СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном динамичном мире, где информационные потоки неустанно нарастают, а требования к эффективности и организации личного и рабочего времени становятся все более строгими, неотъемлемым инструментом становится систематизация задач и планирование деятельности. Одним из ответов на вызовы современной жизни является разработка приложения для планирования и управления рабочими и личными задачами, что представляет собой актуальную и перспективную область информационных технологий.

Проблема, с которой сталкиваются люди, связана с неэффективным управлением временем и задачами в современном мире. Требования к организации повседневной деятельности становятся все более строгими, а поток информации неустанно растет. В таких условиях разработка инновационного приложения становится насущной необходимостью, направленной на облегчение этой задачи для пользователей.

Актуальность темы курсового проекта обусловлена наличием современных задач, которые требуют эффективных средств управления. Существующие приложения для планирования задач, несмотря на свою распространенность, не всегда полностью отвечают потребностям пользователей, оставляя пробелы в функциональности и удобстве использования.

Смотря на готовые решения, можно заметить, что они часто ограничены в функциональности, не предоставляют гибкости в настройке и не всегда соответствуют современным стандартам эффективного управления временем. В связи с этим возникает необходимость в разработке собственного приложения, которое бы удовлетворяло все требования пользователей.

Целью данного курсового проекта является разработка приложения для планирования и управления рабочими и личными задачами, способного поддерживать пользователя в организации своей повседневной деятельности. Проект ориентирован на обеспечение простоты использования, гибкости

настроек и максимальной функциональности для учета разнообразных потребностей пользователей.

Были определены следующие задачи, которые необходимо решить в ходе курсового проектирования: исследовать предметную область; провести аналитический обзор существующих решений; определить стек используемых при разработке технологий; спроектировать и разработать структуру базы данных; спроектировать и разработать структуру приложения и алгоритмы работы его компонентов; реализовать приложение; провести тестирование и отладку.

В итоге курсового проектирования необходимо создать приложение, практическая значимость которого заключается в организации рабочего и личного времени пользователей, предоставляя простоту использования, гибкость настроек и максимальную функциональность. Его практическая значимость заключается не только в устранении трудностей, связанных с управлением задачами, но и в том, чтобы поддерживать пользователей на пути к повышению эффективности и достижению поставленных целей. Внимание к современным тенденциям в управлении временем и учет потребностей пользователей является ключевым элементом, обеспечивающим создание приложения, способного успешно интегрироваться в их повседневную жизнь.

1. ОБЩИЙ РАЗДЕЛ

1.1. Исследование предметной области

Без тщательного анализа задачи невозможно эффективно разработать приложение, отвечающее потребностям пользователей.

В современном обществе, где информационные потоки без устали нарастают, и требования к эффективности и организации времени становятся все более строгими, важность глубокого понимания задач и проблем, с которыми сталкиваются конечные пользователи, не может быть недооценена.
Этот анализ представляет собой ключевой этап, в ходе которого выявляются уникальные потребности разнообразных категорий пользователей. Различные контексты использования, особенности повседневной деятельности и индивидуальные предпочтения требуют внимательного рассмотрения, чтобы создать инструмент, который действительно будет соответствовать реальным потребностям пользователей.

Целевая аудитория разрабатываемого приложения тесно связана с пользователями, стремящимися к повышению своей продуктивности и эффективности управления своим личным и рабочим временем. В современном мире, насыщенном информацией и постоянно меняющимися рабочими условиями, возникает острая необходимость в инструментах, способных помочь пользователям эффективнее распределять и контролировать свои задачи.

Целевая аудитория включает в себя широкий спектр пользователей, начиная от школьников, студентов и предпринимателей до заботливых домохозяек.

После тщательного анализа поставленной задачи, было принято решение представить основные компоненты системы в форме UML-диаграммы.

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур [X].

На основе изучения предметной области были выделены роли, которые взаимодействуют с системой, каждая из них выполняет определенный набор функций. Результаты этого анализа отражены в виде диаграммы прецедентов, представленной на рисунке 1.1.

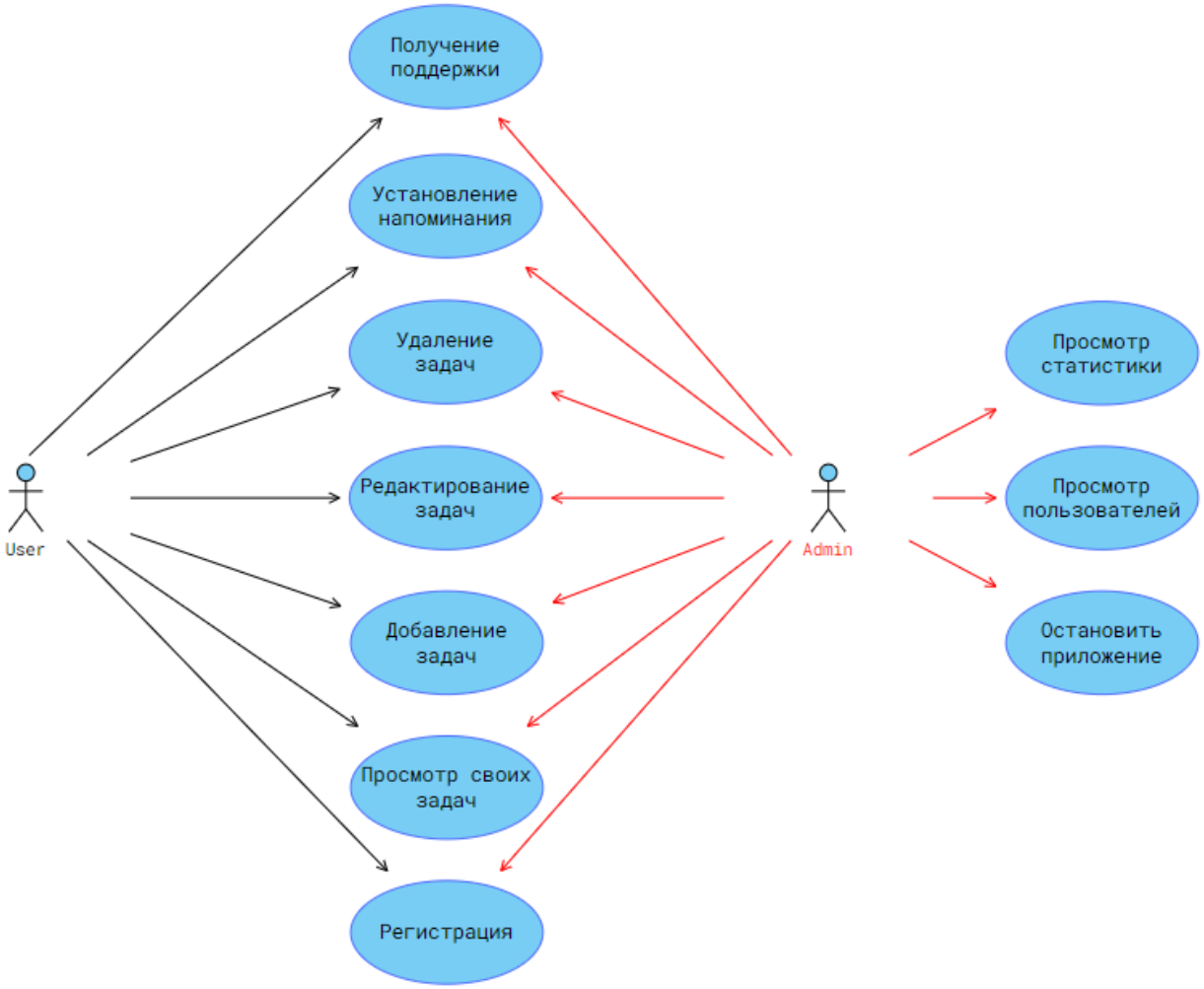

Рисунок 1.1 - UML-диаграмма вариантов использования

На диаграмме вариантов использования можно увидеть, что в системе будут присутствовать две роли: пользователь и администратор.

У пользователя доступны все функции разрабатываемого приложения, такие как: регистрация, просмотр своих задач, добавление задач, редактирование задач, удаление задач, установление напоминания, получение поддержки. У

администратора есть все выше перечисленное и в дополнение - остановка приложения, просмотр пользователей приложения, просмотр статистики.

Для отображения последовательности действий в информационной системе, была разработана диаграмма последовательности, представленная на рисунке 1.2.

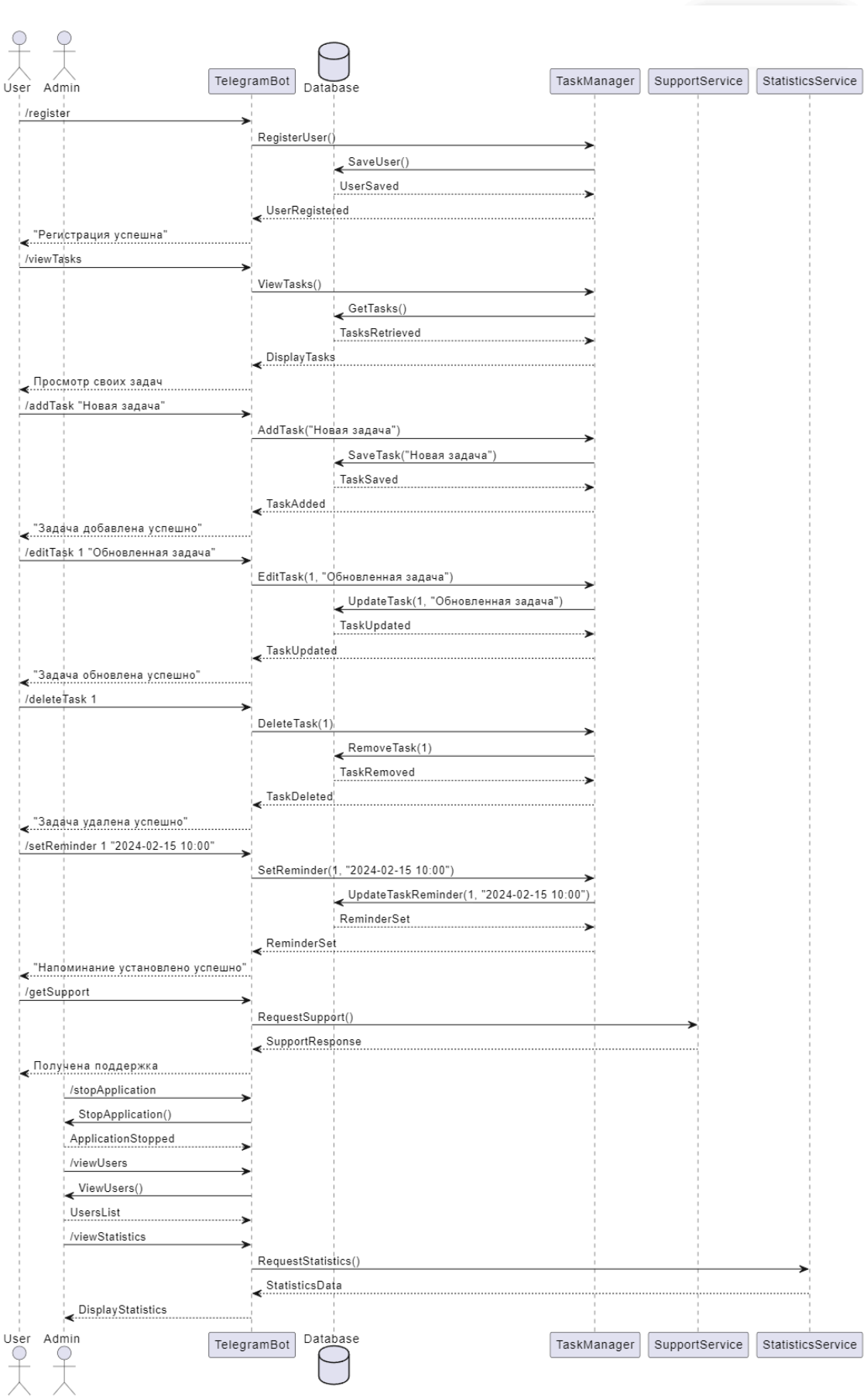

Рисунок 1.2 - UML-диаграмма последовательности

На данной диаграмме последовательности представлены действия, которые пользователь и администратор выполняют во взаимодействии с приложением управления задачами. Для использования приложения управления задачами пользователь должен следовать нескольким шагам. Сначала, он должен зарегистрироваться, отправив соответствующий запрос боту. После успешной регистрации пользователь может просматривать свои задачи, добавлять новые, редактировать, удалять, а также устанавливать напоминания для задач. В случае необходимости он может обратиться за поддержкой, отправив запрос боту. Все эти действия осуществляются с использованием механик Telegram API, что делает использование бота удобным и доступным для пользователя.

Представленные выше диаграммы отражают взаимодействие модулей системы в процессе выполнения основных функций. При регистрации пользователя система обращается к модулю управления задачами для сохранения данных в базе, после чего возвращает подтверждение. Запрос на просмотр задач активирует обращение к модулю для получения списка из базы, который затем предоставляется пользователю. Аналогично, операции добавления, редактирования, удаления задач и другие взаимодействия с базой данных и компонентами системы выполняются для обеспечения функциональности приложения.

Подводя итоги данного раздела, можно утверждать, что были выявлены ключевые аспекты, необходимые для разработки приложения, соответствующего потребностям пользователей. Внимание к уникальным запросам различных пользовательских категорий, особенностям контекстов использования и индивидуальным предпочтениям является неотъемлемой частью процесса. Результатом этого анализа стала UML-диаграмма прецедентов, отражающая взаимодействие пользователей и администратора с приложением, а также диаграмма последовательности для визуализации последовательности действий в системе. Полученная информация формирует основу для разработки приложения, способного эффективно удовлетворять

разнообразные потребности пользователей и обеспечивать эффективное управление задачами.

1.2. Аналитический обзор существующих решений

Сегодня можно найти много инструментов и решений для управления своим временем и задачами.

Проведем анализ двух существующих решений, предназначенных для планирования и управления задачами.

TickTick — сервис для создания и организации списков задач (Рисунок 1.3).

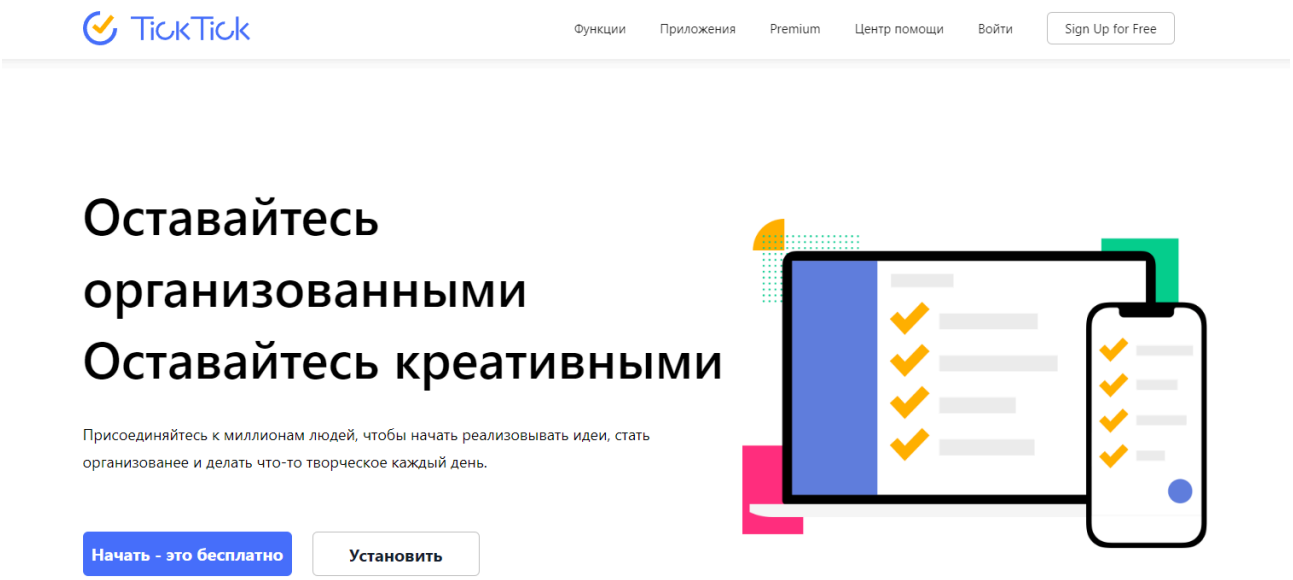

Рисунок 1.3 - Главный экран проекта TickTick

TickTick предоставляет обширный набор функций для управления задачами, включая создание подзадач, установку напоминаний и интеграцию с календарем, что делает его полезным инструментом для эффективного планирования времени. Многоплатформенность приложения обеспечивает синхронизацию данных между устройствами, а интуитивный интерфейс и гибкие настройки облегчают использование. Однако ограниченные бесплатные функции и необходимость в платной подписке для доступа к продвинутым возможностям могут ограничить опыт для некоторых пользователей. Возможны также проблемы с синхронизацией.

Следующая платформа для рассмотрения — Trello. Это платформа для управления проектами и задачами, основанная на концепции карточек и досок.(Рисунок 1.5).

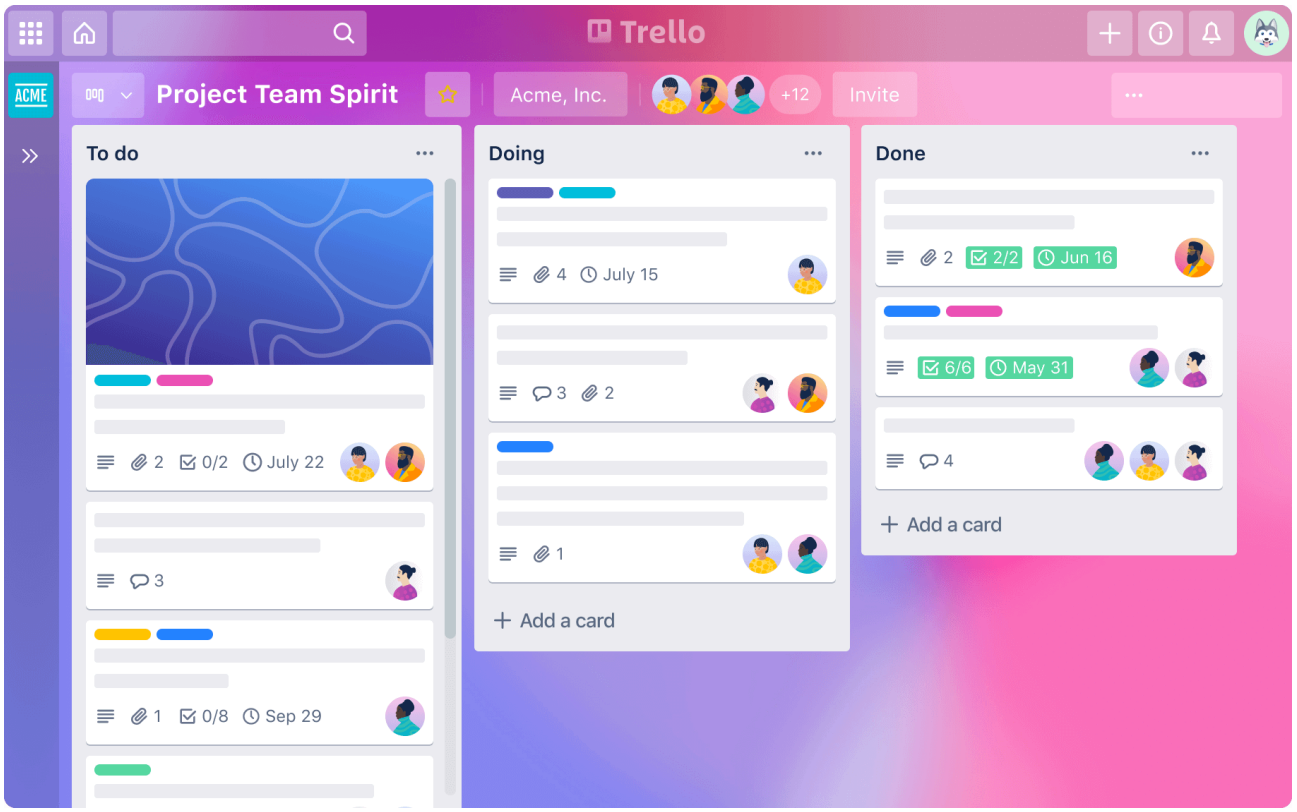

Рисунок 1.4 - Пример проекта Trello

Его преимущества включают интуитивный интерфейс, где пользователи могут создавать задачи в виде карточек, организовывать их на досках и следить за ходом выполнения проектов. Интеграция с различными приложениями и сервисами, а также наличие бесплатной версии, делают Trello привлекательным для широкого круга пользователей. Однако у Trello также есть свои недостатки. Ограниченные возможности в бесплатной версии могут ограничивать пользователя, а более расширенные функции доступны только в платных планах. Кроме того, в сравнении с TickTick, Trello предоставляет меньше инструментов для управления временем и создания задач. Как к минусам Trello для пользователей в России можно добавить, что в некоторых случаях могут возникнуть проблемы с доступом, так как сервис может быть блокирован. Это может создать неудобства для пользователей, пытающихся использовать Trello в России.

Чтобы выявить достоинства и недостатки вышерассмотренных инвестиционных решений по сравнению друг с другом, был проведен сравнительный анализ этих продуктов, результаты которого представлены в таблице 1.1.

Таблица 1.1 – Сравнительный анализ существующих решений

Функциональные особенности	TickTick	Trello
Создание задач и подзадач	Да	Да
Установка напоминаний	Да	В ограниченной степени
Интеграция с календарем	Да	В ограниченной степени
Кроссплатформенность	Да (веб, мобильные устройства, десктоп)	Да (веб, мобильные устройства, десктоп)
Интуитивный интерфейс	Да	Да
Гибкие настройки	Да	Ограничены в бесплатной версии
Бесплатная версия	Да (с ограничениями)	Да (с ограничениями)
Платные возможности	Да (расширенные функции)	Да (расширенные функции)
Проблемы с синхронизацией	Иногда возникают	В ограниченных случаях
Интеграция с другими приложениями	В ограниченной степени	В ограниченной степени
Доступность в России	Да	Могут возникнуть проблемы из-за блокировки

Оба рассмотренных решения, TickTick и Trello, предоставляют разнообразные инструменты для управления задачами, но выбор между ними зависит от конкретных потребностей пользователя. TickTick обеспечивает более

обширный функционал, включая управление временем, однако для полного доступа могут потребоваться платные подписки. Trello привлекателен своей интуитивной структурой карточек и досок, но его ограниченные возможности в бесплатной версии и региональные ограничения в России могут быть ограничивающими для целевой аудитории. Разработка собственного решения может быть рассмотрена для учета конкретных требований и избежания ограничений существующих платформ.

1.3. Обоснование выбора стека технологий

1.3.1. Выбор языка программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под ее управлением.

Для разработки проекта по планированию и управлению задачами, рассмотрим следующие языки программирования.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Преимущества Python:

- Простота и читаемость кода, что ускоряет разработку и облегчает поддержку.
- Богатая экосистема библиотек, включая те, которые предназначены для работы с Telegram API.
- Широкое использование в разработке ботов и наличие обширного сообщества разработчиков.
- Легкость освоения благодаря простоте, логичности и понятному синтаксису, что делает его подходящим даже для новичков без знания английского языка.
- Кроссплатформенность, позволяющая программам, написанным на Python, запускаться на различных операционных системах. Отличия в поведении могут быть предварительно изучены в документации.
- Скорость разработки, так как для написания программы на Python требуется значительно меньше кода, чем при использовании, например, Java.

Недостатки Python:

- В некоторых случаях может быть менее эффективным в асинхронном выполнении операций.
- Трудности переноса проектов на другие системы из-за зависимости от библиотек.
- Ресурсоемкость, что делает Python неоптимальным выбором для проектов, требующих больших объемов памяти.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

Преимущества JavaScript:

- Асинхронная природа JavaScript обеспечивает эффективную обработку запросов, что важно для интерактивных ботов.
- Обширный выбор библиотек и фреймворков (например, Telegraf.js) упрощает разработку и обеспечивает функциональность для сложных приложений.
- Возможность использования JavaScript на обеих сторонах (клиент и сервер) обеспечивает единый стек технологий.
- Динамическая типизация и удобный синтаксис способствуют быстрой разработке гибких приложений.

Недостатки JavaScript:

- Возможные уязвимости в коде могут привести к проблемам безопасности, особенно при использовании сторонних библиотек.
- Ошибки времени выполнения из-за отсутствия строгой типизации требуют внимательного тестирования.

C# — объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[6] как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core.

Преимущества C#:

- C# интегрируется хорошо с платформой .NET, что обеспечивает богатый набор инструментов и библиотек для разработки приложений, включая телеграм-боты.
- C# предоставляет богатые возможности языка, поддерживая объектно-ориентированное программирование, асинхронные операции, и другие современные концепции.
- Интегрированная среда разработки Visual Studio обеспечивает удобную и продуктивную среду для создания и отладки C#-проектов.
- Существуют библиотеки, такие как Telegram.Bot, которые упрощают разработку телеграм-ботов на C#, предоставляя абстракции для работы с API Telegram.

Недостатки C#:

- В сравнении с некоторыми другими языками, такими как Python и JavaScript, C# может иметь меньшее распространение в области разработки телеграм-ботов.
- Новичкам может потребоваться больше времени для освоения C# и платформы .NET, чем других языков.
- В сравнении с некоторыми популярными языками, экосистема ботов на C# может быть менее развитой, и некоторые библиотеки могут быть менее обширными.

Выше рассмотренные языки программирования имеют свои преимущества и недостатки, но для данного проекта я выбираю Python из-за его простоты, читаемости и обширной поддержки для работы с API, включая API Telegram.

1.3.2. Выбор системы управления базами данных

СУБД — комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД

Рассмотрим следующие СУБД для хранения данных приложения.

SQLite — это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

Преимущества SQLite:

- SQLite является встроенной базой данных, что упрощает её использование без необходимости установки дополнительных серверов.
- SQLite обладает простым SQL-синтаксисом, что делает его достаточно легким для использования и понимания, особенно для простых проектов.
- База данных SQLite компактна и не требует большого объема ресурсов, что может быть важно для маленьких проектов или проектов с ограниченными ресурсами.
- SQLite поддерживает транзакции, что обеспечивает целостность данных и защиту от сбоев в процессе записи.
- SQLite не требует запуска отдельного сервера, что может упростить развертывание и управление базой данных.

Недостатки SQLite:

- Для больших и сложных проектов с большими объемами данных SQLite может оказаться менее производительным по сравнению с более мощными базами данных.

- Несмотря на свою простоту, SQLite может не предоставлять некоторых возможностей, доступных в более продвинутых системах управления базами данных.
- SQLite может иметь проблемы с производительностью при одновременных записях из-за блокировок на уровне базы данных.

PostgreSQL — это объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире. Имеет открытый исходный код и является альтернативой коммерческим базам данных.

Преимущества PostgreSQL:

- PostgreSQL предоставляет богатый набор функций, включая поддержку сложных запросов, триггеры, хранимые процедуры и многое другое, что может быть полезно при разработке сложных телеграм-ботов.
- PostgreSQL известен своей хорошей производительностью, особенно при работе с большими объемами данных и сложными запросами.
- PostgreSQL обеспечивает поддержку транзакций, что важно для обеспечения целостности данных в случае ошибок или сбоев.
- PostgreSQL поддерживает хранение и манипулирование данными в формате JSON и JSONB, что может быть удобным для работы с данными телеграм-бота, особенно если используется Telegram Bot API, возвращающее данные в формате JSON.
- PostgreSQL является открытым исходным кодом, что обеспечивает гибкость и возможность настройки под специфические требования проекта.

Недостатки PostgreSQL:

- В отличие от встроенных баз данных, таких как SQLite, PostgreSQL требует установки и настройки отдельного сервера.

- PostgreSQL может требовать больше ресурсов по сравнению с более легкими базами данных, что может быть проблемой для проектов с ограниченными ресурсами.
- Для небольших проектов использование PostgreSQL может показаться избыточным из-за его мощных функций, и более простые базы данных могут быть более подходящими.

Выше рассмотренные СУБД имеют свои преимущества и недостатки, но для данного проекта я выбираю SQLite из-за его легкости и простоты интеграции с приложением. Поскольку приложение будет маломасштабным и не требует мощной СУБД, SQLite будет более подходящим вариантом.

1.3.3. Выбор дополнительных средств разработки

Библиотека — сборник подпрограмм или объектов, используемых для разработки программного обеспечения. С точки зрения операционной системы и прикладного программного обеспечения, библиотеки разделяются на динамические и статические.

Библиотека служит для того, чтобы облегчить труд человека и ускорить его показатели во много раз. Для разработки проекта была выбрана библиотека TeleBot.

Git — для управления версиями кода в проекте планируется использовать систему контроля версий.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.