Geekbrains

**Разработка приложения для управления командами, проектами и задачами**

Курс «Программист Python. Цифровые профессии»

Дюльберова Диана Альбертовна

Пермь

2023

Оглавление

[Введение 3](#_Toc153474164)

[Глава 1. Проектирование веб-приложения 5](#_Toc153474165)

[1.1 Анализ аналогичных проектов 5](#_Toc153474166)

[1.2 Технологии и инструменты 17](#_Toc153474167)

[Глава 2. Реализация веб-приложения 23](#_Toc153474168)

[2.1 Постановка задачи и требований к приложению. 23](#_Toc153474169)

[2.2 Описание приложения 24](#_Toc153474170)

[2.3 Проектирование интерфейса 29](#_Toc153474171)

[Глава 3. Тестирование веб-приложения 31](#_Toc153474172)

[Заключение 35](#_Toc153474173)

[Список используемой литературы 38](#_Toc153474174)

[Приложения 39](#_Toc153474175)

[Приложение 1. Результаты анализа аналогичных проектов. 39](#_Toc153474176)

[Приложение 2. Структура базы данных проекта. 40](#_Toc153474177)

[Приложение 3. Макеты реализации интерфейса приложения. 41](#_Toc153474178)

[Приложение 4. Код веб-приложения 42](#_Toc153474179)

[4.1 manage.py 42](#_Toc153474180)

[4.2 requirements.txt 43](#_Toc153474181)

[4.3 taskmanager/asgi.py 44](#_Toc153474182)

[4.4 taskmanager/settings.py 45](#_Toc153474183)

[4.5 taskmanager/urls.py 51](#_Toc153474184)

[4.6 taskmanager/wsgi.py 52](#_Toc153474185)

[Приложение 5. Ссылка на приложение 53](#_Toc153474186)

# Введение

Для того чтобы стать успешным приложением для управления проектами, оно должно удовлетворять разнообразные запросы своих пользователей. Приложение должно обеспечивать эффективное управление задачами и ресурсами, при этом минимизируя информационную перегрузку и помогая в налаживании контактов между членами команды. Кроме того, оно должно иметь возможность быть настроенным, масштабируемым и поддерживаться с помощью надежной системы поддержки для исправления ошибок и обновлений.

В данной работе основной целью является разработка веб-приложения, которое будет использоваться для управления задачами в процессе командной разработки.

Главной задачей, которая поставлена перед проектом, является следующая: разработать веб-приложение, позволяющее осуществлять управление задачами (формировать / изменять / удалять).

В приложении реализованы следующие возможности:

• С помощью функции управления можно управлять статусом задачи.

• Разделение заданий по проектам дает возможность выполнять различные действия с ними: разрабатывать, изменять или удалять их.

• В программе есть возможность разделения задач, которые находятся в разных проектах, на спринты (а также, изменять или удалять их).

• Чтобы понять, какие именно задачи были решены, достаточно посмотреть историю изменений, чтобы увидеть их решение.

• Предоставляет возможность предупреждать пользователей о необходимости изменения текущего состояния задачи.

• Возможность авторизации через определенный уровень доступа предоставляется пользователям.

Основополагающими требованиями к проекту являются следующие:

• Порядок авторизации должен быть соблюден.

• Юнит-тесты должны быть пройдены.

• Важно использовать сериализаторы drf, а также rest api.

• Реализовать логгирование.

• Пользователи должны иметь возможность пользоваться графическим представлением, а также оно должно быть функциональным и обладать достаточной функциональностью для управления приложением.

Данное веб-приложение позволяет сделать работу более легкой и удобной, делает ее понятной и прозрачной, а также способствует распределению заданий между сотрудниками, при этом отслеживая выполнение каждого из них. В связи с тем, что в настоящее время наблюдается тенденция к удаленной работе, необходимость в данном устройстве возросла.

В реализованном дипломном проекте установленные требования к системе были учтены. Также, на базе указанных выше требований были созданы макеты страниц приложения и схема базы данных.

# Глава 1. Проектирование веб-приложения

## Анализ аналогичных проектов

Этот раздел посвящен анализу аналогичных проектов. На данный момент мы должны определить основные требования к приложению, а также рассмотреть преимущества и недостатки аналогичных проектов. Многие системы управления проектами предлагают бесплатную версию, но существуют два принципиально разных типа.

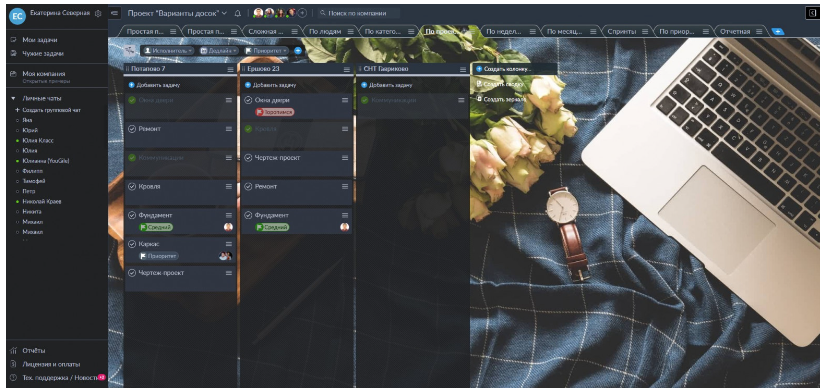
Первый тип - бесплатная версия. С помощью таких систем разработчики выражают искреннюю заботу о том, чтобы вы могли использовать продукт в любое время, не испытывая ограничений по времени, и рекомендовали его своим друзьям.

Вторая разновидность – это так называемая "пробная версия". В таких продуктах намеренно отключаются те функции, которые необходимы для полноценного использования, так как они предназначены для того, чтобы превратить вас в платного клиента. Это может стать для многих пользователей неприятным сюрпризом, когда они только начнут пользоваться системой.

В этом разделе можно ознакомиться с обзорами 10 бесплатных систем управления проектами.

1. YouGile

В этой системе можно бесплатно получить облачную версию для 10 человек. Это означает то, что данная версия является "чистой", в ней есть все функции, которые необходимы. Оплата будет произведена только после 11-го пользователя.



Используя бесплатный вариант приложения YouGile, вы сможете:

- Определить состав сотрудников в системе, обозначив их должности, а также функционал и направления деятельности.

- Чтобы начать сотрудничество с подрядчиками и фрилансерами, необходимо настроить доступ к системе для них.

- Создавать проекты, доски и задания.

- Для того, чтобы упорядочить свои и других пользователей, можно воспользоваться личным планировщиком.

- В процессе работы необходимо устанавливать связи между различными отделами. Так, например, заказы, которые были переданы отделом продаж в производство, могут быть использованы.

- С помощью месседжера можно вести диалог. Для решения каждой задачи используется чат или же возможно личное общение, в котором используются смайлики, цитаты и другие элементы, которые есть в мессенджерах.

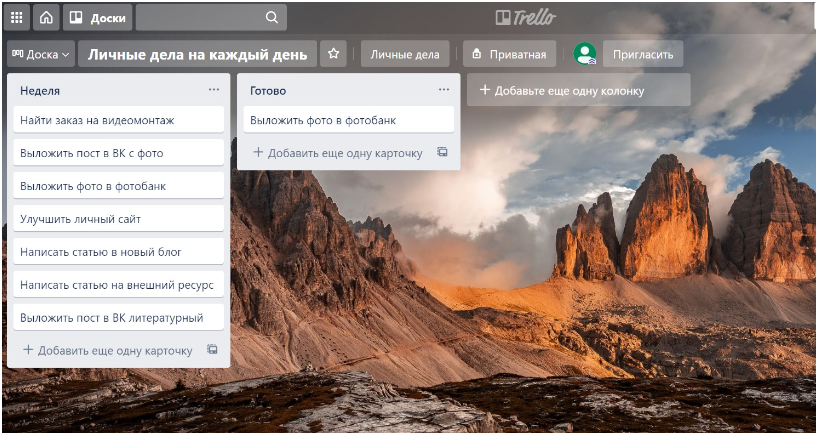
- В распоряжении находятся все виды отчетов и сводок, которые могут быть использованы для проверки качества выполняемой работы.

- Возможности для создания своей рабочей среды: собственная доска для записей, выбор фона и отслеживание рабочего времени.

Если использовать бесплатный тариф YouGile, можно легко организовать работу небольшой команды или отдела. Всего можно создать столько проектов и досок для рисования, сколько необходимо. Также можно установить права доступа для сотрудников, назначить задания и отслеживать их выполнение. Делая акцент на важных задачах, можно отметить их стикерами или же добавить в список "Избранное". Быть в курсе всех событий, происходящих в команде. Обсуждать с коллегами новые идеи и проекты. Делиться файлами, не обращая внимания на их размер.

2. Trello

В качестве альтернативы Trello предлагает бесплатную версию, которая может быть охарактеризована как честная и продуктивная. Несмотря на это, она не накладывает ограничений на время и количество участников. Однако существует ограничение в отношении количества досок - их можно создать до 10 штук. Помимо этого, некоторые функции имеют некоторые ограничения: они не могут назначить права доступа, отсортировать карточки и экспортировать данные, а также использовать более одного расширения для одной доски. В то время, когда компания Atlassian приобрела Trello, он был еще более совершенным. В то же время, несмотря на это, мы все же считаем эту версию достаточно успешной для работы с небольшими коллективами.



Пользователи могут рассчитывать на то, что в бесплатной версии этой системы управления проектами им будет предоставлен весь спектр возможностей, которые они ожидают от данного инструмента. Это дает возможность не только создавать доски для выполнения заданий, но и добавлять задачи, а также устанавливать дедлайны. У вас есть возможность создавать и редактировать множество карточек и списков, а также приглашать столько людей, сколько вам будет угодно.

В вашем распоряжении есть возможность скачивать файлы, но при этом существует ограничение по размеру - 10 МБ. Данная функция является очень важной, так как она поможет вам организовать рабочие материалы и документацию.

Для того, чтобы увеличить количество общения в процессе выполнения задач, стоит воспользоваться интеграцией со Slack. Это дает возможность командам обмениваться информацией более эффективно, а также отслеживать прогресс в работе.

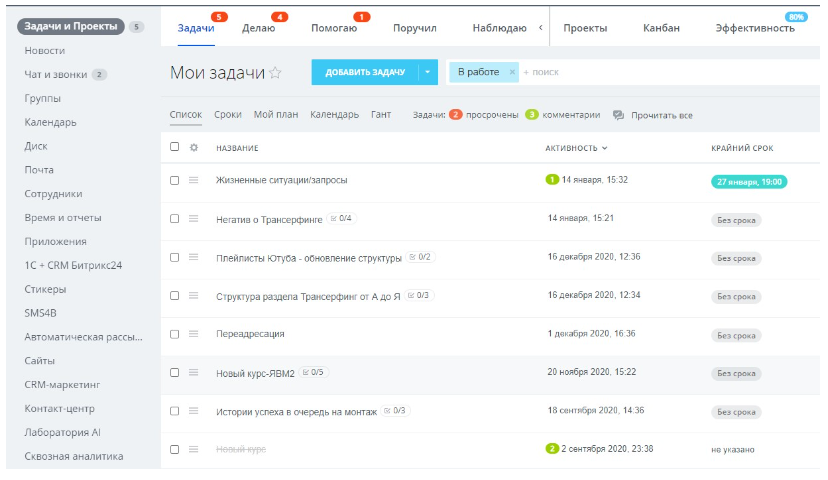
Для использования в личных целях, также существует этот инструмент. При желании вы можете создать план действий на день, неделю или месяц и отслеживать его выполнение. У вас всегда будет возможность контролировать свои задачи и проекты.

Также система имеет ряд преимуществ, которые делают ее удобной для организации работы в небольшой группе или отделе. В вашем распоряжении будет возможность легко распределять задачи между сотрудниками, следить за их выполнением и контролировать выполнение проектов.

С помощью бесплатных версий Trello можно воспользоваться множеством полезных возможностей, которые повысят эффективность вашего труда и сделают его более эффективным.

3. Bitrix24

Bitrix24 – это новый формат платформы, в которой уже нет ограничений на количество пользователей. Теперь, при желании, каждый может пригласить на платформу неограниченное количество коллег. Это стало возможным благодаря тому, что раньше это было возможно только в случае ограничения числа участников до 12 человек. Функционал, который является базовым для Bitrix24, позволяет использовать все необходимые инструменты для того, чтобы эффективно выполнять проекты.



При помощи Bitrix24 можно объединить в одно рабочее пространство даже самые большие команды. При общении с другими людьми, используя живую ленту или чаты, можно обмениваться видео и аудио материалами, а также передавать файлы.

Еще в Bitrix24 можно создать группы, проекты и создавать задачи на доске и ставить задачи и подзадачи, назначая исполнителей для каждой из них. Это дает возможность рационально организовать работу, а также контролировать процесс выполнения работы.

Благодаря платформе Bitrix24 появилась возможность просмотра различных отчетов, что дает возможность проводить анализ деятельности команды и принимать обоснованные решения.

В Bitrix24 можно без проблем сохранять и получать доступ к файлам из любой точки мира, благодаря использованию 5 ГБ облачного хранилища.

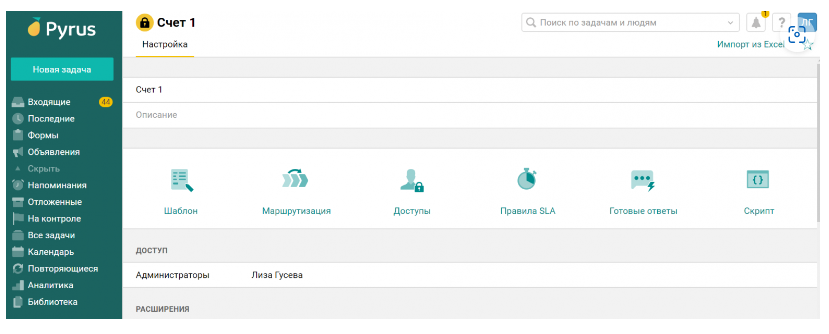
Помимо всего прочего, Bitrix24 помогает экономить время. Он позволяет интегрироваться с такими популярными облачными сервисами как Google Drive и Dropbox, Яндекс Диск и OneDrive, а также предоставляет возможность экономного использования времени. Это даёт возможность работы с файлами, которые могут быть размещены не только на Bitrix24, но и на других облачных платформах.

Благодаря технологии редактирования документов в режиме онлайн, Bitrix24 дает возможность редактировать документы, используя GoogleDocs и MS Office Online, что позволяет повысить гибкость и удобство работы с текстами и документами.

В конечном счете, благодаря всем этим возможностям и функциям, Bitrix24 в бесплатной версии предоставляет все необходимое для продуктивной работы команды над проектами.

4. Pyrus

У системы, специализированной для работы с документами, есть ограничения по количеству задач, которые можно выполнить на основе предоставленных форм. Она может создать до 100 заданий за один свободный тариф. Важно понимать, что если ваш бизнес-процессуальный процесс требует больше задач, то в скором времени вам придется оплатить дополнительный платный тариф. Наиболее удобным вариантом для пользователей, которые хотят использовать доски и стандартные задачи, является бесплатная версия.



Какие действия можно совершить в бесплатной версии?

- Можно приглашать сотрудников в любое количество и создавать задачи, которые требуют минимального количества исполнителей.

- Объединять задачи, используя их взаимосвязь, и группировать их по времени и активности.

- Можно обмениваться файлами, которые имеют отношение к задачам. Благодаря этому, у вас появится возможность использовать 1 ГБ свободного места для хранения файлов.

- Просто, удобно и быстро решать задачи с помощью досок или персонального планировщика. Планировщик позволяет отображать задачи, которые вы поставили перед собой и которые будут выполнены. Есть также разделы, предназначенные для активных и завершенных задач.

- Функции, которые доступны в карточках задач, включают в себя: контроль и отложенный выход, а также настройку напоминаний и повторение задачи.

- У нас есть возможность создания и настройки четкой структуры компании, а также разграничения прав доступа для разных подразделений с помощью соответствующих возможностей.

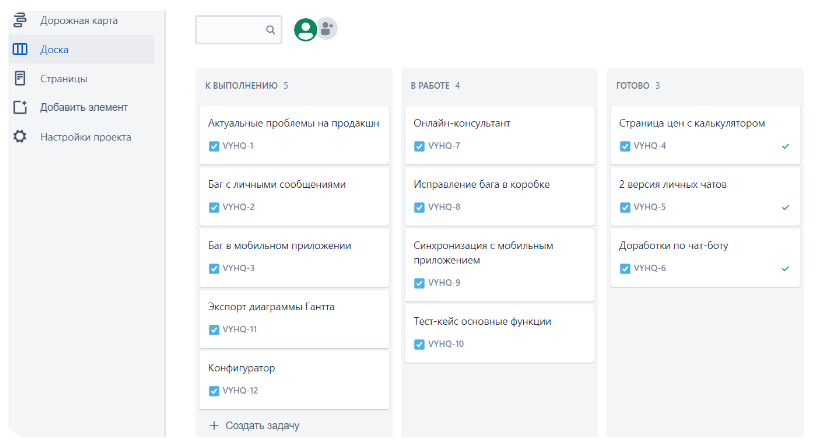
- Установлено, что каждый сотрудник может отслеживать свое время, которое он затратил на выполнение заданий за определенный период времени. Подробные отчёты и аналитика доступны на сайте.

- В число возможных вариантов интеграции входят такие службы хранения данных, как Google Drive, Dropbox, Box, OneDrive, а также системы управления взаимоотношениями с клиентами (CRM), которые поддерживают интеграцию с данными.

- В некоторых случаях возможно проведение автоматического документооборота. С помощью этой системы можно проводить согласование платежей и договоров, а также оформление заявок и работу с запросами клиентов.

5. Jira

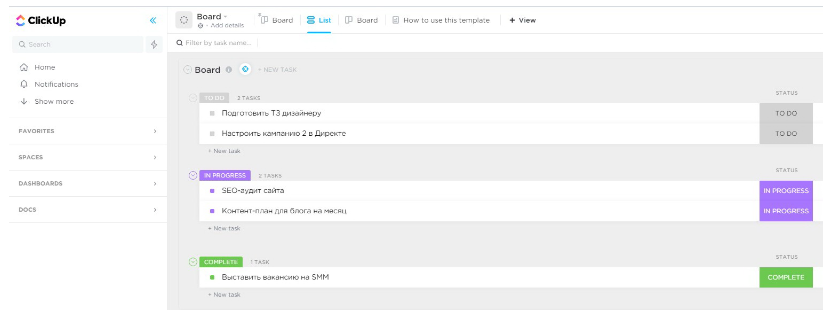
По словам представителей компании, Jira предоставляет бесплатную версию для до 10 пользователей, но она имеет ограничения в 2 ГБ места и не позволяет использовать расширенные права доступа.



С помощью этой версии можно проводить планирование работы, использовать шаблоны проектов, объединять папки с рабочими материалами в одном месте и распределять приоритеты между задачами. Также можно проводить обсуждение целей и задач коллектива, а также получать поддержку от Jira.

6. ClickUp

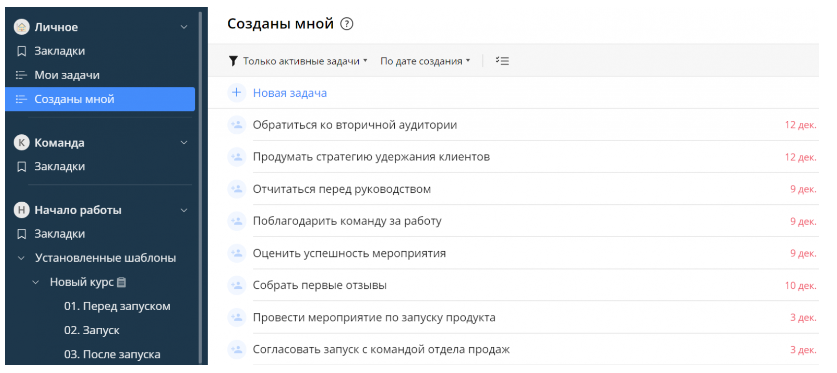
В свою очередь, ClickUp имеет ограничения на выполнение некоторых функций в размере 100 действий.



В бесплатной версии есть возможность приглашать неограниченное количество пользователей для создания досок и задач, использования блокнотов, напоминаний и календарей, проведения анализа производительности спринтов, отслеживания времени выполнения заданий и многое другое. Также доступна круглосуточная техническая поддержка.

7. Wrike

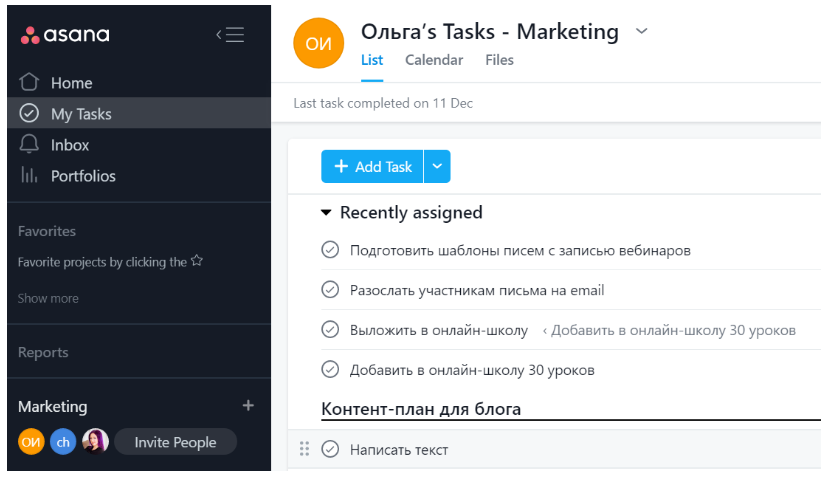
Компания Wrike, предоставляющая доступ к бесплатной версии для до 5 пользователей с возможностью установки дополнительных прав доступа и оценки загруженности сотрудников, имеет ограничения на настройку параметров доступа и оценку загрузки сотрудников.



На бесплатной версии можно создавать проекты и доски, назначать задачи и исполнителей, организовывать файлы и папки, обмениваться информацией с коллегами, а также использовать другие платформы для взаимодействия с системой.

8. Asana

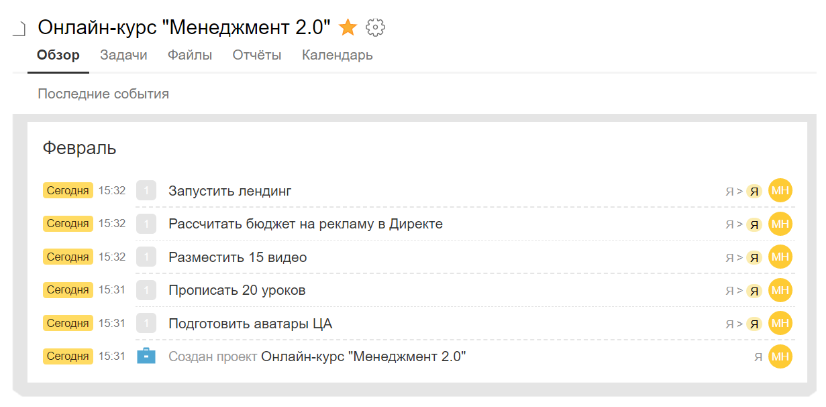
В Asana, для которой доступно до 15 пользователей, есть ограничения на отчеты и контроль выполнения заданий, а также отслеживание обновлений.



Бесплатная версия предоставляет возможность создавать проекты, доски и задачи, назначать исполнителей, присваивать им квалификационные категории и использовать теги для классификации задач. Кроме того, в ней можно общаться с помощью карточек задач.

9. Worksection

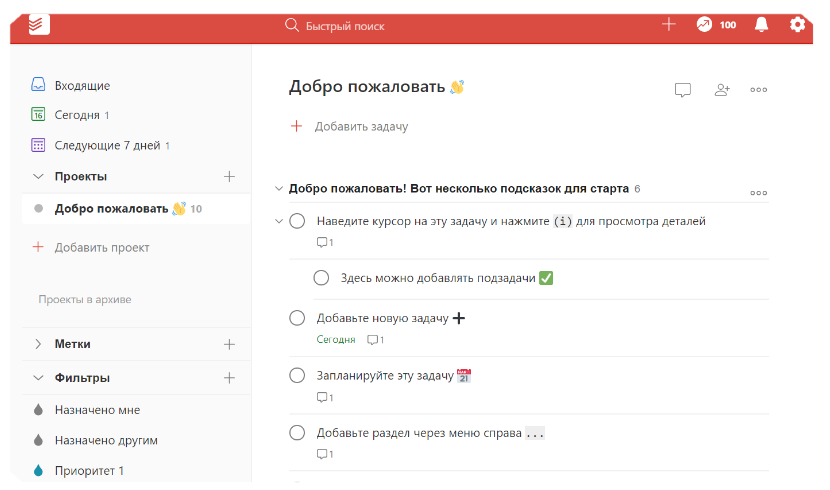
Worksection предоставляет возможность бесплатного использования до 5 пользователей, но при этом она имеет ограничения в 2 проектах и 100 МБ места.



С помощью данной версии возможно осуществлять создание проектов, привлечение пользователей, создание задач и обсуждение их. Также можно присоединять файлы и создавать проекты.

10. ToDoist

В бесплатной версии ToDoist можно использовать до пяти пользователей и 80 проектов, она ограничивает количество задач до 150, а также не разрешает добавлять комментарии или скачивать файлы. Не смотря на это, в бесплатной версии есть возможность размещения 150 задач и ознакомления с системой.



На выходе каждая из этих платформ имеет свои ограничения и возможности, которые она может предоставить в своих бесплатных версиях. Выбор зависит от потребностей и требований команды. Но ни одна из них не может учесть все нюансы, которые касаются индивидуальных потребностей бизнеса.

Оценочные результаты анализа приведены в Приложении 1.

## Технологии и инструменты

Этот раздел посвящен анализу выбранных для реализации технологий и инструментов. Технологии были выбраны, основываясь на их функциональных характеристиках.

Существует множество возможностей, которые имеют место быть в настоящее время, и их использование имеет актуальность.

Наиболее подходящими для реализации оказались следующие технологии:

**Python** - является высокопроизводительным языком программирования общего назначения, в котором динамическая строгая классификация и автоматическое управление памятью используются для повышения производительности программиста, качества кода и его доступности, а также для обеспечения возможности переноса программ, которые были написаны на нем. Язык обладает абсолютной объектно-ориентированностью, в результате чего все предметы являются объектами. Наиболее интересным свойством этого языка является выделение блоков кода с помощью отступов. За счёт минимализма в языке, практика редко сталкивается с необходимостью обращения к документам. К тому же, данный язык широко известен как интерпретируемый, который используется в том числе для создания скриптов. Среди недостатков языка можно отметить его низкую скорость работы и более высокую потребляемость памяти программ, которые написаны на нем. В то время как код, написанный на других языках, таких как C или С++, имеет гораздо более высокие показатели производительности и потребления памяти.

**Фреймворк Django 4.1** - фреймворк, который используется для создания веб-приложений. Создание сложных и многофункциональных веб-приложений с большим количеством функций требует использования большого количества готовых модулей, надстроек и инструментов. Джанго предлагает разработчикам большой выбор таких модулей, надставок и инструментов, которые значительно ускоряют и упрощают этот процесс.

Благодаря своим "батарейкам" фреймворк имеет возможность создавать различные модули и надстройки, которые существенно облегчают процесс работы. Django включает в себя множество разнообразных модулей: это и панель управления, которая отвечает за администрирование, аутентификацию и авторизацию, формы с автоматической валидацией данных, сервер, который поддерживает автоматическую обработку данных. Помимо этого, фреймворк обладает рядом других преимуществ:

* Универсальность - это то, что отличает данный фреймворк. Django имеет множество дополнительных модулей, позволяющих реализовать на платформе любое приложение с любой функциональностью и сферой применения.
* Оборудование для работы с базами данных, которое имеет удобную прослойку в виде ORM, позволяет разработчику не прибегать к написанию запросов на языке SQL.
* Поиск оптимальных путей повышения производительности труда. Настраивая Django-приложение на максимальную производительность, разработчики использовали встроенные механизмы кэширования и обработки статических файлов, а также оптимизировали запросы.
* Демонстрационная легкость в интеграции. Интеграция фреймворка возможна со всем: от блога на WordPress до GraphQL. Данный фронт для Django приложения может быть реализован на любом JavaScript-фреймворке, при этом передача данных с бэкенда на фронт и обратно будет осуществлена без использования API. Среди множества высоконагруженных платформ, на которых используются Python и Django, некоторые из них используют совместно с другими языками программирования.
* Сжатие. Для того чтобы создавать фуллстек-приложения, которые используют фреймворк Django, необходимо использовать его средства для вывода данных и осуществления всех действий с ними на фронтенде. Существует множество вариантов создания гибридных программ, в которых обработка данных разделена между некоторыми модулями Django и JavaScript фронтендом. Можно использовать Django (и Djangos REST Framework) только API и бэкенд, предоставляя фронтенду полную свободу действий.
* Правила безопасности. У фреймворка есть надёжная встроенная защита от использования SQL-кода, подделки межсайтового запроса и межсайттового скриптинга.
* Повышение скорости процесса разработки. В связи с большим количеством готовых модулей, процесс разработки приложений на данном фреймворке занимает значительно меньше времени, чем при использовании других программных платформ. Это происходит за счет того, что в распоряжении разработчиков имеется высококачественная документация и большое количество готовых модулей.

Django имеет некоторые недостатки.

* Имеет место жесткая архитектура. Однако, на начальном этапе изучения структуры Django-проекта может показаться сложным и не подверженным изменениям, что характерно для Flask. В то же время, жесткая архитектура имеет свои преимущества. Она обеспечивает четкую и понятную структуру проекта, а также простоту в использовании всех приложений. Например, в случае создания модуля блога или форума его можно будет легко перенести в любой другой проект, связанный с Django.
* Нехватка производительности для приложений, которые имеют высокую степень нагрузки. У фреймворка есть свои пределы по оптимизации производительности, о которых мы говорили ранее. С помощью Django невозможно создать большой маркетплейс, аналогичный Amazon. Фреймворк в «чистом» виде не подходит для использования в более масштабных приложениях, так как для них лучше использовать бэкенд на Django и API на Djanjo REST Framework, а также использовать их совместно с фронтендом на React или Angular. В то же время, Django идеально подходит для большинства веб-приложений, которые имеют умеренную нагрузку.
* Принцип работы, при котором происходит синхронное выполнение всех действий. Каждый запрос, отправленный в Django, блокирует выполнение других запросов до того момента, как он будет завершен. При этом могут возникнуть проблемы, связанные с обработкой большого количества одновременных запросов. Чтобы разработать приложения для обработки огромного количества запросов и получения ответов, которые имеют место быть в реальном времени, необходимо использовать асинхронные решения на основе веб-сокетов, такие как Django Channels. Чтобы уменьшить требования к приложениям, которые все же должны принимать запросы от большого количества пользователей и отвечать на них в процессе подготовки результатов на бэкенде, можно воспользоваться асинхронными представлениями asyncio. Благодаря Asyncio появилась возможность создания асинхронного кода, который может выполнять параллельные операции с несколькими запросами и выполнять длительные операции без ограничения других операций.

Как устроен процесс работы Django.

При разработке этого фреймворка был использован архитектурный принцип MVT (Model-View-Template, модель - представление - шаблон). Основываясь на данном подходе, код программы состоит из данных (моделей), логики и пользовательского интерфейса.

Структура хранения данных в базе зависит от модели (Model). Django использует ORM (объект-реляционное моделирование, или объектно-ориентированное представление) для создания виртуальной объектной базы данных. При этом разработчики могут использовать стандартные объекты и методы, а также другие способы для взаимодействия с БД. Это позволит им обходиться без использования SQL-запросов в процессе работы с БД.

На основе запроса пользователя формируется ответ, который затем передается в представление (View). Оно обрабатывает данные из модели, после чего передает их в шаблон (template), чтобы отображать их.

Шаблон (Template) - это часть представления, которая определяет способ отображения информации на веб-странице. Django является одним из разработчиков шаблонов, использующих свой собственный язык программирования. Его язык очень похож на Python. Во время загрузки страницы HTML, данный язык позволяет выполнять любые действия с данными.

**База данных Postgres.** В основе базы данных Postgres лежит язык SQL, который поддерживает многие из возможностей стандарта SQL:2011 и некоторые из возможностей SQL :2016 в части работы с данными в формате JSON.

**Django REST Framework.** В состав Django REST входит новый модуль - Django. Software Django REST Framework позволяет взаимодействовать любому устройству через API с сайтом на сервере, используя интерфейс соединения. DRF, по сути, является инструментом создания API для сайта, который позволяет удаленно взаимодействовать с ним.

**HTML.** Во всемирной паутине используется стандартизированный HTML-код для разметки документов. Основная часть веб-страниц содержит информацию о методах разметки на английском языке.

Веб-страницы содержат описание разметки на языке HTML или XHTML. Далее, язык HTML подвергается интерпретации браузерами. По итогу интерпретации форматированный текст отображается на экране компьютера, а также мобильного устройства.

**CSS** является формальным языком, который описывает внешний вид документа, который был создан с использованием языка разметки.

**Итоги главы**

Произведен анализ аналогичных проектов, в ходе которого были выявлены основные функции, которые необходимы для данного приложения, и на основе этого можно сделать вывод о том, какие именно технологии необходимы для реализации данного проекта. Также был проведен анализ способов, которые помогут реализовать данное приложение.

# Глава 2. Реализация веб-приложения

## 2.1 Постановка задачи и требований к приложению.

Когда в проекте задействовано много людей, становится сложно контролировать процесс и следить за тем, чтобы все задачи выполнялись качественно и в срок.

Популярный таск-менеджер, или Task Tracker, является специализированной программой, которая предназначена для управления задачами и проектами. Она значительно облегчает работу, делает их понятными и прозрачными, способствует распределению заданий между исполнителями и отслеживанию их выполнения. В настоящее время, в связи с активным развитием удаленной работы, необходимость в данном устройстве возросла.

При помощи таск-менеджера есть возможность контролировать множество разных задач, которые одновременно выполняют. Увеличить производительность каждого сотрудника и команды в целом можно с помощью этого метода. Сейчас в IT-сфере нет необходимости в найме таск-менеджера, так как они уже давно перестали быть частью этой сферы. На данный момент эти удобные инструменты широко используются в различных стартапах, крупных корпорациях и обычных пользователях, которые стараются рационально использовать свое время.

Разработка веб-приложения для управления задачами в процессе командной разработки является основной целью данного проекта.

Основные задачи, которые поставлены перед проектом:

Данный проект включает в себя разработку веб-приложения, позволяющего управлять задачами (создавать / изменять / удалять).

В приложении можно:

• Управлять статусом задачи с помощью функции управления:

• Разделять задачи по проектам, позволяет выполнять различные действия с ними: создавать, изменять или удалять их.

• Позволяет разделять задачи, которые находятся в разных проектах, на спринты (а также, изменять или удалять их).

• Возможность просматривать историю изменений, чтобы понять, какие именно задачи были решены.

• Позволяет предупреждать пользователей о необходимости изменения текущего статуса задачи.

• Пользователи могут авторизоваться, используя определенный уровень доступа.

К проекту предъявляются следующие требования:

* Наличие процедуры авторизации;
* Проведение юнит-тестов;
* Следует использовать сериализаторы drf, а также rest api;
* Наличие логгирование.
* Графическое представление должно быть доступно для пользователей, и оно должно быть достаточно функциональным, чтобы управлять приложением.

## 2.2 Описание приложения

Две основные составляющие таск-менеджера - проект и задача - объединены в три категории: план, спринт и задача. Принт является частью проекта. Эта задача может быть как частью проекта, так и отдельным спринтом.

У всех пользователей есть возможность просматривать информацию, находящуюся в интерфейсе, и получать данные с помощью API. Авторизация пользователей, имеющих право создания новых записей в определённой категории, позволяет им создавать новые записи на сайте. Запись может быть изменена или удалена автором. Дата завершения задания может быть изменена исполнителем.

Во время внесения изменений в проект, спринт или задачу, происходит запись информации в истории зависимых не закрытых задач. Также, для всех авторов и исполнителей зависимых не закрытых задач и спринтов будет создан e-mail.

В интерфейсе, при отображении списка, отображается список из пяти записей. Всего в API насчитывается десять записей.

В момент открытия страницы сайта или выбора в меню пункта "Главная" отображается список задач, которые не были выполнены.

**Проекты**

Когда выбирается пункт меню "Проекты", в списке отображается список тех проектов, которые не были завершены. Нажав кнопки «Список», «Выполнение», «В работе», можно отсортировать проекты по статусу. В случае необходимости, авторизованные пользователи могут воспользоваться кнопками "Автор" или "Без автора", чтобы добавить проект в список. У проектов, которые имеют разные права, будут отображаться кнопки "Добавить" и "Редактировать", а также кнопка "Удалить" в зависимости от прав в проекте.

На странице детальной информации можно увидеть проект, который вы выберете из списка. С помощью нее можно будет увидеть список спринтов и задач проекта, а также кнопки "Добавить", "Редактировать" и "Удалить".

По причине отсутствия задач и спринтов проект нельзя считать закрытым. Для того чтобы определить минимальную дату закрытия, необходимо знать дату создания проекта, даты начала и окончания спринтов и задачи проекта.

Отмена проекта устраняет все вспомогательные задачи и спринты.

**Спринты**

Список не завершенных спринтов отображается при выборе пункта меню "Спринты". В соответствии с пользовательскими правами кнопки отображаются. После выбора спринта, который будет выбран из списка, откроется страница, где можно найти более детальную информацию.

Принт не может быть закрыт, пока существуют другие нерешенные задачи. Для того чтобы определить минимальную дату закрытия, необходимо знать дату создания спринта и даты завершения задач.

Планируемая дата является минимальной по сравнению с теми датами, которые будут определены для выполнения проекта и спринта.

Отмена спринта устраняет все задачи, которые не были связаны с ним.

Из списка проектов, которые не были закрыты при редактировании спринта выбирается тот проект, который будет добавлен в список.

После нажатия кнопки добавить информацию на странице с детальной информацией, в создаваемом спринте автоматически создается проект предстоящего спринта.

**Задачи**

Если выбрать пункт "Задачи" в меню, то можно увидеть список неоконченных задач. В соответствии с пользовательскими правами кнопки отображаются. Когда будет выбрана задача, которая соответствует выбранному списку, откроется страница с подробной информацией. Пункт "Исполнитель" появляется в фильтрах по пользователю.

Занимательные задачи, которые имеют отношение к проекту, задаче или спринту отображаются на странице детальной информации. Там можно найти список зависимых задач и историю их изменения, а также узнать о влиянии изменений в родительской задаче, спринте и проекте. Добавить информацию в историю возможно благодаря взаимодействию между исполнителем и автором задачи.

Разница между задачами, которые имеют отношение друг к другу, является одноуровневой. Задача, которая имеет в своем арсенале зависимые задачи, не может быть независимой.

Дата, которую следует назначить, должна быть минимальной по отношению к запланированным срокам выполнения проекта, спринта и задачи.

При изменении задачи, проект будет выбран из списка, который не закрыт. В процессе выбора проекта, спринт выбирается из списка не закрытых спринтов, который может быть изменен при выборе проекта. В случае, если спринт является основным видом деятельности, родительская задача будет выбрана из списка не закрытых задач. Если не определен спринт, то это может стать причиной для составления списка задач проекта. После выбора проекта или спринта, список задач будет переработан.

У исполнителя есть возможность изменять только дату завершения выполнения задачи.

В случае использования кнопки добавить информацию со страницы с детальной информацией, в создаваемой задаче автоматически будут показаны проект и спринт текущей задачи.

**Поиск**

Во время выбора в меню пункта "Поиск" отображаются списки проектов, которые не были завершены, а также спринтов и задач. В качестве фильтрации можно использовать статус и имя автора. В соответствии с пользовательскими правами кнопки отображаются. При поиске не имеет значения, какой регистрационный номер стоит в наименовании проекта или задачи.

**API**

Если выбрать пункт меню "API", то можно увидеть веб интерфейс фреймворка Django REST. Он имеет простой фильтр, который позволяет определить название поля и точное совпадение всех данных. В качестве примера можно привести:

/api/tasks/?date\_end=none&date\_beg=2023-12-11

С помощью этого способа можно только получать информацию. Отсутствует возможность создания и редактирования записей, используя API.

**Авторизация**

Пункт меню "Вход" отображается в правом верхнем углу, если пользователь не авторизован. Когда пользователь выбирает его, открывается страница авторизации. Во время успешной авторизации пользователя перенаправляет на главную страницу сайта.

Пункты меню "Смена" и "Выход" отображаются в правом верхнем углу при авторизации пользователя. Когда пользователь выбирает «Смена», происходит открытие страницы авторизации. Пользователи, которые выбирают вариант "Выход" получают статус не авторизованного пользователя. Пользователи, которые прошли процесс переавторизации или выхода из системы, получают доступ к главной странице.

## 2.3 Проектирование интерфейса

Каждый пользователь может создать свой уникальный контент для приложения. Поэтому желательно при разработке интерфейса выделить зону, предназначенную для интернет-пользователя, поэтому важно учитывать это при

Для обеспечения одинаковой эффективности работы при различных объемах информации, следует разработать систему, которая будет одинаково эффективно работать при разных видах контента.

Пользователи начинают изучать страницу с левой части страницы, что обусловлено тем, что она расположена в верхнем левом углу.

Уведомления о задачах, которые требуют повышенного внимания пользователя, представлены в виде карточек, которые расположены в центре экрана. Для того чтобы они были удобны в использовании, их необходимо было расположить так, чтобы они не занимали много места на экране, поэтому было принято решение сделать ее динамически увеличивающейся при нажатии.

В развернутом виде на карточке содержится вся информация о задаче, а также кнопки для редактирования и удаления. Цветовое оформление зависит от сферы, к которой относится задача.

Сферы и цели имеют абсолютно идентичные страницы, отличающиеся только содержимым.

На карточках сферы изображены в виде динамических объектов с информацией, которая постоянно скрыта. При нажатии на название пользователь может получить всю информацию о сфере.

Эта страница имеет ряд полезных функций: добавление новых сфер, редактирование и удаление уже существующих, карточка окрашивается в цвета, которые указал пользователь, также на ней расположена кнопка для отображения прогресса по достижению целей. Кроме того, на странице отображается шкала, где отображается прогресс по достижению поставленной цели.

В результате были разработаны макеты всех основных страниц и элементов приложения, которые послужили основой для создания страницы сайта.

**Итоги главы**

Создание веб-приложения для управления и планирования целей, задач и их решения было осуществлено на основе выявленных требований к создаваемой системе. На основе этого были разработаны схема базы данных и макеты страниц приложения.

Приложение 3 предоставляет возможность посмотреть на реализацию макетов интерфейса.

# Глава 3. Тестирование веб-приложения

Проведение тестирования состоит из двух блоков.

В случае если база не содержит информации, в первом блоке происходит генерация случайных данных и проверка того, насколько верно отображена информация в базе.

Если база пустая, в этом блоке запускаются случайные данные, а также происходит имитация открытия страниц со списками спринтов, проектов и задач. Данные проверяются на наличие шаблона и соответствие выбранным данным.

**Генерация данных**

Данный проект предоставляет возможность создания данных. Для того чтобы запустить файл generator.bat, можно воспользоваться следующей командой:

python manage.py shell -c="from app\_task.functions import gen\_data; gen\_data(cnt=50, close=60, clear=True, parent=True, clear\_user=True)"

В модуле app\_task.functions запустится функция gen\_data. С помощью ниже приведенного описания можно ознакомиться с параметрами функции, которые имеют значение по умолчанию.

Cnt=0 - количество проектов, которые планируется реализовать.

close=0 - На данный момент процент закрытых проектов и спринтов составляет 0.

clear=False - Следует производить очистку базы перед тем, как будет произведена генерация.

Для создания зависимых задач необходимо использовать метод Parent=False.

Используйте функцию удаления тестовых пользователей, если они не соответствуют требованиям безопасности.

Для данных выбираются даты, которые находятся в интервале +/- 3 месяца от текущего периода. Настройки MY\_TEST\_USER содержат шаблон, который определяет начало имени тестового пользователя. В том случае, если его нет, то следует использовать user. Пароль может быть следующим: MY\_TEST\_PASS или же, если он отсутствует, то U-321rew. Шаг за шагом происходит процесс генерации данных.

**Шаги генерации данных:**

Все проекты (а также зависимые спринты и задачи) с которыми сталкиваются тестовые пользователи, будут удалены в случае наличия такой очистки базы.

Когда производится удаление тестового пользователя, автоматически удаляются все пользователи, которые не соответствуют шаблону.

Если количество создаваемых проектов не равно нулю, а также отсутствуют тестовые пользователи, то в этом случае будут созданы пять пользователей по шаблону:

Счетчик, начиная с 0, показывает значение MY\_TEST\_USER.

Изменения в настройках MY\_TEST\_PASS:

Проекты создаются в указанном количестве. В проекте указываются случайные автор и дата создания, а также планируемая дата прекращения деятельности.

Создается двойное число спринтов. В качестве авторов и проектов для спринтов можно использовать случайных людей, а также их даты создания и планируемые даты завершения.

При этом количество задач увеличивается до десяти. Задачи могут быть решены с помощью случайных авторов и исполнителей, проектов и спринтов, а также дат их создания и планируемой даты закрытия.

В случае, если в проекте имеются задачи, которые имеют зависимость от других, следует попробовать создать их. На основе случайного выбора выбираются две основные и зависимые задачи проекта.

Попытка закрыть количество спринтов, которое было указано. Предполагается выбор случайных дат, когда будут закрыты все задачи в течение спринта и дата завершения мероприятия.

Попытки закрытия определенного количества проектов. Занимается подбором случайных дат, когда будут закрыты все задачи в проекте, а также дата завершения проекта.

Благодаря использованию метода save model происходит запись данных. Исходя из этого, при записи проверяется корректность информации.

даты больше/меньше чем надо,

Возможно ли создание зависимости?

Необходимо ли будет закрыть проект или выполнить какую-либо задачу? И т.д.

**Итоги главы**

В процессе разработки приложений, тестирование является неотъемлемым этапом. Оно помогает выявить проблемы и проверить их решение, а также охватывает все функции приложения.

Приложение было протестировано и показало готовность к использованию в процессе использования.

Для полной проверки всех функций приложения, а также корректности отображения страниц на различных видах экранов в будущем будут написаны дополнительные тесты.

# Заключение

Чтобы стать успешным приложением, которое будет эффективно управлять проектами, оно должно удовлетворять различные запросы своих пользователей. Данное приложение призвано обеспечить эффективное управление задачами и ресурсами. Оно также должно стремиться к минимизации информационной нагрузки, а также содействовать в налаживании контактов между сотрудниками команды. Помимо этого, оно должно иметь возможность быть настроенным, масштабируемым и поддерживаться с помощью надёжной системы поддержки для исправления ошибок и обновлений, а также поддержкой, которая может гарантировать исправление ошибок и обновление.

В дипломном проекте была выполнена задача разработки веб-приложения, которое будет использоваться в процессе командной разработки для управления задачами.

В приложении реализован ряд дополнительных возможностей:

• Управление статусом задачи возможно осуществить с помощью функции управления.

• Удобство разделения заданий по проектам заключается в том, что можно выполнять различные действия с ними: разрабатывать, изменять или удалять их.

• В программе присутствует опция разделения задач, которые находятся в различных проектах на спринты, а также их изменения и удаления.

• Возможность посмотреть историю изменений, чтобы понять какие именно задачи были решены.

• Предоставляет возможность предупреждать пользователей о необходимости изменения текущего состояния задачи.

• С помощью определенного уровня доступа пользователям предоставляется возможность авторизации.

Проект соответствует следующим требованиям:

• Соблюдение правил авторизации.

• Наличие юнит-тестирования.

• Использование сериализаторов drf, а также rest api.

• Наличие системы логгирования.

• У пользователей есть возможность использовать графическое представление.

Удобное и функциональное веб-приложение позволяет сделать работу более легкой, понятной и прозрачной. Оно помогает распределять задания между сотрудниками таким образом, чтобы отслеживать их выполнение. По причине того, что на данный момент наблюдается тенденция к удаленной работе, необходимо наличие данного устройства.

Первая глава дипломного проекта посвящена анализу подобных приложений. В ходе этого исследования были определены те ключевые особенности, которые необходимы для каждого из приложений и на их основе можно сделать выводы о том какие именно технологии будут использованы при реализации данного проекта. Были проанализированы способы, которые помогут реализовать данное приложение.

Вторая глава описывает процесс создания веб-приложения, которое будет использоваться для управления и планирования целей, задач и их решения. Оно было создано на основе выявленных требований к системе. На базе этих данных были созданы макеты страниц приложения и схема базы данных.

В третьей главе было описано тестирование приложения. Чтобы провести полную проверку всех функций приложения и проверить корректность отображения страниц на различных видах экранов, в будущем планируется создание дополнительных тестов.

Требования к системе, которые были указаны в дипломной работе, были учтены в ходе реализации приложения. Помимо этого, на основе требований, которые были указаны выше, были созданы макеты страниц приложения и схема базы данных.

На будущее запланированы следующие работы:

• Создать систему, которая бы оценивала реалистичность выполнения задачи.

• Для проектов необходимо реализовать функционал чата.

• Функцию слияния проектов необходимо реализовать;

• С помощью API можно осуществить создание и редактирование записей.

• Установить дополнительные модули тестирования.

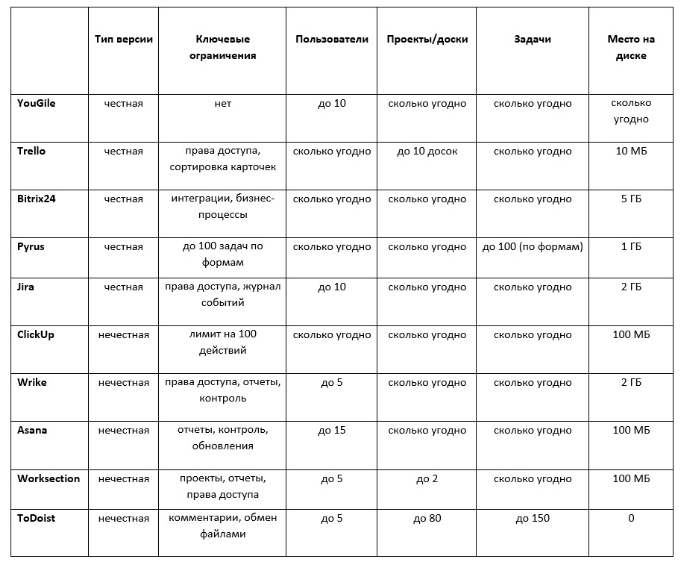
• Провести исправление ошибок и оптимизацию кода до уровня clean architecture.

# Список используемой литературы

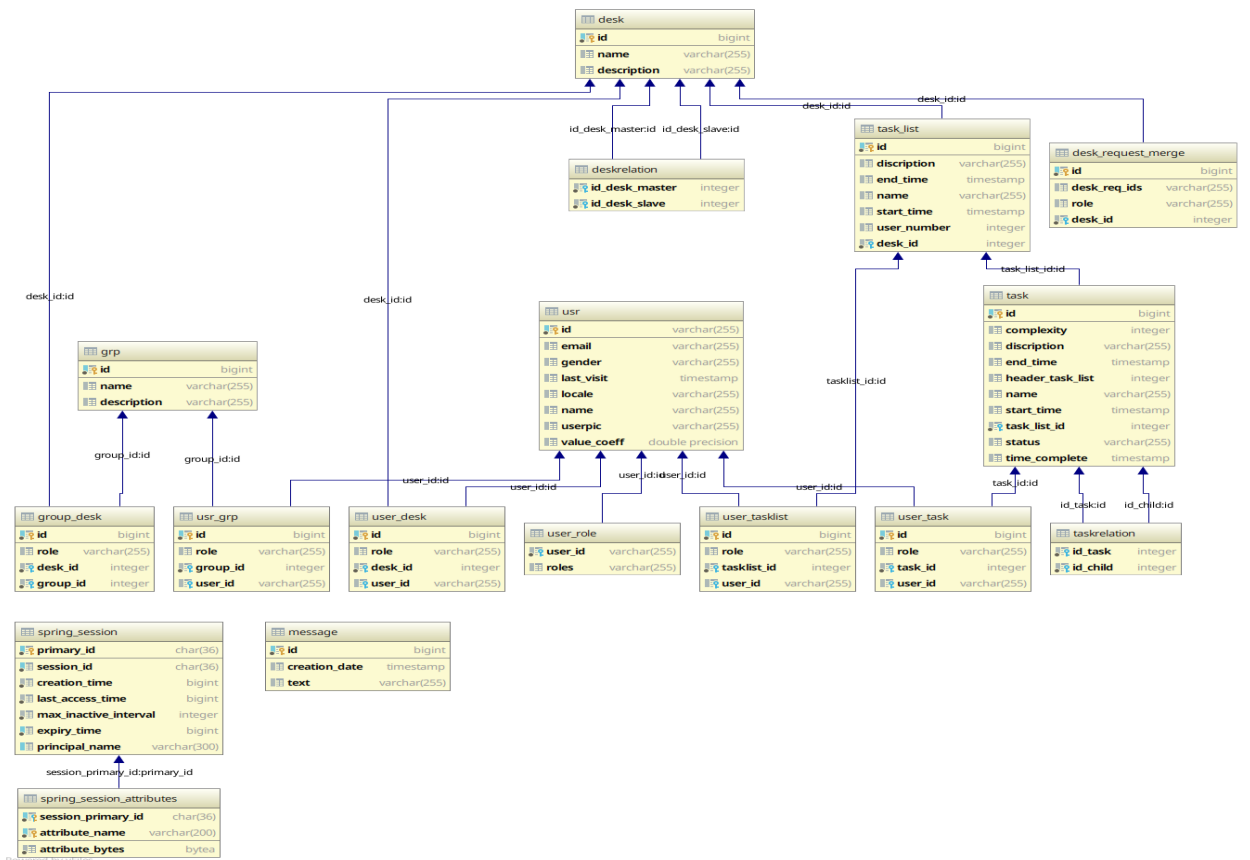
1. PEP 8 - руководство по написанию кода на Python - URL: https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html (дата обращения: 06.12.2023).
2. Python 3.12.0 documentation - URL: https://docs.python.org/3/ (дата обращения: 06.12.2023).
3. Аллен Д. Приводим жизнь в порядок. Экспресс курс по методике GTD / Д. Аллен. – М.: Альпина Бизнес Букс, 2012. – 224 с
4. Вайсфельд М. Объектно-ориентированное мышление / Вайсфельд М. – СПб.: Питер, 2014. – 304 с.
5. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.
6. Лутц М. Изучаем Python, том 1, 5-е издание. – Пер. с англ. – СПб.: Диалектика, 2020. – 832 с.
7. Лучано Рамальо. Python. К вершинам мастерства. – М.: ДМНК Пресс, 2016. – 768 с.
8. Прохоренок Н.А. Самое необходимое. — СПб.: БХВ-Петербург, 2011. — 416 с.
9. Россум, Г. Язык программирования Python / Г. Россум, Ф.Л.Дж. Дрейк, Д.С. Откидач и др.. - М.: [не указано], 2020. - 578 c.
10. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямc, 2016. – 592 с.

# Приложения

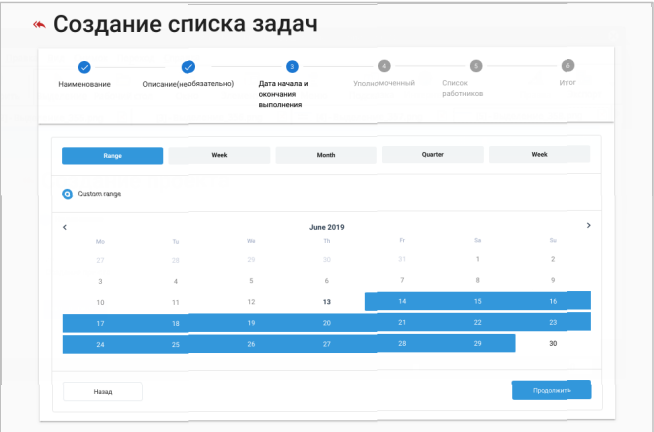
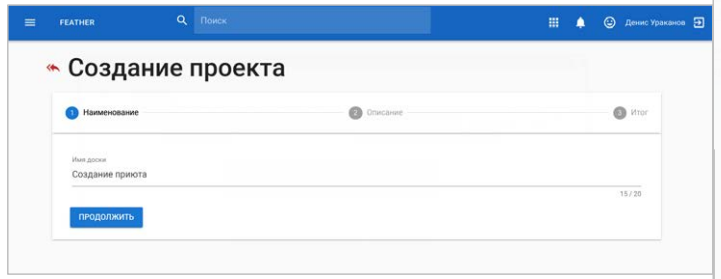
## Приложение 1. Результаты анализа аналогичных проектов.



## Приложение 2. Структура базы данных проекта.

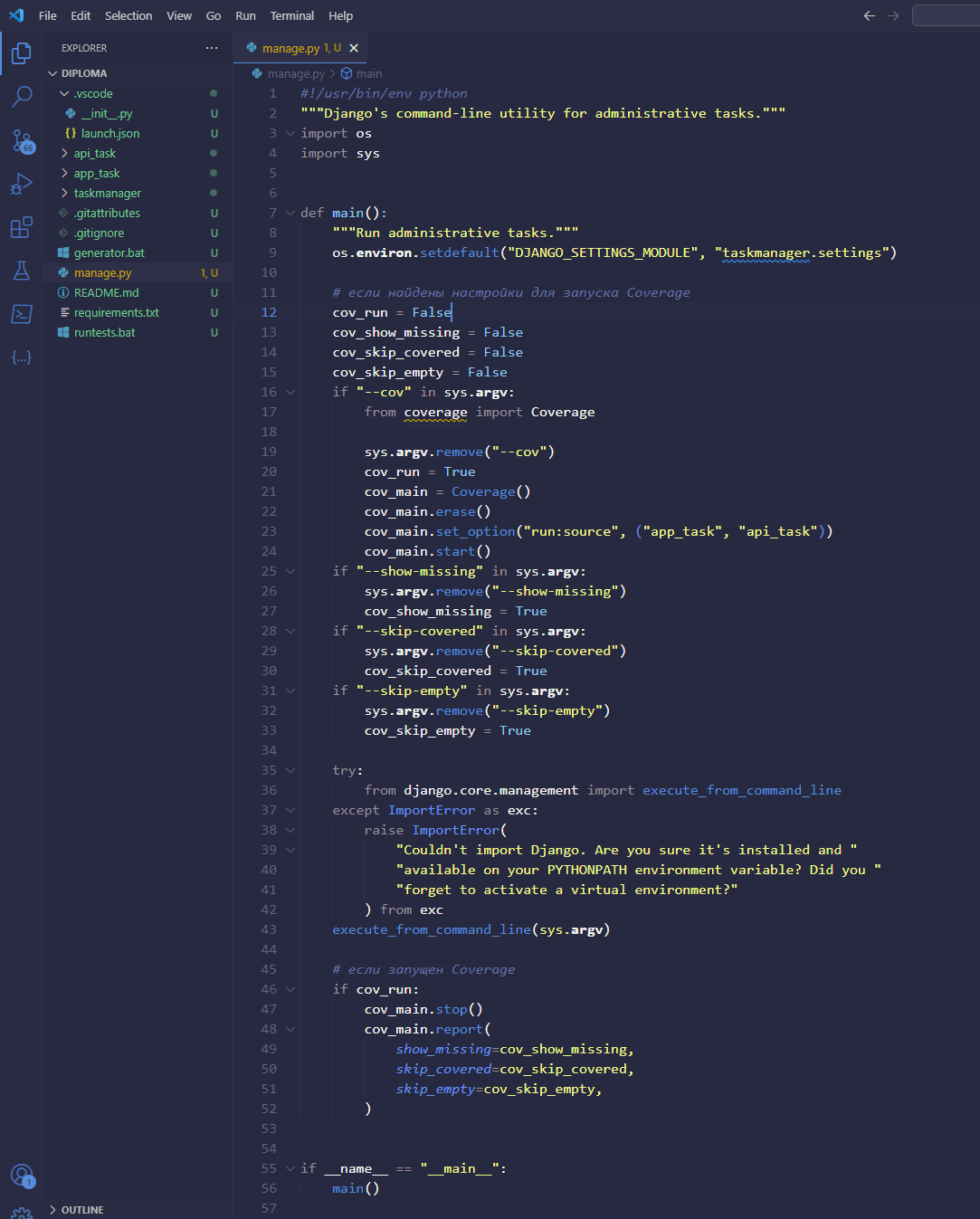


## Приложение 3. Макеты реализации интерфейса приложения.

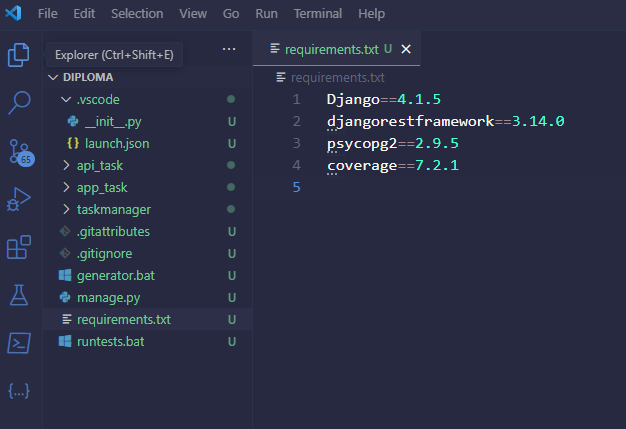


## Приложение 4. Код веб-приложения

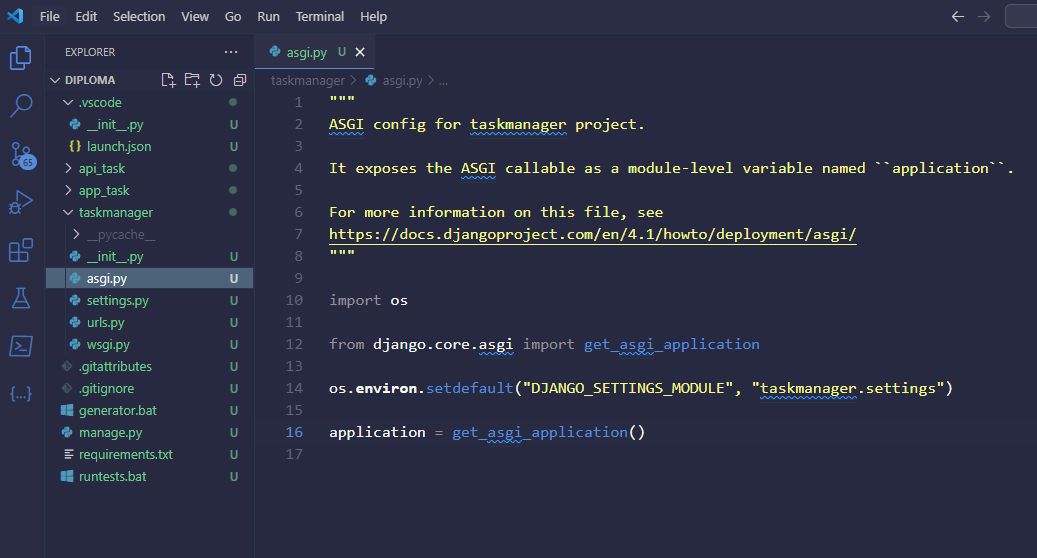
### 4.1 manage.py



### 4.2 requirements.txt

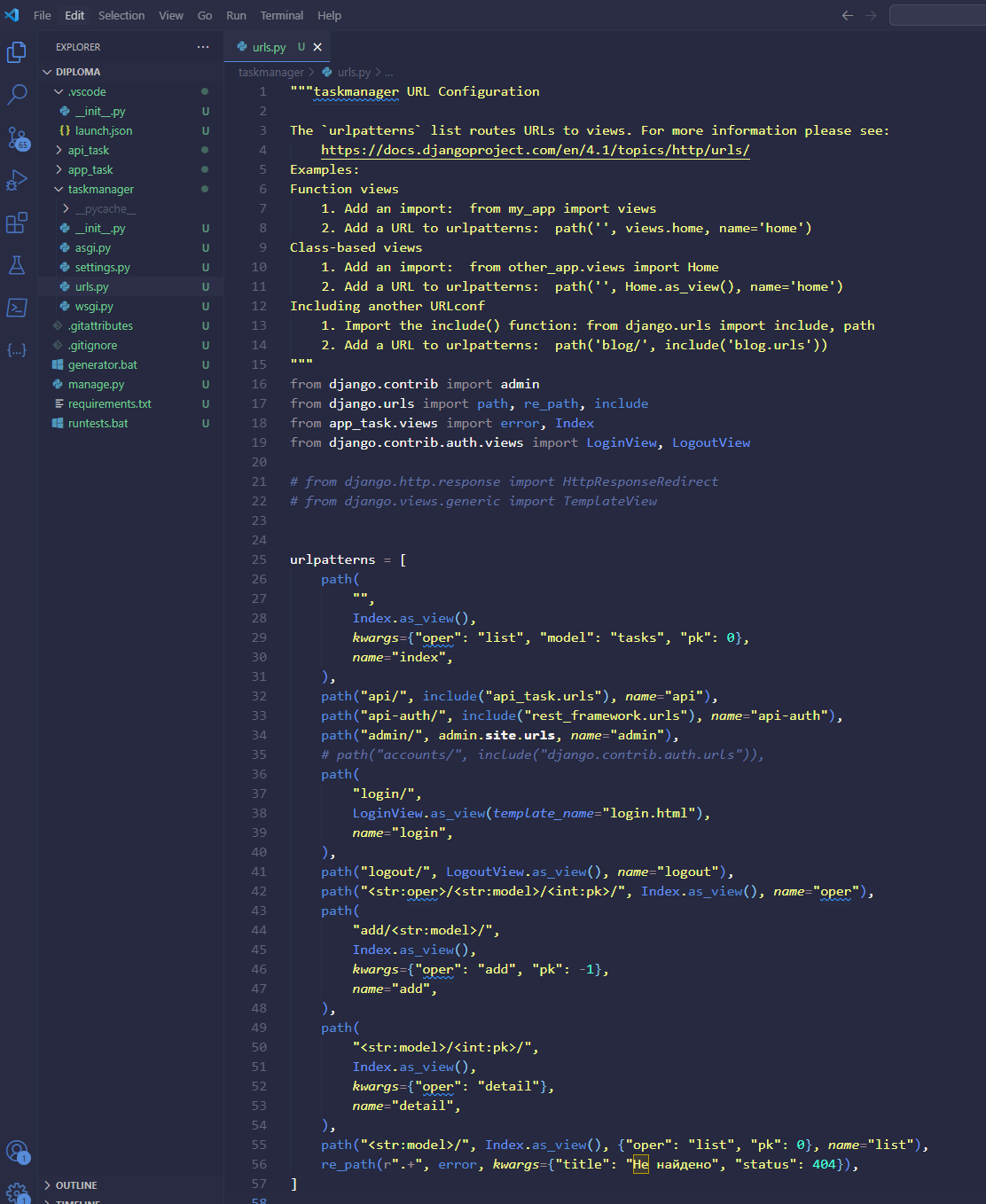


### 4.3 taskmanager/asgi.py

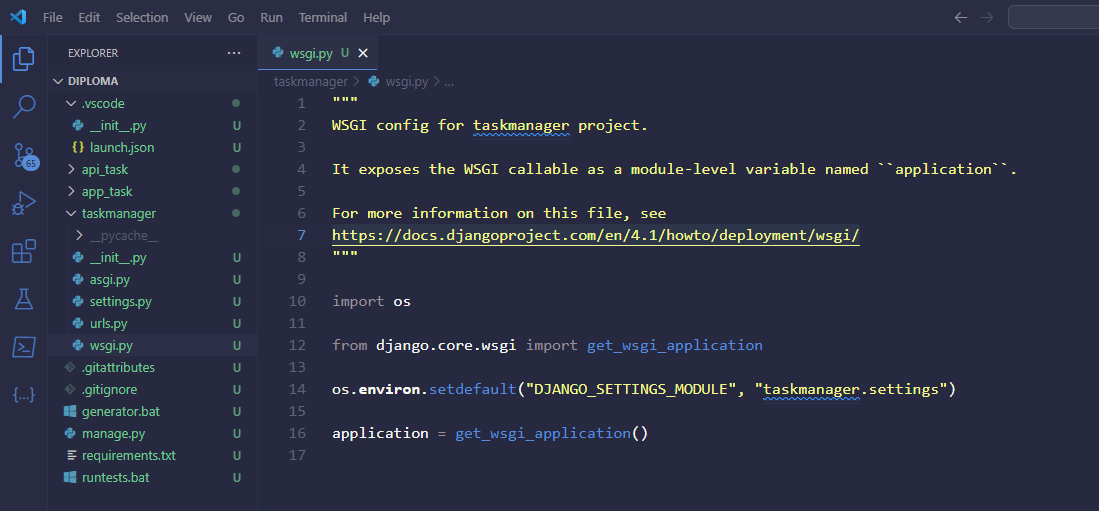


### 4.4 taskmanager/settings.py

### 4.5 taskmanager/urls.py



### 4.6 taskmanager/wsgi.py



## Приложение 5. Ссылка на приложение

<https://github.com/diana-diulberova/taskManager/tree/main/taskManager/taskManager>