**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **09.03.03 Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

**К У Р С О В О Й   П Р О Е К Т**

Тема: «Телеграм бот для напоминаний»

Обучающийся: Ступичев Михаил Николаевич К3139

Санкт-Петербург 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_heading=h.gjdgxs)

[ОСНОВНАЯ ЧАСТЬ 5](#_heading=h.30j0zll)

[Описание проекта 5](#_heading=h.1fob9te)

[Этапы работы над проектом 7](#_heading=h.3znysh7)

[Мой вклад в проект 10](#_heading=h.2et92p0)

[Анализ своей работы 12](#_heading=h.tyjcwt)

[Взаимодействие с командой 14](#_heading=h.3dy6vkm)

[Взаимодействие с руководителем проекта 16](#_heading=h.1t3h5sf)

[Оценка работы руководителя 17](#_heading=h.4d34og8)

[ЗАКЛЮЧЕНИЕ 18](#_heading=h.2s8eyo1)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_heading=h.17dp8vu)

[ПРИЛОЖЕНИЕ 21](#_heading=h.3rdcrjn)

# ВВЕДЕНИЕ

В современном ритме жизни, когда у каждого из нас каждый день количество задач, целей и дел постоянно растет, управление своим временем становится критически важным. Вероятно, у каждого человека была ситуация, когда не специально он пропускал какой-либо важный созвон или встречу. Различные приложения-напоминания, заметки и ежедневники помогают нам в организации, но необходимость постоянно переключаться между ними и проверять их может отвлекать от текущих дел. В таком потоке информации легко забыть о важных событиях или пропустить дедлайн. Потребность в надежном и удобном инструменте, который бы своевременно отправлял напоминания непосредственно в привычную среду общения, является крайне актуальной, особенно для студентов.

Наш бот в Telegram представляет собой решение этой проблемы, предоставляя пользователям простой и эффективный способ устанавливать и получать напоминания, не отвлекаясь от текущей работы или общения. Это позволит сэкономить время, не пропустить важные дела и повысить общую продуктивность.

Основной целью данного курсового проекта является разработка асинхронного Telegram-бота для отправки напоминаний. Бот предоставляет возможность добавлять, просматривать, удалять и, конечно, получать напоминания.

Для достижения поставленной цели были решены следующие задачи:

* Разработка модели данных для хранения информации о напоминаниях пользователей в базе данных PostgreSQL.
* Реализация функционала обращения к базе данных.
* Реализация асинхронной функциональности приема и обработки команд пользователя для добавления, просмотра, удаления и отправки напоминаний.
* Использование Docker для контейнеризации приложения и базы данных.
* Тестирование и отладка системы для обеспечения надежной работы бота.

Для разработки проекта был выбран язык программирования Python, как наиболее подходящий для создания асинхронных приложений и работы с Telegram Bot API. Основной библиотекой для работы с Telegram API является aiogram, обеспечивающая асинхронное взаимодействие с API Telegram. Для работы с базой данных PostgreSQL используется библиотека psycopg2, а для асинхронного взаимодействия с ней – aiopg. Также используется стандартная библиотека time для отслеживания времени. Для упрощения развертывания и управления приложением используются Docker и Docker Compose, с двумя отдельными Dockerfile для контейнеризации приложения и базы данных.”

# ОСНОВНАЯ ЧАСТЬ

## Описание проекта

Telegram-бот для напоминаний представляет собой инструмент, предназначенный для упрощения управления временем и задачами пользователей. В современном ритме жизни люди часто сталкиваются с трудностями при запоминании важных событий, сроков и дел. Наш бот решает эту проблему, предоставляя простой и эффективный способ устанавливать напоминания и получать уведомления прямо в мессенджере.

Основными пользователями бота являются все, кто нуждается в своевременных напоминаниях, будь то студенты, или же просто люди, желающие лучше организовывать свое время. Бот предоставляет им следующие возможности:

* Добавление напоминаний с указанием текста, даты и времени.
* Просмотр списка всех запланированных напоминаний.
* Удаление ранее созданных напоминаний.
* Получение уведомлений в заданное время через личные сообщения от бота.

Архитектура системы построена на основе асинхронного взаимодействия с Telegram Bot API и базой данных. aiogram, отвечает за обработку команд пользователя в Telegram, планирование уведомлений и взаимодействие с базой данных PostgreSQL. База данных, управляемая psycopg2 и aiopg, хранит информацию о напоминаниях пользователей. Все компоненты развернуты в отдельных Docker-контейнерах для удобства масштабирования и управления.

Описание технологий:

* Python. Является популярным языком для разработки ботов и асинхронных приложений. Имеет множество библиотек для работы с Telegram API и базами данных.
* Aiogram. Асинхронная библиотека для работы с Telegram Bot API, обеспечивает эффективное взаимодействие с API и высокую производительность.
* Psycopg2 и aiopg. Позволяют эффективно работать с PostgreSQL, одной из самых популярных, безопасных и эффективных моделей баз данных, обеспечивая надежное хранение и доступ к данным.
* Docker и Docker Compose. Обеспечивают удобство развертывания, масштабирования и управления компонентами приложения. Позволяют изолировать приложение и базу данных.
* Библиотека time. Достаточно точный и удобный инструмент для отслеживания времени.

## Этапы работы над проектом

Работа над проектом была разделена на следующие этапы, учитывая распределение ролей в команде:

Планирование и анализ требований – нулевой этап:

Сначала мы с командой провели созвон для обсуждения целей нашего проекта, функциональности бота и технических деталей реализации.

Определи основные возможности бота: добавление, просмотр и удаление напоминаний, а также их получение в заданное время.

Провели распределение ролей и задач между членами команды, учитывая их компетенции:

1. Дизайнер:
   * Разработка дизайна для напоминаний и самого интерфейса бота.
   * Поиск аналогов, отчет по их минусам и плюсам.
   * Составление документации для бота.
2. SQL разработчик БД:
   * Разработка схемы базы данных PostgreSQL, необходимой для визуализации хранения пользователей и информации о их напоминаниях.
   * Реализация базы данных в PostgreSQL с использованием Docker для контейнеризации.
3. БД API разработчик:
   * БД API разработчик разработал API для взаимодействия с базой данных PostgreSQL.
   * Реализация основных CRUD методов (create, read, update, delete) для обращения к базам данных.
4. Разработчик Telegram API:
   * Реализация основной логики бота, включая обработку команд пользователя, взаимодействие с Telegram Bot API через aiogram, а также планирование и отправку уведомлений.
   * Реализация логики работы с БД API для сохранения, чтения и удаления напоминаний.
   * Проведение тестирования функционала для обеспечения стабильной работы.
5. Докладчик:
   * Подготовка презентации для доклада на защите проекта, описывающая работу остальных участников команды и их работу.

Разработали общий план работы и определили сроки для каждого ее этапа.

Первый этап:

Первым этапом для всех участников было изучение шаблонного репозитория в Github, в котором мы и должны были работать. Далее каждый участник изучал свою область, которую он должен был реализовать по заданным требованиям:

1. Дизайнер

* Изучить существующие решения для напоминаний и управления задачами в формате телеграмм бота.

2. SQL разработчик БД

* Изучить основы работы с SQL

3. БД API разработчик

* Изучить основы работы с SQL
* Изучить основы python asyncio

4. Разработчик Telegram API

* Изучить основы python asyncio
* Изучить aiogram

Так как последняя участница нашей команды присоединился к нам уже на половине работы, для нее была выбрана роль Докладчика. В первом этапе она не участвовала.

Второй этап:

Вторым и самым трудным этапом была реализация и тестировка заданных задач для каждого из участников. Наша команда работала в Github и поддерживала несколько веток разработки, в каждой из которых разрабатывалась определенная часть программы. Это позволило нам избежать нескольких ошибок, неизбежных для работы в команде, и так же каждый мог отслеживать процесс разработки других участников и их прогресс.

Третий этап:

На последнем – третьем этапе, нашей задачей было объединить все части кода и проверить бота на работоспособность. Также была проведена работа над подготовкой к защите, в которой каждый участник составлял небольшой отчет, по которому он хотел презентовать.

Четвертый этап:

Защита проекта. Каждый участник рассказал о проделанной работе и результате, которого удалось добиться.

## Мой вклад в проект

Так как мне досталась роль SQL разработчика БД, моими задачами было разработать схему базы данных PostgreSQL, необходимую для пользователей и информации о их напоминаниях, и реализовать ее с использованием Docker для контейнеризации.

В начале мне, как и другим, нужно было изучить шаблон проекта на Github, сделанный нашим руководителем для структуризация нашего проекта. Благодаря данному шаблону, мне удалось узнать, как нужно правильно оформлять проекты и код в командных работах для лучшего его понимания другими участниками и в том числе тестировщиками. Также данный шаблон уже включал в себя нужные Dockerfiles, что облегчило дальнейший запуск, но также пришлось разобраться с таким понятием как контейнеризация и как работает сам Docker [1].

После изучения шаблона я приступил к своей части работы. До этого мне уже приходилось делать telegram ботов и создавать базы данных, но в рамках нашего проекта мне пришлось расширить свои знания и поэтому сначала я изучил основы работы с PostgresSQL и придумал схему базы данных для нашего бота, наиболее подходящую под наши задумки [2]. Она представлена на рисунки 1.



Рисунок 1 – Схема базы данных телеграмм бота.

Далее моей задачей было создать данную базу данных, написав ее на SQL в нашем репозитории [3], проверить ее работоспособность, запустив в Docker и совместить ветку, в которой я работал, с основной.

Последним, что мне нужно было сделать, было составление доклада для защиты проекта для успешной его сдачи. В нем мне нужно было также рассказать о базе данных нашего бота, из чего она состоит и как работает.

## Анализ своей работы

Я считаю, что успешно справился с поставленными задачами. В ходе работы я столкнулся с несколькими трудностями, например, запуск Docker и самой базы данных, которые приходилось решать, узнавая информацию в интернете. Самым часто используемым инструментом для меня был ChatGPT [4]. Я использовал множество идей, которые он предложил для проекта, дорабатывая их до идеала, и также обращался к нему для поиска узкоспециализированной информации.

Так как роли в нашей команде были четко распределены, я общался с сокомандниками в основном уже на стадии объединения кода в полноценную, рабочую программу. В этом нам сильно помогла система Git [5], с помощью которой все заработало довольно быстро.

Что у меня получилось:

* Разработана схема базы данных PostgreSQL, необходимая для визуализации хранения пользователей и информации о их напоминаниях.
* Реализована база данных в PostgreSQL с использованием Docker для контейнеризации.

Основные трудности:

* Работа с структурированным, но довольно сложным для начального понимания шаблонным репозиторием.
* Работа с контейнерами и Dockerом, в особенности их запуск и проверка на работоспособность.

Я научился:

* Создавать схемы баз данных.
* Работать с контейнерами (Dockerом).
* Работать в команде и взаимодействовать с другими разработчиками и руководителем.
* Работать с ветками в системе Git.
* Навыкам поиска информации и составления нужных промтов для ChatGPT.
* Навыкам эффективного планирования своей работы и распределения времени.

## Взаимодействие с командой

Взаимодействие в нашей команде полностью было в Telegram канале, организованном руководителем. На первом созвоне мы обсудили, какие навыки уже есть у нас, и кто больше подходит на каждую роль, а также какие задачи перед нами стоят первоначально. Далее мы постоянно обсуждали ход работы и обращались к другим, если от них нужна была какая-то информация. Несколько раз проводились созвоны для обсуждения продвижения по проекту и выставления новых, задач. Так как у моих сокомандников также был опыт работы над ботами, они всегда помогали при каких-либо проблемах друг-другу и мне.

На середине работы нам добавили в команду еще одного участника, что было довольно тяжело, так как уже все задачи были распределены, но я считаю, что наша команда и руководитель разрешили эту ситуацию правильно, оставив на нее составление документации и презентации для защиты проекта.

Также мы взаимодействовали, отслеживая продвижение каждого из участников в Github и отдельных ветках, что помогало понять на какой стадии продвижения мы сейчас находимся. На рисунке 2 ниже представлена получившееся схема веток.

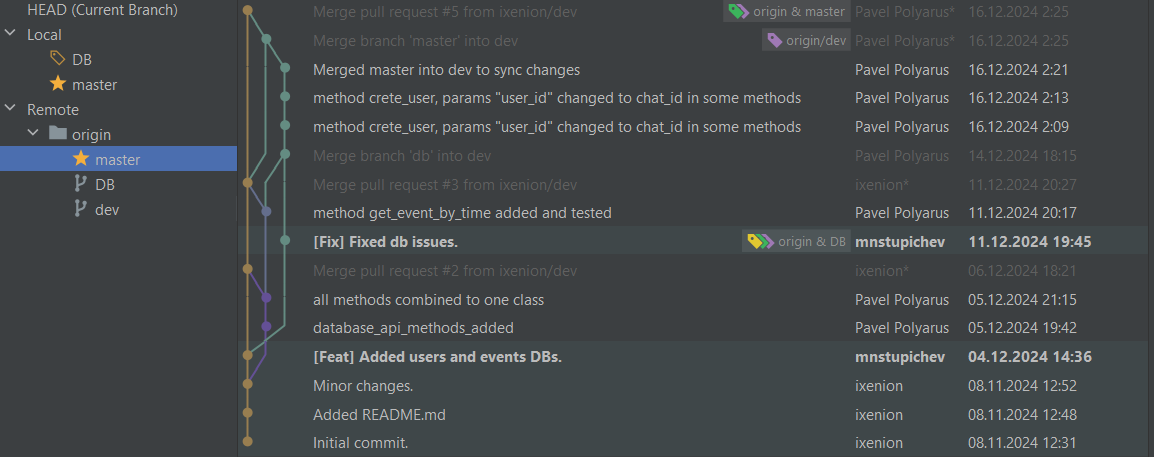


Рисунок 2 – Схема веток в системе Git.

Так как вся команда делала все свои задания в срок и отвечала на все вопросы своевременно, мы успешно выполнили нашу задумку. На защите проектов из нашей команды выступал каждый участник, что значит, что каждый внес вклад и вся командная работа была выполнена. На рисунке 3 представлена полностью завершенная схема задач нашей команды.

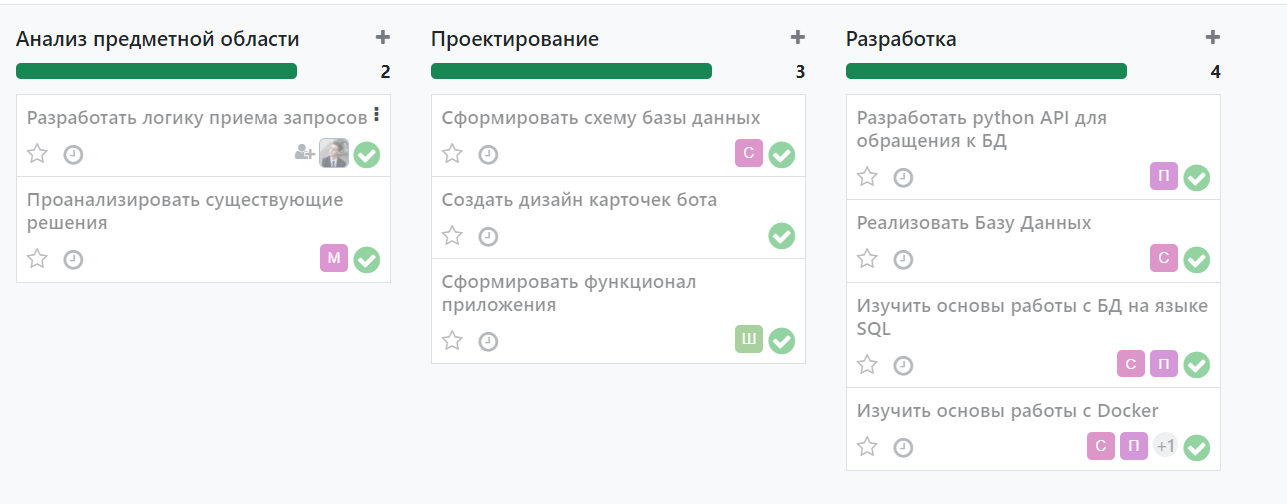


Рисунок 3 – Поэтапная схема выполненных задач.

## Взаимодействие с руководителем проекта

Взаимодействие с руководителем также происходило в Telegram. На первом созвоне он рассказал нам о целях, задачах и самом проекте. Также провел распределение по ролям на основе способностей. В дальнейшем руководитель в основном контролировал дедлайны, которые он заготовил, полагаясь на наш уровень и данное нам время на проект, и помогал, при возникновении вопросов по коду.

Очень важной и нужной частью работы с руководителем был его заготовленный шаблон на Github. Он безусловно помог определиться каждому, в какой части кода ему нужно работать. Руководитель немного рассказал про него, и в первую очередь рассказал, как запустить Docker контейнер.

На стадии тестирования и объединения кода, руководитель контролировал нас и помогал, если мы не справлялись. Также он очень оперативно ввел в курс дел новоприбывшего и, дав задание ей, также помог справиться с задачей.

## Оценка работы руководителя

Я бы оценил работу своего руководителя проекта на отлично. Он был организован и был заинтересован в успешном завершении и сдаче проекта. Также он всегда был готов помочь и ответить на мои вопросы, дополнительно скинув ссылки на проверенные источники.

Хочу отметить, что именно организованность руководителя и его четкие дедлайны и полностью прописанные задачи помогли лично мне не забывать о проекте, и на протяжении всего времени постепенно дорабатывать его.

Также я считаю, что наш руководитель отлично справился, когда нам на половине работы добавили человека. Он смог придумать, как перестроить наш уже готовый план под нее и включить в процесс разработки.

На будущее стоило бы еще более детально разбирать подготовленный шаблон проекта на Github, тем более, если для людей работа с контейнерами и твоим стилем кода в новинку. Также неплохо было бы делать побольше промежуточных дедлайнов, чтобы полностью отслеживать процесс разработки.

# ЗАКЛЮЧЕНИЕ

Цель проекта, заключавшаяся в создании удобного бота для создания и получения напоминаний, была достигнута. Все поставленные задачи по разработке успешно выполнены. Бот предоставляет основные функции: добавление напоминаний с указанием текста, даты и времени, просмотр списка напоминаний и удаление ранее созданных напоминаний. Пользователи получают уведомления в заданное время в виде личных сообщений от бота.

Хотя наша команда и выполнила все поставленные изначально задачи, есть еще множество функций, которые можно интегрировать в нашего бота: отправление напоминания несколько раз, добавление календаря, расширенный интерфейс, но из-за ограниченного времени нам не удалось это сделать.

Мой личный вклад в достижение цели проекта:

Мой вклад в проект состоял в разработке базы данных. В качестве SQL разработчика БД, я отвечал за:

* Разработка схемы базы данных PostgreSQL, необходимой для визуализации хранения пользователей и информации о их напоминаниях.
* Реализация базы данных в PostgreSQL с использованием Docker для контейнеризации.

Моя работа обеспечила надежную основу для всего проекта, предоставив структурированное и эффективное хранилище для данных о напоминаниях. В процессе работы над проектом я получил ценный опыт в проектировании баз данных для приложений и работе с PostgreSQL, Docker и некоторыми библиотеками Python. Я научился:

* Создавать схемы баз данных.
* Работать с контейнерами (Dockerом).
* Работать в команде и взаимодействовать с другими разработчиками и руководителем.
* Работать с ветками в системе Git.
* Навыкам поиска информации и составления нужных промтов для ChatGPT.
* Навыкам эффективного планирования своей работы и распределения времени.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Принципы работы с Docker [Электронный ресурс] – URL: [https://tproger.ru/translations/how-to-start-using-docker](https://tproger.ru/translations/how-to-start-using-docker%20) (дата обращения 29.11.2024).
2. Инструмент для создания блок схем [Электронный ресурс] – URL: [https://www.smartdraw.com/entity-relationship-diagram/database-diagram-tool.htm](https://www.smartdraw.com/entity-relationship-diagram/database-diagram-tool.htm%20) (дата обращения 4.12.2024).
3. Документация PostgreSQL [Электронный ресурс] – URL: [https://www.postgresql.org/docs/current/tutorial.html](https://www.postgresql.org/docs/current/tutorial.html%20) (дата обращения 27.11.2024).
4. ChatGPT [Электронный ресурс] – URL: [https://chatgptchatapp.com/](https://chatgptchatapp.com/%20) (дата обращения 25.11.2024 – 17.12.2024).
5. Принципы работы с Git [Электронный ресурс] – URL: [https://habr.com/ru/articles/541258](https://habr.com/ru/articles/541258%20) (дата обращения 26.11.2024).

# ПРИЛОЖЕНИЕ

Техническое задание на разработку Telegram бота

1. Общие положения
   1. Название проекта: телеграмм бот для напоминаний.
   2. Цель (назначение): разработка Telegram бота, который позволит пользователям устанавливать напоминания и получать уведомления в заданное время через Telegram, тем самым улучшая их организацию и продуктивность.
   3. Сроки выполнения: начало - 2024-11-01, конец - 2024-12-18.
   4. Команда проекта: исполнитель проекта (руководитель проекта) – Шаламов Павел Валерьевич, дизайнер – Мамаева Варвара Александровна, разработчик бд – Ступичев Михаил Николаевич, разработчик бд api – Полярус Павел Анатольевич, разработчик telegram api – Грицкевич Константин Олегович, докладчик – Кузнецова Анна Александровна.
   5. Этапы задач: анализ предметной области, проектирование, разработка, тестирование.
2. Технические требования
   1. Возможность добавления напоминаний с указанием текста, даты и времени
   2. Возможность просмотра списка всех запланированных напоминаний
   3. Возможность удаления ранее созданных напоминаний
   4. Получение уведомлений в заданное время через личные сообщения от бота
   5. Бот должен работать с использованием асинхронного программирования
   6. Использование базы данных PostgreSQL для хранения информации о напоминаниях
   7. Интуитивно понятный и простой интерфейс
   8. Взаимодействие с пользователем через текстовые команды в Telegram
   9. Контейнеризация приложения и базы данных с использованием Docker
   10. Бот должен обрабатывать запросы быстро и эффективно
   11. Система должна быть способной обрабатывать запросы множества пользователей
3. Ожидаемый результат
   1. Полностью функциональный Telegram-бот, отвечающий всем описанным в техническом задании требованиям.