элементов в проекте, возможность назначать важность (критичность) элементов и назначать их взаимосвязь, а также изменять иерархию, вложенность элементов в соответствии с потребностями компании (таблица 1.2).

Таблица 1.2 – Сравнение характеристик планирования

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Управление задачами	5	5	5	5	5
Классификация задач	5	5	5	5	5
Редактирование иерархии	5	5	1	3	3
Суммарный балл	15	15	11	13	13

Далее идет мониторинг отслеживания прогресса. Тут можно выделить управление контрольными точками и вехами проекта, точный учет времени, затраченного на выполнение задач, анализ расхождений между плановыми и фактическими показателями, а также удобный механизм обновления статусов задач (таблица 1.3).

Таблица 1.3 – Сравнение характеристик отслеживания прогресса проекта

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Вехи и цели	5	5	5	5	5
Учет времени	5	5	5	5	5
Обновление статусов	5	5	5	5	5
Оценка прогресса проекта	5	5	5	5	5
Суммарный балл	20	20	20	20	20

Затем следует управление затратами. Это комплексное управление бюджетом, автоматизацию учета расходов, а также инструменты сравнительного анализа плановых и фактических показателей (таблица 1.4).

Таблица 1.4 – Сравнение характеристик управления затратами

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Отслеживание затрат	3	3	3	5	3
Автоматизация учета затрат	3	1	1	5	1
Сравнение с ожидаемыми затратами	5	5	1	5	5
Суммарный балл	11	9	5	15	9

Система отчетности и аналитики должна включать два ключевых компонента: генерацию различных видов отчетов и построение аналитических сводок (таблица 1.5).

Таблица 1.5 – Сравнение характеристик генерации отчетов и аналитики

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Генерация отчетов	5	5	5	3	5
Генерация аналитики	5	5	5	3	5
Суммарный балл	10	10	10	6	10

Также очень важна кастомизация и интеграция (таблица 1.6). Инструменты должны предлагать гибкую настройку под нужды предприятия, а еще интегрироваться с различного рода корпоративными системами.

Таблица 1.6 – Сравнение характеристик настройки и интеграции

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Настройка	5	5	3	3	5
Интеграция	5	5	5	3	5
Суммарный балл	10	10	8	6	10

Ну и наконец, средства визуализации (таблица 1.7). В платных и бесплатных версиях количество этих представлений различается, что и отражено в таблице..

Таблица 1.7 – Сравнение характеристик представления проекта

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Количество представлений	11	9	1	5	6
Платная версия	11	9	6	5	6
Суммарный балл	5	5	1	3	3

В таблице 1.8 представлены итоговые результаты.

Таблица 1.8 – Сравнение инструментов по характеристикам

Характеристика	ClickUp	Wrike	Trello	Paymo	Asana
Коммуникация и сотрудничество	5,0	4,0	4,0	2,5	5,0
Планирование и проектирование проекта	5,0	5,0	3,7	4,3	4,3
Отслеживание прогресса	5,0	5,0	5,0	5,0	5,0
Управление затратами	3,7	3,0	1,7	5,0	3,0
Формирование отчетов и аналитика	5,0	5,0	5,0	3,0	5,0
Адаптивность и интеграция	5,0	5,0	4,0	3,0	5,0
Визуализация проекта	5	5	1	3	3
Суммарный балл	33,7	32,0	24,4	25,5	30,3

Для объективной оценки я проанализировал как комбинированные показатели, так и отдельные ключевые функциональные возможности каждого инструмента. Методика расчета заключалась в делении общего количества баллов по каждой характеристике на количество соответствующих ключевых функций с последующим округлением до одного десятичного знака.

Результаты исследования выявили четкую градацию среди рассматриваемых инструментов. Безусловным лидером стал ClickUp, набравший 33,7 балла из возможных. Незначительно отстает Wrike с 32 баллами, демонстрируя сравнимый уровень функциональности. Третью позицию заняла Asana (30,3 балла), тогда как Paymo и Trello замкнули рейтинг с 25,5 и 24,4 балла соответственно.

По результатам сравнения мы пришли к выводу, что инструменты ClickUp и Wrike наиболее близки к критерию универсального инструмента. Эти инструменты получили 3 и более баллов по всем ключевым функциональным возможностям. Наибольшее преимущество перед остальными они получили за счет количества средств визуализации, возможности кастомизации и интеграции с другими приложениями.

На втором месте в рейтинге оказался инструмент Asana. Он получил хорошие оценки по большинству ключевых функций, но потерял много баллов из-за недостатка средств визуализаций и не самых качественного функционала

управления затратами, а также возможностей кастомизации и интеграции.

За Asana следует Раумо. Хотя Раумо обладает наиболее развитым функционалом управления расходами, большинство других возможностей не дотягивают до уровня других инструментов.

Последним в списке оказался инструмент Trello. Благодаря подключению интеграций инструмент получил хорошие оценки за большинство функций, но недостаток средств визуализаций и возможностей управления расходами снизило баллы за ключевые функции и поставило его на последнее место.

Как мы видим, лучшими инструментами оказались те, которые находили баланс между многообразием функций и удобным представлением, а также те, которые предоставляли широкие возможности для кастомизации и подстройки функционала под нужды предприятия. Мы будем это учитывать при последующем проектировании системы.

1.3 Постановка задачи

Разработать систему управления задачами и рабочим временем сотрудников на предприятии.

Исходные данные:

- предприятие — организация, где работают сотрудники и выполняются проекты;
- руководитель — формирует проект, распределяет задачи между сотрудниками и следит за ходом их выполнения;
- проекты — группы задач, которые имеют определенные цели и сроки выполнения;
- задачи — конкретные действия, требующие выполнения в рамках проекта;
- сотрудники — работники предприятия, которые могут быть назначены на выполнение задач.

Проблемы, которые необходимо решить:

- плохое планирование;
- изменение целей в процессе работы;
- плохая коммуникация в команде;
- неконтролируемые временные затраты;
- низкая прозрачность рабочих процессов.

Для решения указанных проблем необходимо разработать систему, которая обеспечит:

- управление задачами и проектами – создание, назначение, приоритезация и контроль выполнения задач, визуализацию статусов задач, гибкое планирование этапов проекта с учетом сроков и ресурсов;
- учет рабочего времени – автоматический и ручной трекинг времени, затраченного на задачи, анализ загрузки сотрудников и выявление перегрузок/-простоев;
- отчетность и аналитика – генерация отчетов по выполнению задач, проектов и показателей сотрудников, визуализация данных (графики, диаграммы, сводные таблицы).

1.4 Выводы

В данной главе были изучены основные понятия сферы управления проектами, мы проследили путь развития данной области, рассмотрели все этапы жизненного цикла проекта, а также сравнили традиционные и "гибкие" методологии управления. также мы выбрали и проанализировали существующие инструменты управления, выделили ключевой функционал и сделали выводы, что система по управлению задачами и рабочим временем должна находить баланс между многообразием функций и удобным представлением. Наконец, на основе всего этого была сформулирована постановка задачи.

ГЛАВА 2

ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1 Профили пользователей

Профилями называют описания главных категорий пользователей. Одна из таких категорий может быть принята за основной профиль.

Я выделил два профиля: профиль руководителя и профиль сотрудника.

Профиль руководителя:

- социальные характеристики - мужчины/женщины, 30-50 лет, высшее образование, руководящая должность;
- мотивационно-целевая среда - контроль выполнения задач, оптимизация рабочего времени команды;
- навыки и умения - опыт работы с ПК, уверенное использование корпоративных систем;
- требования к системе - делегирование задач, аналитика времени выполнения, гибкие отчеты;
- рабочая среда - офисный ПК/ноутбук, корпоративная сеть.

Профиль сотрудника:

- социальные характеристики - мужчины/женщины, 22-45 лет, высшее/среднее специальное образование;
- мотивационно-целевая среда - фиксация задач и времени, прозрачность в оценке работы;
- навыки и умения - базовые навыки работы с ПК (3+ года), опыт использования таск-менеджеров;
- требования к системе - простое создание/отметка задач, тайм-трекер, приоритизация;
- рабочая среда - офисный ПК/ноутбук, гибридный режим работы.

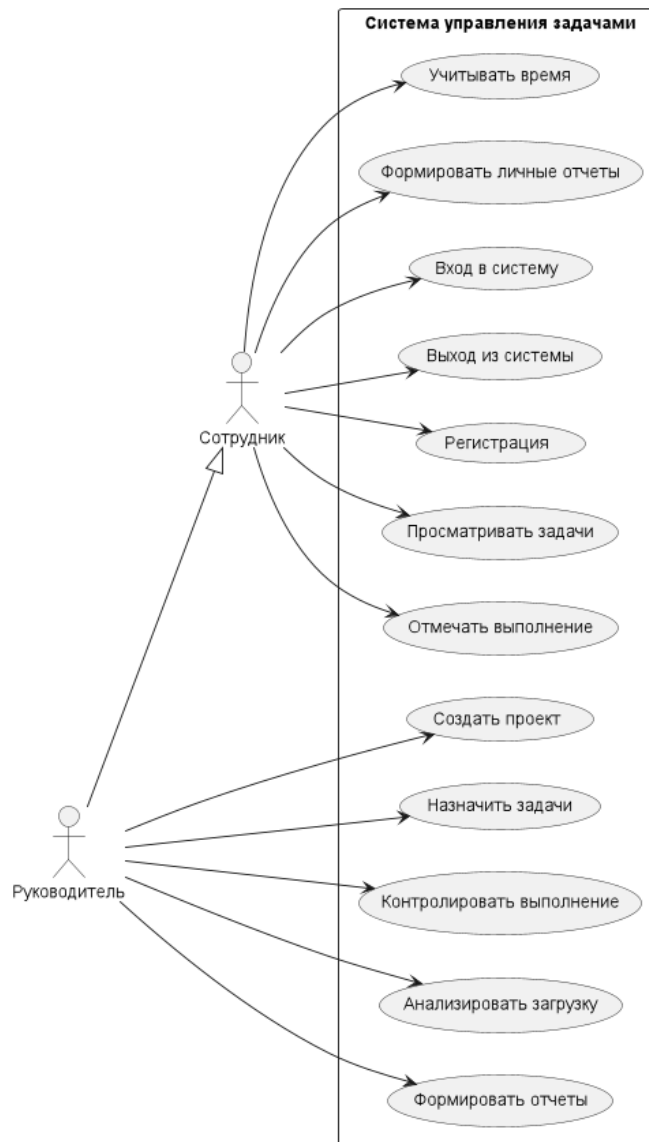


Рисунок 2.1 – Диаграмма вариантов использования системы

На рисунке 2.1 отображены различные варианты использования системы.

2.2 Профиль задач

Профиль задач описывает ключевые действия пользователей в системе управления задачами и рабочим временем.

- создать проект;
- определить цели и сроки проекта;
- просмотр деталей проекта - информация о прогрессе, участниках и связанных задачах;
- назначить задачи сотрудникам;
- просмотр недавней активности пользователя;

- установить приоритеты и дедлайны - определение важности задач и крайних сроков выполнения;
- расширенный поиск задач - фильтрация по статусу, приоритету, сроку или связанному проекту;
- редактирование задачи;
- изменить статус задачи;
- добавить исполнителя из списка сотрудников - выбор ответственного из доступной команды;
- отслеживать статус выполнения задач;
- добавить подзадачу в проект - разбиение крупной задачи на компоненты;
- анализировать загрузку сотрудников;
- формировать отчеты по проекту;
- просматривать список назначенных задач;
- отмечать выполнение задач;
- фиксировать время на выполнение задач;
- регистрация/авторизация;
- добавлять комментарии или заметки к задаче;
- просматривать историю изменений задачи;
- оставлять отзывы о выполненных задачах.

2.3 Профиль среды

Профиль среды определяет условия эксплуатации системы и требования к её адаптивности. Анализ рабочего окружения пользователей позволяет:

- оптимизировать интерфейс под типовые сценарии использования;
- обеспечить стабильную работу на разном оборудовании.

Ключевые факторы среды представлены в таблице 2.1.

Таблица 2.1 – Профиль среды для системы управления задачами и рабочим временем

Характеристика	Признак	Влияние на интерфейс
Место использования	Офисные и удалённые рабочие места с разной скоростью интернета	Настройка количества отображаемых задач на странице, возможность отключения графических элементов
Рабочее место	Разнообразные конфигурации (десктоп/ноутбук)	адаптивный интерфейс под разные разрешения
Освещённость	Разный уровень освещения в рабочих зонах	Высококонтрастный интерфейс, чёткие границы элементов управления
Программное обеспечение	Современные ОС и браузеры	Кроссплатформенная веб-версия, прогрессивное веб-приложение (PWA)
Аппаратное обеспечение	Различная производительность	Отключение сложных анимаций, крупные интерактивные элементы
Прерывания	Частые переключения между задачами	История последних действий, визуальные индикаторы изменений
Шумность	Разный уровень фонового шума	Настраиваемые звуковые уведомления, визуальное дублирование важных событий

Таким образом, система управления задачами должна поддерживать различные устройства и скорость интернета, обеспечивать высокую контрастность интерфейса и оптимизированную производительность.

2.4 Объектная модель

Объектная модель позволяет понять интерфейс через объекты ментальной модели пользователя, систематизация всех объектов продукта.

Основные объекты системы, их атрибуты и возможные действия представлены в таблице 2.2.

Таблица 2.2 – Основные объекты системы управления задачами

Объект	Мощность	Представления	Действия	Атрибуты
Задача	Тысячи	Карточка, список, календарь	Создать, изменить, назначить, завершить, удалить, комментировать, изменить статус	ID, Название, Описание, Срок, Приоритет, Статус, Теги, Автор, Исполнитель, Проект
Проект	Сотни	Дерево задач, доска Kanban, Timeline, календарь	Создать, редактировать, добавлять участников, генерировать отчёты	ID, Название, Описание, Сроки, Статус, Владелец, Участники
Пользователь	Десятки	Профиль, таблица сотрудников	Добавить, редактировать, деактивировать, назначать роли, сбрасывать пароль	ID, Имя, Должность, Email, Роль, Отдел, Аватар, Статус
Рабочее время	Тысячи	Таблица учёта, график загрузки	Зафиксировать, отредактировать, подтвердить	ID, Дата, Время начала, Время окончания, Задача, Статус, Комментарий
Комментарий	Тысячи	Текст, лента обсуждений	Добавить, изменить, удалить, ответить	ID, Текст, Автор, Дата, Задача
Отчёт	Десятки	PDF/Excel	Сформировать, фильтровать по периоду/сотруднику/проекту, экспортировать	ID, Тип, Период, Автор, Данные

Функциональные объекты перечислены в таблице 2.3

Таблица 2.3 – Функциональные объекты системы управления задачами

Объект	Мощность	Представления	Действия	Атрибуты
Поиск	-	Поле ввода с подсказками	Искать задачи по названию, тегам, исполнителю	Фильтры (дата, статус)
Фильтр	-	Чекбоксы, выпадающие списки	Фильтровать задачи по проекту, статусу, приоритету, сроку, тегу	Критерии (исполнитель, проект)
Уведомление	Десятки	Бейдж, рор-уп, email	Отправлять при изменениях, просрочках, новых комментариях	Тип, Получатель, Статус прочтения

Связи между объектами и их структурная организация отражены на рисунке 2.2.

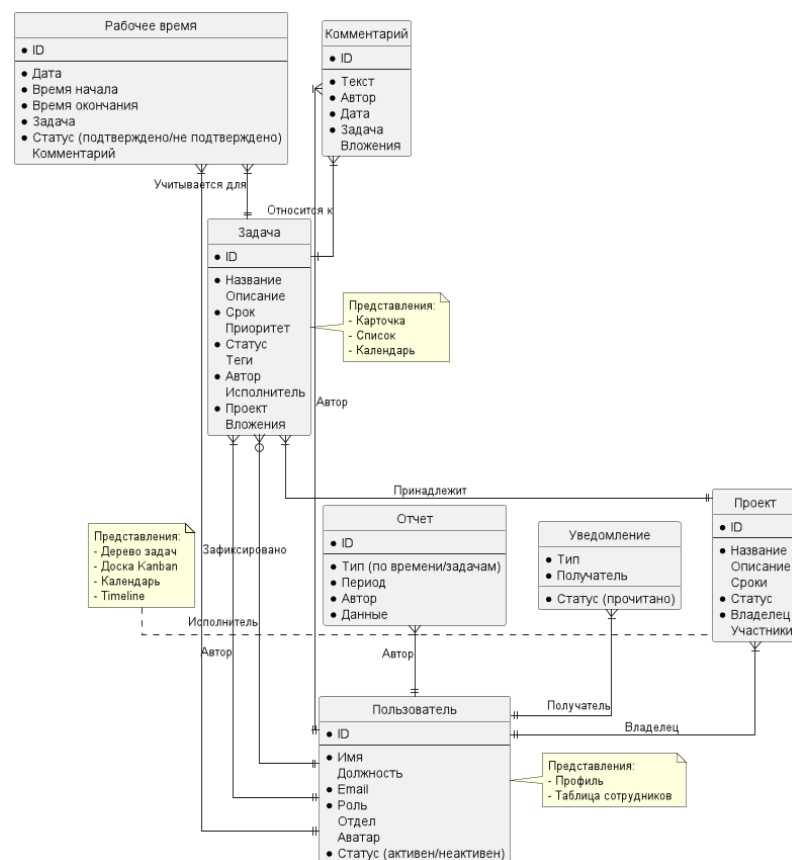


Рисунок 2.2 – связи между объектами и их структурная организация

Данный рисунок визуализирует отношения между сущностями системы.

2.5 Выводы

В данном разделе мы определили основные профили пользователей, которые будут работать с системой. На основе этих профилей был определен основной функционал будущей системы, будь то создание и редактирование отчетов, распределение задач между сотрудниками, анализ загруженности персонала, отслеживание хода выполнения проекта, генерация отчета и т.д. С помощью профиля среды были выделены атрибуты, которые должна соблюдать будущая система, чтобы работать с ней было комфортно в определенных условиях. Также была разработана объектная модель системы, которая пригодится в последующей реализации программного обеспечения.

ГЛАВА 3

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Архитектура системы

В качестве архитектуры системы была выбрана хорошо знакомая "клиент – сервер" которая легко масштабируется а также отлично взаимодействует с различными корпоративными системами.

В качестве стека технологий я остановился на проверенной комбинации Django и PostgreSQL. Выбор обоснован богатым опытом в реализации приложений на основе данных технологий.

Для бэкенда Django хорош вследствие множества факторов:

- основной язык данного фреймворка – python, для которого характерна простота а также расширение функционала за счет установки пакетов;
- функционал Django можно беспрепятственно поделить на различные модули, называемые приложениями, что уменьшает связность и позволяет беспрепятственно работать над разными частями системы;
- в комплекте с фреймворком также идет и административная панель, которая позволяет в удобной форме добавлять, редактировать и удалять объекты без дополнительной разработки.
- встроенная система аутентификации;
- удобная работа с базами данных.

Для фронтенда я выбрал Django шаблоны. В наше время существует множество фреймворков типа Vue.js или React, которые позволяют реализовать сложный функционал, что предпочтительно для полноценного проприетарного приложения, однако целью данной работы не является создание подобного ПО, а реализация прототипа. В таком случае гораздо удобнее выбрать технологию попроще, и шаблоны Django справляются с этой задачей как нельзя лучше.

Для обеспечения гибкости интеграции с другими системами, я реализовал REST API на базе Django REST Framework. Это позволило:

- создать четкую структуру endpoints для всех сущностей системы;
- реализовать стандартные CRUD-операции через API;
- добавить пагинацию, фильтрацию и сортировку для больших наборов

данных;

- автоматически генерировать документацию API.

Как я уже упоминал, В качестве СУБД была выбрана PostgreSQL. Помимо богатого опыта работы с данной системой, в комплекте с ней идет и приложение pgAdmin 4. Оно предоставляет графический интерфейс, который упрощает выполнение различных задач администрирования и разработки.

3.2 Реализация архитектуры

3.2.1 Серверная часть

В процессе разработки серверной части я использовал многочисленные фреймворки и библиотеки python, среди которых можно выделить:

- Django, основной веб-фреймворк для backend-разработки;
- Django REST Framework, для реализации REST API, обработки запросов и сериализации данных;
- psycopg2 – адаптер для работы с PostgreSQL;
- django-dirtyfields – отслеживание изменений полей моделей;
- pyOpenSSL – криптографические функции для защиты данных;
- cryptography – дополнительные средства шифрования;
- Pillow – для работы с изображениями (например, с аватарками);
- django-environ – позволяет удобно управлять настройками с помощью переменных окружения;
- drf-spectacular – нужен для генерации OpenAPI документации;
- reportlab – необходим для генерации PDF-отчетов;
- django-calendarium – модуль календаря для управления сроками.

Функциональность системы разделена на несколько частей (приложений):

- projects – отвечает за управление проектами и связанными с ними задачами. Включает функции создания, редактирования и удаления проектов, распределения задач между участниками, установки сроков и приоритетов. Также отслеживает прогресс выполнения проектов и анализирует загруженность сотрудников;
- users – управляет учётными записями пользователей, их ролями и правами доступа. Реализует функционал регистрации, авторизации, просмотра профили сотрудников, настройки учётной записи;

- dashboard – предоставляет визуализацию данных в виде графиков, диаграмм и сводок. Позволяет отображать статистику по выполненным задачам, загруженности сотрудников, срокам проектов и другой аналитике;
- activity – отслеживает действия пользователей в системе, будь то время работы над задачами, историю изменений, активность в проектах;
- core – нужен для отображения вспомогательных страниц, типа домашней или страницы с помощью по работе с системой

Django идёт в комплекте с административной панелью, которая позволяет в удобной форме добавлять объекты системы, редактировать их и удалять. Особенно это оказалось полезно в процессе тестирования приложения (рисунок 3.1).

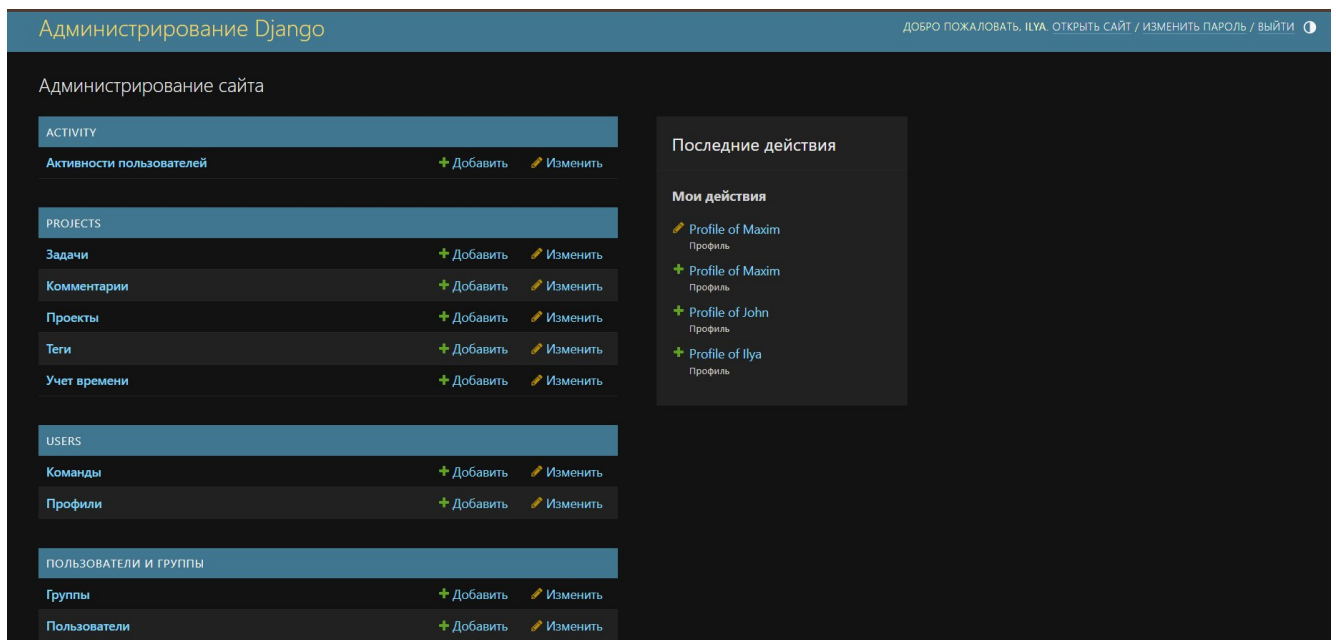


Рисунок 3.1 – Админ-панель Django

Как я уже упоминал, В качестве СУБД я выбрал PostgreSQL. Схему базы данных можно увидеть на рисунке 3.2.

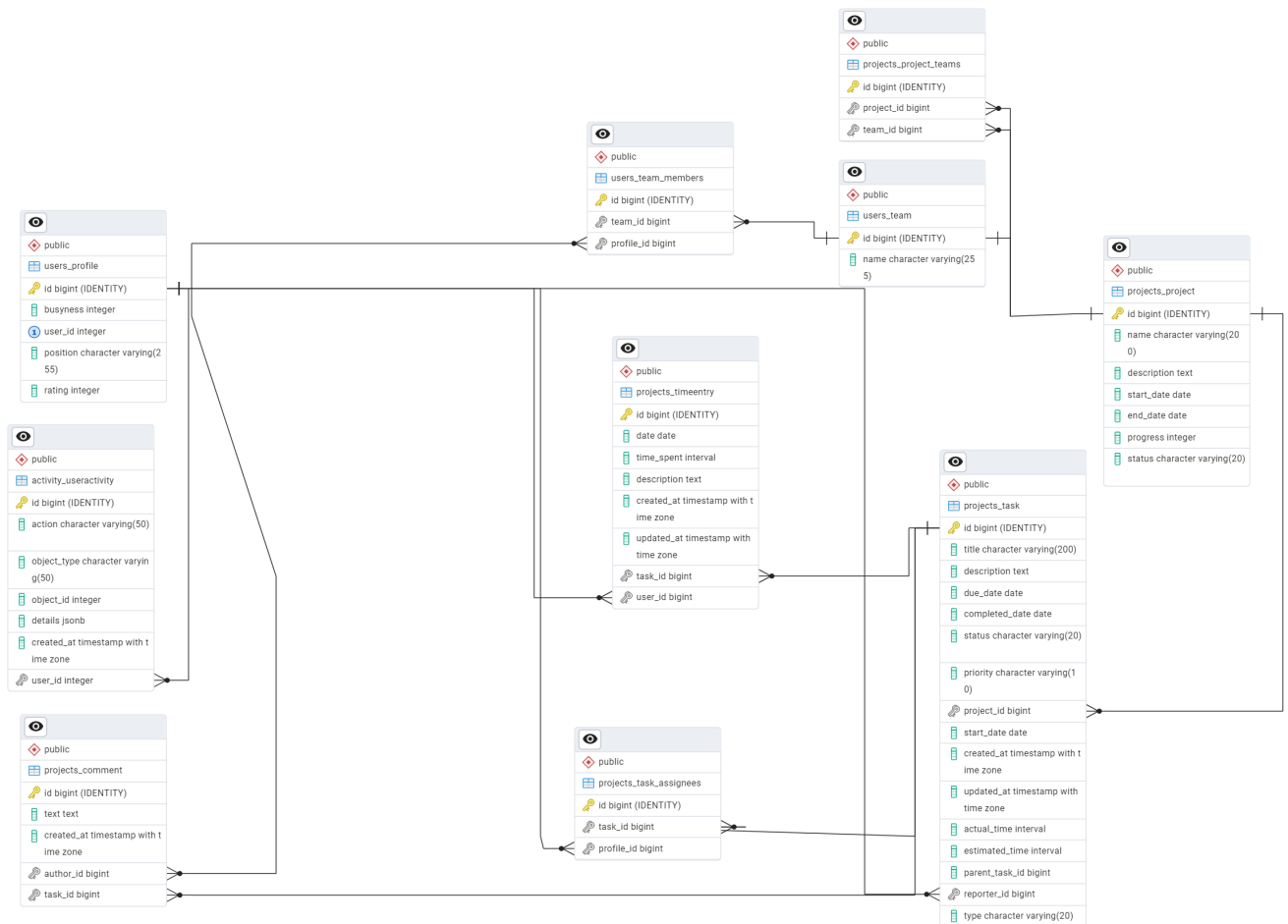


Рисунок 3.2 – Схема базы данных

Основные сущности системы включают:

- Пользователей с различными ролями и правами доступа;
- проекты как основные единицы работы;
- задачи с возможностью декомпозиции на подзадачи;
- учет рабочего времени для контроля производительности;
- комментарии для взаимодействия между участниками.

Стоит отметить, что на схеме выше отображены только самые основные таблицы и связи между ними, необходимые для функционирования системы. Для получения полного представления о системе рекомендуется обратиться к полной схеме базы данных, на которой отображены менее значимые таблицы а также таблицы, идущие в комплекте с Django.

При проектировании RESTful API важно четко определить сущности, их взаимосвязи и соответствующие эндпоинты. В таблице 3.1 представлено соответствие между сущностями предметной области, таблицами в базе данных и маршрутами API.

Таблица 3.1 – Соответствие сущностей, таблиц и эндпоинтов

Сущность	Таблица	Эндпоинт
User	Profile	/users/
Team	Team	/teams/
Project	Project	/projects/
Task	Task	/projects/{project_id}/tasks/
Activity	UserActivity	/activities/
Time Entry	TimeEntry	/projects/tasks/{task_id}/time-entries/
Comment	Comment	/projects/tasks/{task_id}/comments/

В таблице 3.2 детализированы варианты использования API для разных ролей пользователей системы.

Таблица 3.2 – Варианты использования API

Актор	Use Case	Маршрут	Методы	Аутент.
Пользователь	Регистрация, Вход, Выход	/auth/register/, /auth/login/, /auth/logout/	P, P, G/P	Нет/Нет/Да
Пользователь	Домашняя страница	/home/	G	Нет
Пользователь	Помощь	/help/	G	Нет
Пользователь	Профиль	/profile/<uid>/	G	Да
Пользователь	Пользователи	/users/	G	Да
Пользователь	Детали проекта	/projects/<pid>/details/	G	Да
Пользователь	Канбан	/projects/<pid>/kanban/	G	Да
Пользователь	Календарь задач	/projects/<pid>/calendar/	G	Да
Пользователь	Таймлайн	/projects/<pid>/timeline/	G	Да
Руководитель	Команды	/teams/	G,P,PUT,D	Да
Руководитель	Активности	/activities/	G	Да
Руководитель	Дашборд	/dashboard/	G	Да
Руководитель	Проекты	/projects/	G, P, PUT, D	Да
Руководитель	Отчет	/projects/<pid>/report/	G	Да
Сотрудник	Задачи	/projects/<pid>/tasks/	G, P, PUT, D	Да
Сотрудник	Комментарии	.../tasks/<tid>/comments/	G,P, PUT, D	Да
Сотрудник	Учет времени	.../tasks/time-entries/	G, P, PUT, D	Да

В рамках разработки REST API системы был подключен Swagger UI через библиотеку drf-spectacular, которая позволяет автоматически генерировать документацию в формате OpenAPI 3 (рисунок 3.3).

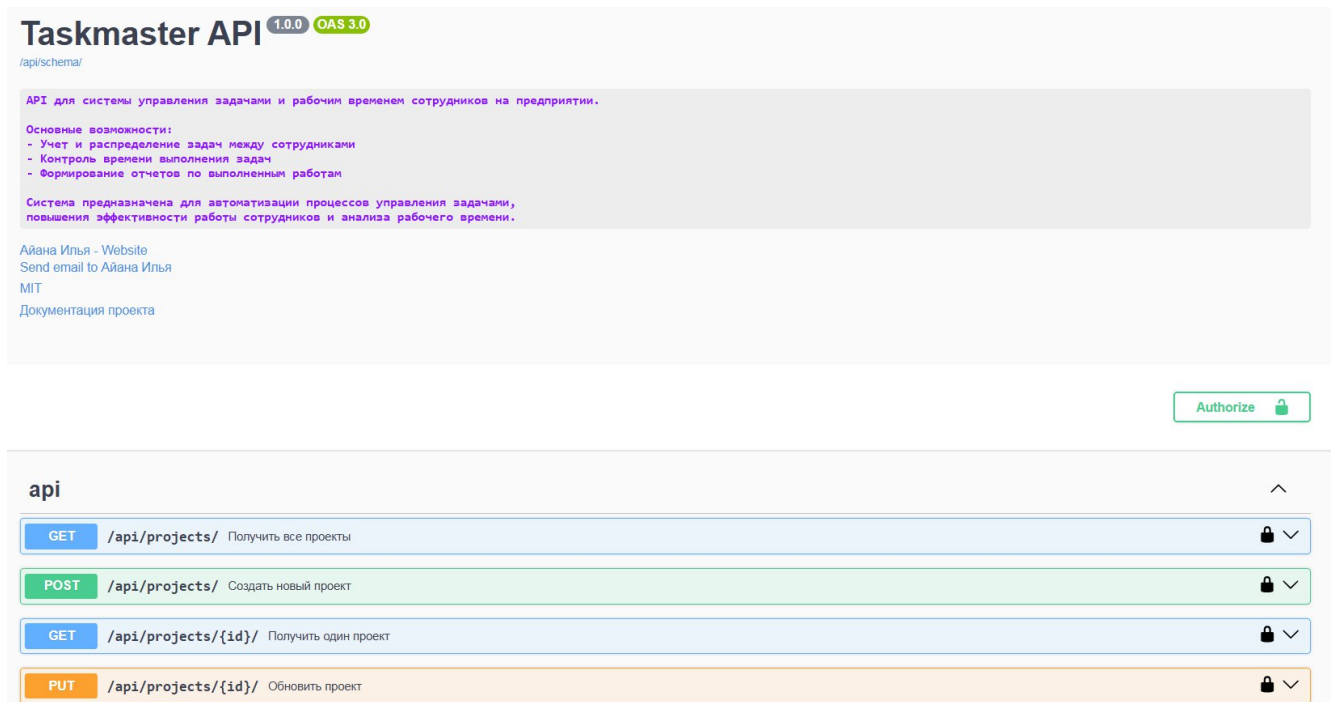


Рисунок 3.3 – Документация Swagger

Документация доступна по маршруту `/api/docs/` и включает:

- описание всех сущностей;
- подробные параметры запросов;
- примеры успешных и ошибочных ответов.

3.2.2 Клиентская часть приложения

Как в начале главы упоминалось, для клиентской части приложения используются шаблоны Django, которые отлично подходят для создания простого, но вместе с тем и полнофункционального прототипа.

Среди используемых технологий можно выделить:

- Django Template Language - для рендеринга страниц на сервере с динамической подгрузкой данных;
- Bootstrap 5.3.2 - как основа CSS-фреймворка для адаптивного дизайна;
- Chart.js 4.4.1 - для визуализации данных в дашборде, личных страницах сотрудников и других страницах.

Интерфейс продемонстрирован на рисунке 3.4

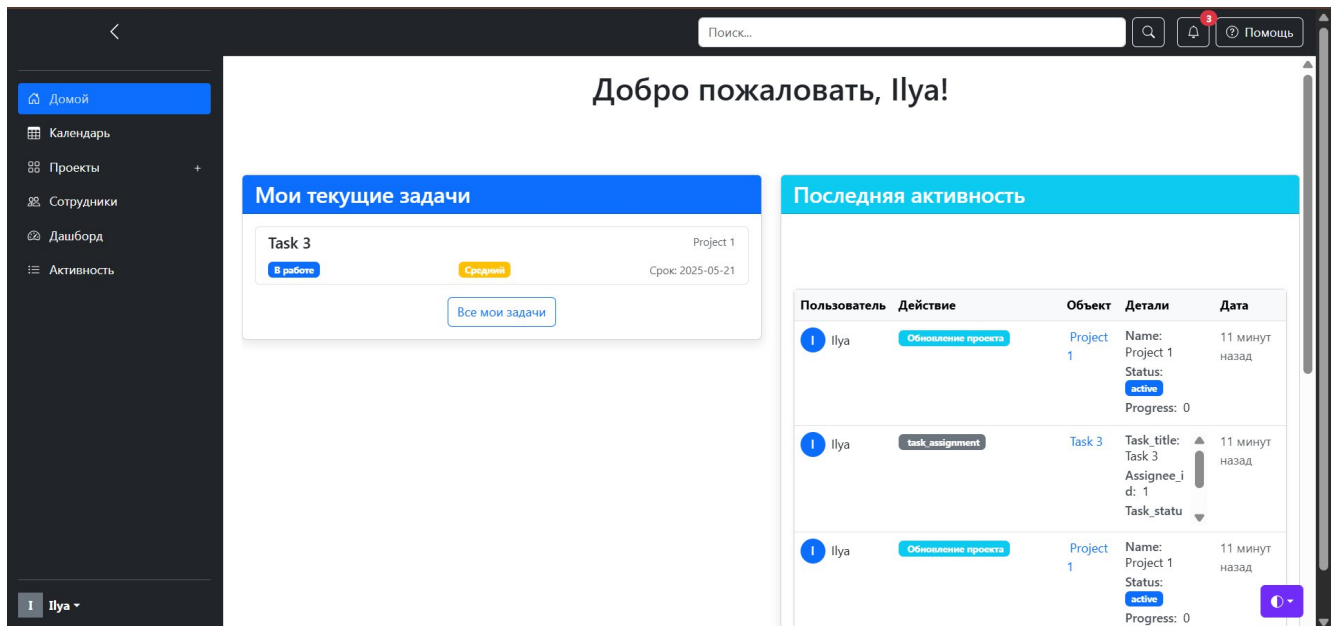


Рисунок 3.4 – Домашняя страница

Для удобного доступа к нужной информации имеется боковая панель, которая позволяет посетить следующие страницы:

- Домашняя страница – самая первая страница, которая встречает пользователя и отображает его статистику;
- Календарь – Позволяет в удобной форме следить за сроками проектов;
- Проекты – позволяет изучить информацию о существующих проектах, изучить задачи, связанные с данным проектом, а также добавить новый проект;
- Сотрудники – показывает список сотрудников. Позволяет посмотреть более подробную информацию о том или ином сотруднике (рисунок 3.5).

Список Сотрудников

Показать 10 записей

Поиск: Что ищете?

Имя Сотрудника	Должность	Рейтинг	Занятость	Команды	Действия
Ilya Ilya@gmail.com	Менеджер	★★★★☆ (3/5)		Не состоит в командах	Профиль ✉
Maxim Maxim@gmail.com	Проектировщик	★★★★☆ (4/5)		Mobile Team	Профиль ✉
Sasha Sasha@gmail.com	Backend-разработчик	★★★★☆ (3/5)	30%	Mobile Team	Профиль ✉
Sergei Sergei@gmail.com	Frontend-разработчик	★★★★★ (5/5)		Mobile Team	Профиль ✉
Vanya Vanya@gmail.com	Тестировщик	★★★★☆ (3/5)		Mobile Team	Профиль ✉

Записи с 1 до 5 из 5 записей

Первая Предыдущая 1 Следующая Последняя

Рисунок 3.5 – Список сотрудников

- Дашборд – предоставляет визуализацию данных в виде графиков, диаграмм и сводок.
- Активность – показывает недавнюю активность пользователей в системе.

3.3 Методика применения системы

Методика применения системы управления задачами и рабочим временем включает следующие этапы:

1. руководитель создает проект и разбивает его на задачи. Для каждого процесса устанавливает сроки выполнения, приоритеты и прочую информацию;
2. руководитель назначает задачи сотрудникам с учетом текущей загрузки;
3. сотрудники выполняют задачи и фиксируют время работы и что за это время было выполнено. Также сотрудники коммуницируют между собой с помощью комментариев на странице задачи;
4. руководитель отслеживает прогресс через календарь, таймлайн, канбан-доску, таблицу задач и дашборд;
5. после завершения проекта руководитель генерирует отчет, в котором отражены выполненные задачи, разница между запланированным и реально затраченным временем на выполнение задачи и прочая информация.

3.4 Выводы

В данной главе было успешно реализовано программное обеспечение системы управления задачами и рабочим временем сотрудников на предприятии, которое удовлетворяет всем базовым требованиям. Также была разработана методика применения данной системы, который позволит руководителям использовать приложение максимально эффективно.

ГЛАВА 4

ПРИМЕР ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

4.1 Постановка задачи

Имеется некоторая фирма "А" которая занимается разработкой программного обеспечения. Однажды ей поступает новый заказ: создать высококачественное и интуитивно понятное мобильное приложение для фитнеса, которое будет удовлетворять потребности пользователей в отслеживании их физической активности и питания. Приложение должно быть разработано до 10 июня 2025 года.

4.2 Решение на основе методики

В первую очередь руководитель создает новый проект(рисунок 4.1):

Создание нового проекта [← Назад к списку](#)

Название проекта*

Мобильное приложение для Фитнеса

Описание

Мобильное приложение для фитнеса

Дата начала*

05/13/2025

Дата окончания

06/18/2025

Оставьте пустым, если срок неизвестен

Команды

☐ Mobile Team

Теги проекта

[Сбросить](#) [+ Создать проект](#)

Рисунок 4.1 – Создание проекта

Далее руководитель разбивает проект на несколько задач и распределяет их между сотрудниками:

- Проектирование UI/UX;
- разработка бэкенда;
- Реализация экрана тренировки и статистики.
- Тестирование;

Руководитель и сотрудники отслеживают выполнение задач с помощью различных форм визуализации:

- Таблицы для получения наиболее полного впечатления о текущем состоянии дел (рисунок 4.2)

The screenshot shows a web application for managing project tasks. On the left is a dark sidebar with navigation links: Домой, Календарь, Проекты, Сотрудники, Дашборд, and Активность. The main area is titled 'Задачи проекта' and includes a search bar, filter dropdowns for Status, Priority, Task Type, and Assignee, and buttons to 'Apply' or 'Reset' filters. Below the filters is a table of tasks with columns for Type, Name, Description, Start Date, End Date, Status, Priority, Assignee, Progress, and Subtasks. The table contains four tasks: 'Backend Development' (in progress), 'Screen Implementation' (backlog), 'Application Testing' (backlog), and 'UI/UX Design' (completed). At the bottom, there are pagination controls showing '1' of 4 records and buttons for 'Previous', 'Next', and 'New Task'.

Тип	Название	Описание	Начало	Срок	Статус	Приоритет	Исполнители	Прогресс	Подзадачи
Задача	Разработка бэкенда	Разработка бэкенда	2025-05-14	2025-05-21	В работе	Средний	Sasha		Нет
Задача	Реализация экрана тренировки и статистики	Реализация экрана тренировки ...	2025-05-22	2025-05-29	Бэклог	Средний	Sergei		Нет
Задача	Тестирование приложения	Тестирование приложения	2025-05-30	2025-06-06	Бэклог	Средний	Valya		Нет
Задача	Проектирование UI/UX	Проектирование UI/UX	2025-05-09	2025-05-13	Выполнено	Средний	Maxim	100%	Нет

Рисунок 4.2 – Таблица задач проекта

- Таймлайна (рисунок 4.3)

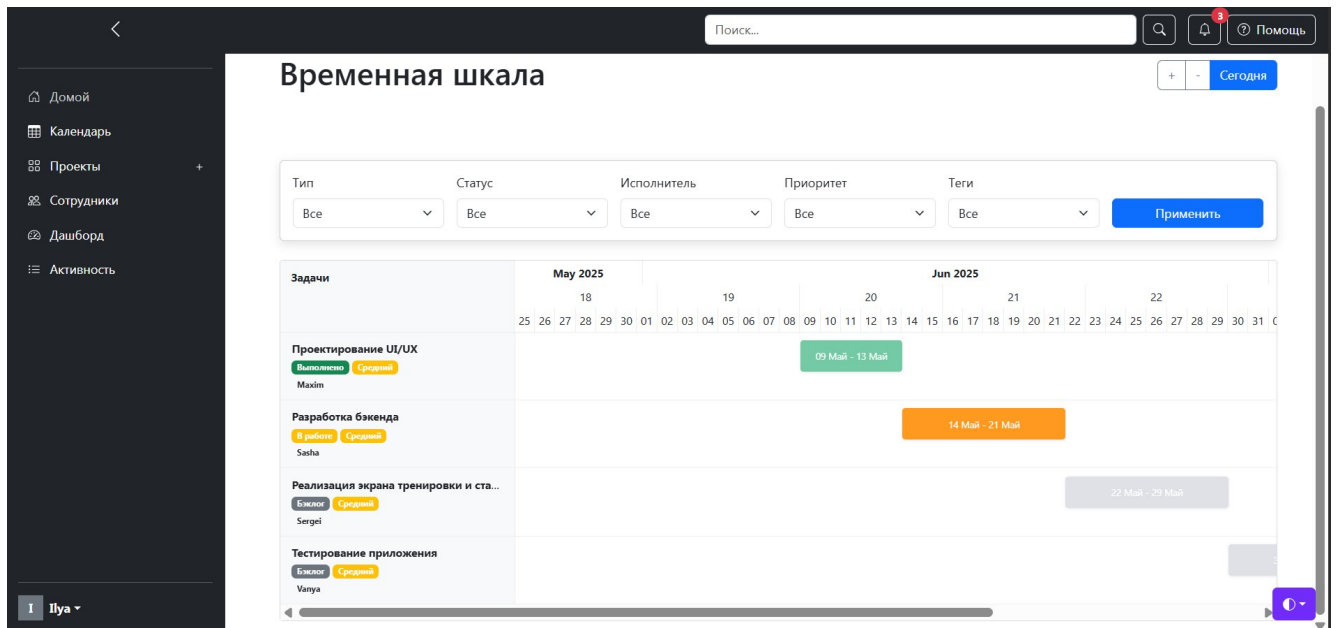


Рисунок 4.3 – Таймлайн проекта

– Календаря (рисунок 4.4)

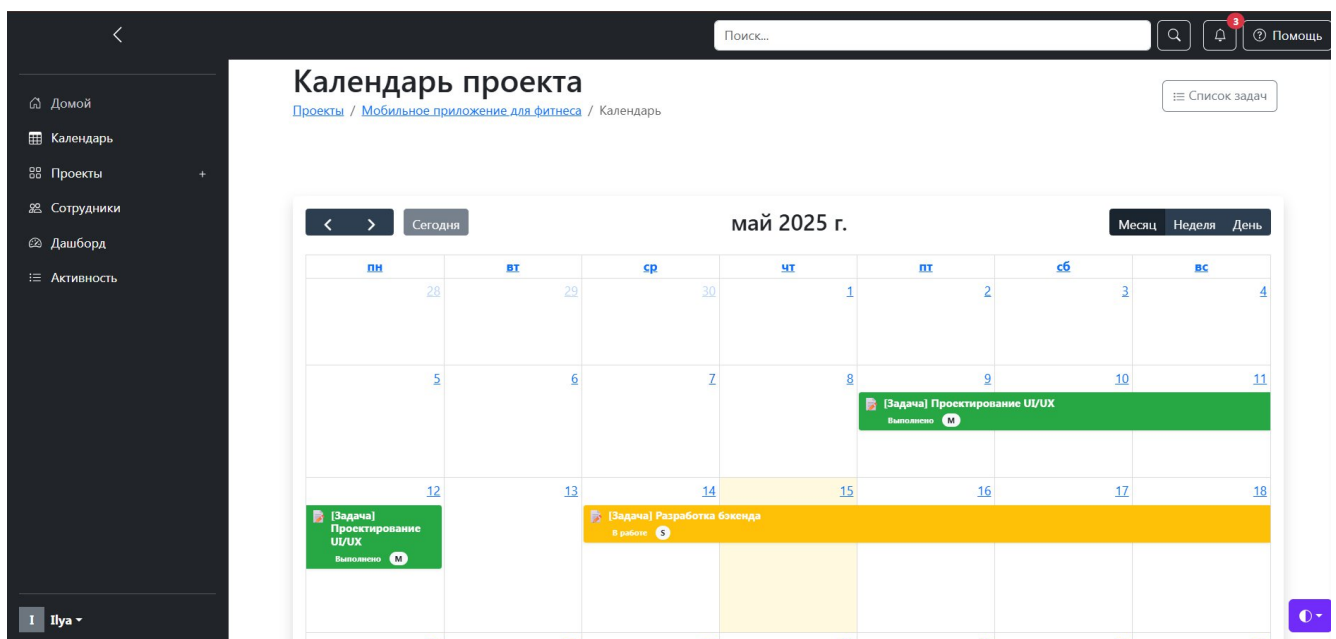


Рисунок 4.4 – Календарь проекта

– Канбан-доски (рисунок 4.5)

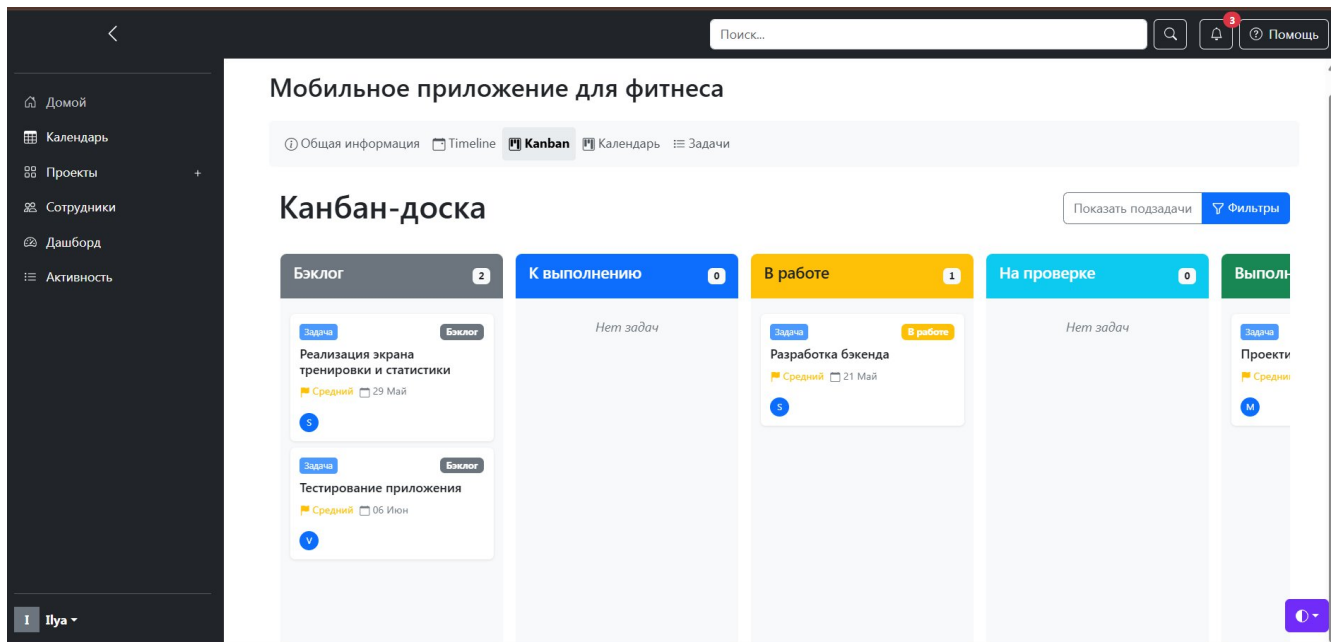


Рисунок 4.5 – Канбан-доска проекта

Сотрудники выполняют поставленные задачи, вместе с тем вводя данные о проведенной работе (рисунок 4.6).

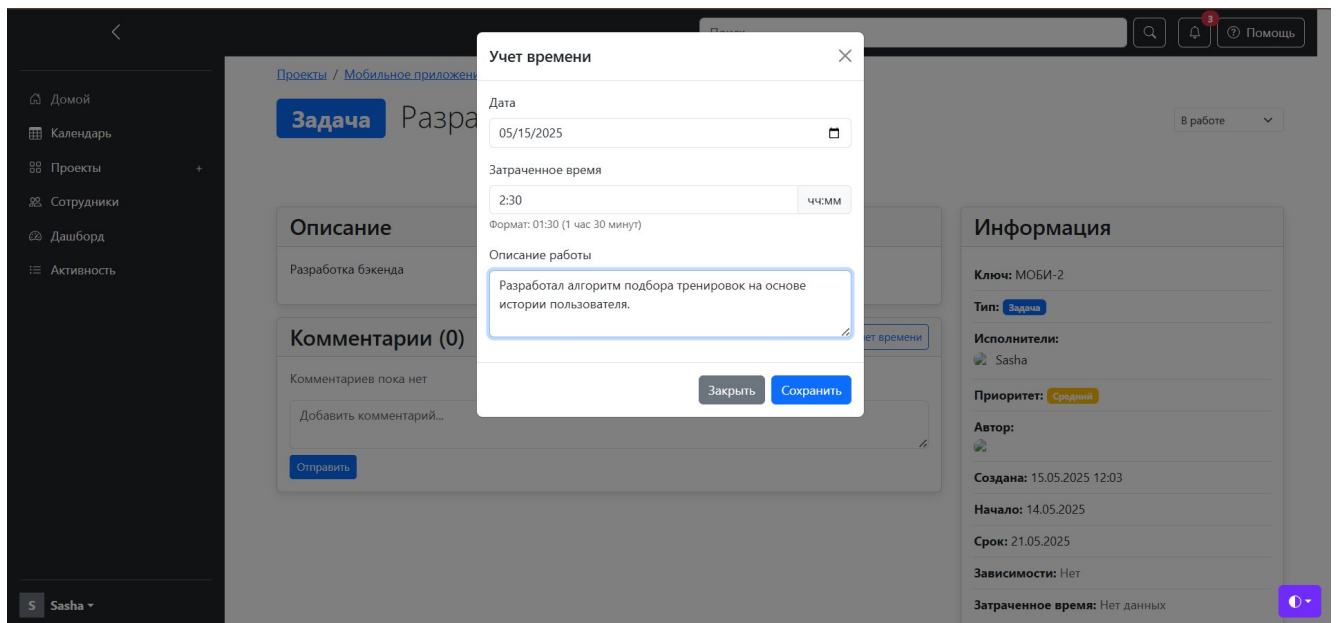


Рисунок 4.6 – Учет времени

Спустя пару недель руководитель решает проверить состояние проекта. Как видно на рисунке 4.7, проект успешно завершен.

Проекты

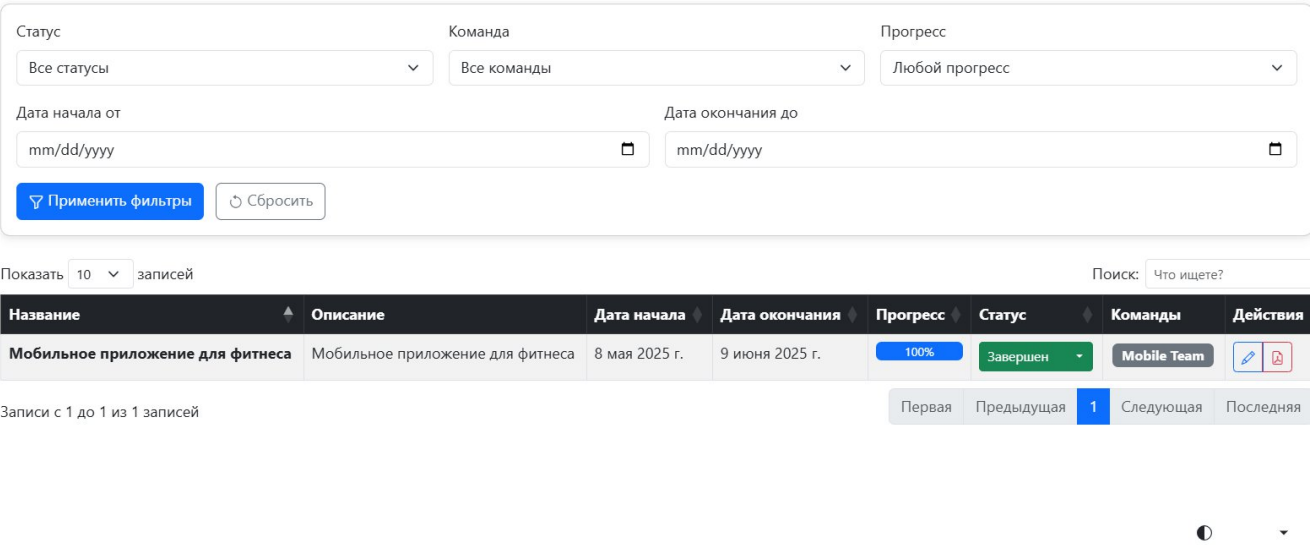


Рисунок 4.7 – Список проектов

Руководитель изучает отчет по проекту для проведения дальнейших действий. Частичный отчет представлен в приложении А.

4.3 Выводы

В представленном примере продемонстрирована эффективность и правильность внедрения системы управления задачами и временем сотрудников. Стоит отметить, что реализованный функционал системы по сути является базовым, поэтому перспективы развития системы широкие. Например, можно внедрить интеграцию с различного рода корпоративными системами или добавить функции или добавить поддержку других гибких методологий, например Scrum.

ЗАКЛЮЧЕНИЕ

Дипломная работа посвящена проектированию и разработке системы управления задачи и рабочим временем сотрудников на предприятии. Были изучены основы управления проектами, а также рассмотрены существующие методологии, инструменты и их ключевой функционал. На основе данного анализа была реализована система, которая хорошо справляется со следующими проблемами:

- управление проектами и задачами – система предоставляет широкий функционал для создания проектов, разбиения на задачи, распределение их между сотрудниками и удобное отслеживание их выполнения;
- учет рабочего времени – был реализован инструментарий для фиксации времени, которое сотрудники потратили на выполнение задачи;
- аналитика и отчетность – система предоставляет возможность генерации отчета по выполненным задачам и проектам, на основе которых руководители могут принимать грамотные управленческие решения.

Основные результаты данной работы:

1. проведен анализ современных методологий и инструментов, выявлены их преимущества и недостатки. На основе данного анализа были сформулированы основные возможности и особенности будущей системы;
2. выполнено проектирование системы с разработкой основных сущностей системы;
3. были успешно реализованы составляющие системы;
4. разработана методика по эффективной работе с инструментом.

Практическая значимость работы подтверждена примером, представленным в главе 4. Система успешно выполнила поставленные задачи на всех этапах жизненного цикла проекта, от его создания до завершения и генерации отчета.

Перспективы развития данной системы включают расширение функционала, а также интеграцию с корпоративными системами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Камиллери, Э. Project Success: Critical Factors and Behaviours / Э. Камиллери. — Gower Publishing, 2011. — 303 с.
2. Робертс, Дж. П. Successful IS Project Management / Дж. П. Робертс, Дж. Фурлонгер. — Gartner, Inc., 2000.
3. PMBOK 6 версия. — Project Management Institute, 2017. — 756 с.
4. What is the project management triangle and how can it help your team? [Электронный ресурс] / Team Asana. — Режим доступа: <https://asana.com/resources/project-management-triangle>. — Дата публикации: 15.02.2025. — Дата доступа: 26.04.2026.
5. Фроуд, Э. The Rise of Project Management Software: A History [Электронный ресурс] / Э. Фроуд. — Режим доступа: <https://thedigitalprojectmanager.com/topics/history-of-project-management-software/>. — Дата доступа: 25.04.2025.
6. Хатун, Т. Understanding the Project Management Cycle [Электронный ресурс] / Т. Хатун. — Режим доступа: <https://www.researchgate.net/publication/381254980>. — Дата доступа: 27.04.2025.
7. Альбрехт, Й. GIS Project Management [Электронный ресурс] / Й. Альбрехт. — Дата публикации: декабрь 2017. — Дата доступа: 28.04.2025.
8. Методологии управления проектами [Электронный ресурс]. — Режим доступа: <https://practicum.yandex.ru/blog/metodologii-upravleniya-proektami/>. — Дата доступа: 14.04.2025.
9. Ахмед, И. The Influence of Kanban Agile Methodology on Software Project Management: A Survey [Электронный ресурс] / И. Ахмед. — Режим доступа: <https://www.researchgate.net/publication/389803226>. — Дата доступа: 29.04.2025.
10. Кендрик, Т. The Project Management Tool Kit: 100 Tips and Techniques for Getting the Job Done Right / Т. Кендрик. — 2013.
11. Чичибаш, Х. A Comparison of Project Management Software Tools (PMST) / Х. Чичибаш, О. Унал, К. Демир. — 2010. — Режим доступа: <https://www.researchgate.net/publication/221610417>. — Дата доступа: 17.04.2025.

ПРИЛОЖЕНИЕ А
Отчет по проекту (частичный)

Отчет по проекту: Мобильное приложение для фитнеса
Основная информация о проекте

Название	Мобильное приложение для фитнеса
Описание	Мобильное приложение для фитнеса
Статус	Активный
Прогресс	100%
Дата начала	2025-05-08
Дата окончания	2025-06-09
Команды	Mobile Team
Теги	Нет тегов

Статистика по задачам

Всего задач	0
Завершено	4
В работе	0
Просрочено	0
Средний прогресс	100%

Список задач проекта

ID	Название	Тип	Статус	Приорите т	Оценка времени	Прогресс	Срок
1	Проектирование UI/UX	Задача	Выполнено	Средний	7ч 0м	100%	2025-05-13
2	Разработка бэкенда	Задача	Выполнено	Средний	10ч 0м	100%	2025-05-21
3	Реализация экрана тренировки и статистики	Задача	Выполнено	Средний	11ч 0м	100%	2025-05-29
4	Тестирование приложения	Задача	Выполнено	Средний	5ч 0м	100%	2025-06-06

Детализация по задачам

Задача #1: Проектирование UI/UX

Тип	Задача
Статус	Выполнено
Приоритет	Средний
Оценка времени	7ч 0м
Прогресс	100%
Дата создания	2025-05-15T12:02:22.754819Z
Срок выполнения	2025-05-13
Дата завершения	2025-05-12
Исполнители	Maxim

Учет времени

Дата	Пользователь	Затраченное время	Описание
2025-05-11	Maxim	3h 30m	Подготовка интерактивного прототипа для демонстрации
2025-05-10	Maxim	3h 0m	Разработка дизайн-системы (цвета, шрифты, компоненты)

Общее затраченное время: 6ч 30м

Оценка времени: 7ч 0м

Экономия времени: 0ч 30м