#### *****Техническое задание (ТЗ)*****

**Название проекта**: Telegram-бот «Расписание АИТ»  
**Исполнитель**: Гегамян Давид , студент 3 курса  
**Цель**: Разработка Telegram-бота для предоставления расписания занятий студентов АИТ в удобной форме.

***1. Общие положения****:*

* **Наименование программы**: Telegram-бот «Расписание АИТ».
* **Область применения**: Использование студентами и преподавателями для быстрого доступа к расписанию занятий.
* **Основание для разработки**: Учебное задание, предоставленное в рамках практической работы.

***2. Функциональные требования****:*

* **Основные функции**:
  + Вывод расписания занятий на конкретный день недели.
  + Возможность запроса расписания для конкретной группы.
  + Вывод расписания на текущую или следующую неделю.
  + Уведомление о ближайших изменениях в расписании.
* **Интерфейс**:
  + Работа через Telegram (чат-бот).
  + Простая и интуитивно понятная структура команд:
    - /start — приветственное сообщение.
    - /расписание — общее расписание на текущую неделю.
    - /группа N — расписание группы N на текущую дату.
    - /завтра — расписание на следующий *день.*

***3. Нефункциональные требования****:*

* **Надежность**: Бот должен корректно обрабатывать некорректные запросы (например, несуществующие группы или неправильный формат даты).
* **Производительность**: Ответ на запросы пользователя не более 1-2 секунд.
* **Совместимость**: Совместимость с последними версиями Telegram API.

***4. Средства разработки****:*

* Язык программирования: Python.
* Используемые библиотеки: python-telegram-bot, requests.
* Серверная часть (реализация через Leadteh.ru API).

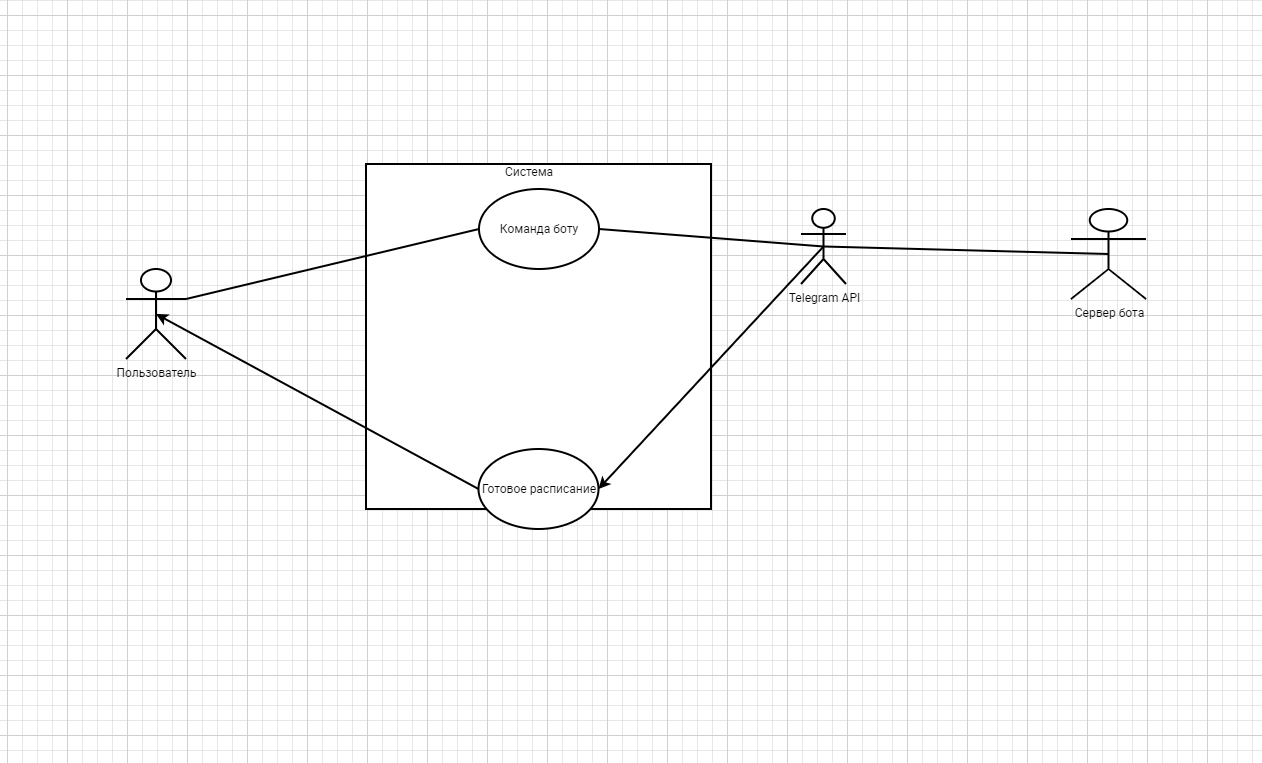
#### ***Шаг 2: Диаграмма UML взаимодействия клиента с сервером***

**Описание диаграммы:**

* **Акторы**:
  + Пользователь (студент, преподаватель).
  + Telegram API (посредник).
  + Сервер https://aitanapa.ru/ (хранение данных расписания).
* **Процесс взаимодействия**:
  1. Пользователь отправляет команду через Telegram-бота.
  2. Бот обращается к серверу Telegram API.
  3. Telegram API перенаправляет запрос серверу https://aitanapa.ru для получения расписания.
  4. https://aitanapa.ru возвращает данные о расписании.
  5. Telegram-бот формирует ответ и отправляет его пользователю через Telegram API.

**Диаграмма UML:**

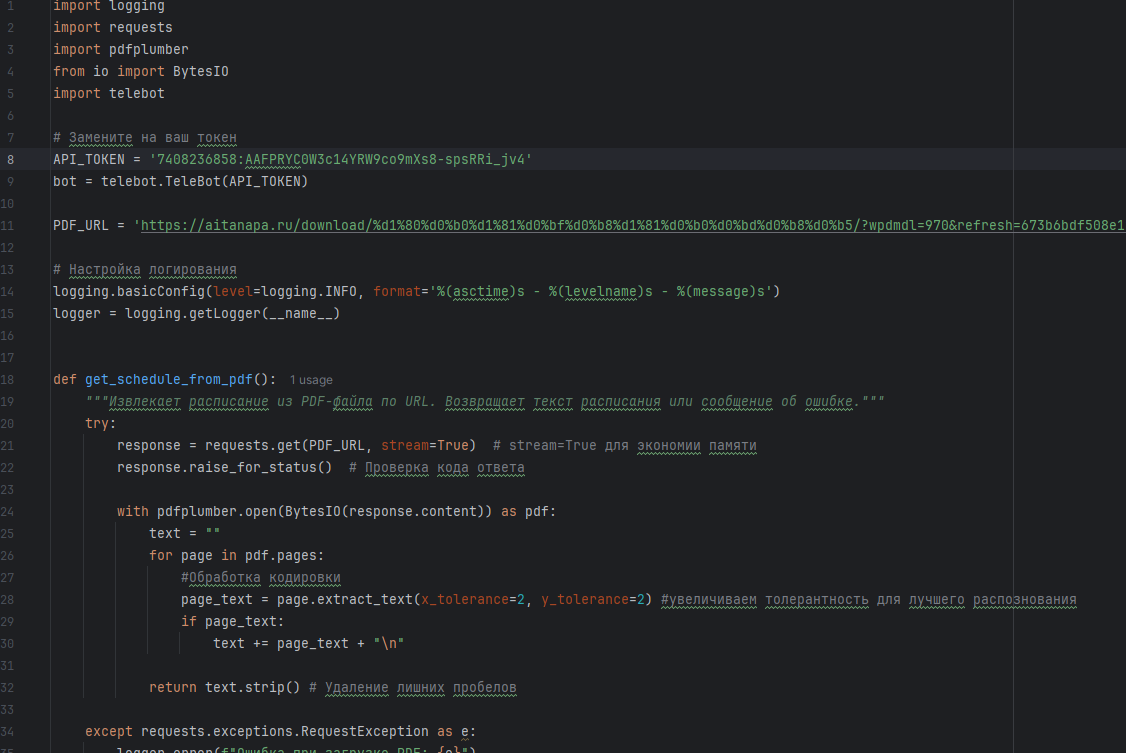
**Создана диаграмма вариантов использования отображающая взаимодействие между пользователем, Telegram API и сервером бота:**

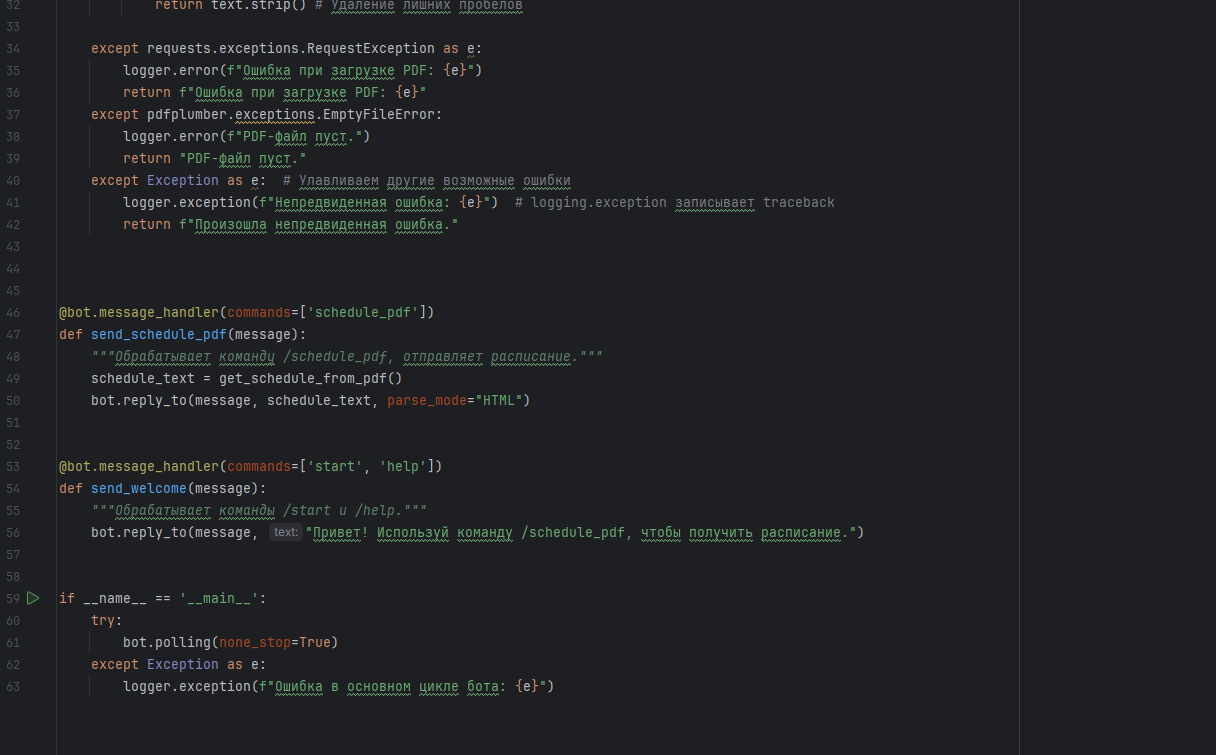


#### *****Шаг 3: Алгоритм работы*****

1. ***Запуск****:*
   * Пользователь вводит команду /start — бот отправляет приветственное сообщение.
2. ***Получение запроса****:*
   * Пользователь выбирает команду (например, /расписание или /группа N).
   * Бот проверяет корректность команды.
3. ***Обработка запроса****:*
   * Если запрос корректен, бот формирует запрос к серверу https://aitanapa.ru для получения данных расписания.
   * Если запрос некорректен, бот отправляет сообщение об ошибке.
4. ***Ответ пользователю****:*
   * Сервер возвращает данные, бот форматирует их и отправляет пользователю.
5. ***Завершение****:*
   * Бот ожидает следующего запроса.\

Разработка и написание исходного кода.





Бот готов к развертыванию на сервере и использованию.