|  |  |
| --- | --- |
|  |  |
|  |  |

СОДЕРЖАНИЕ

[АННОТАЦИЯ 3](#_Toc161781507)

[ВВЕДЕНИЕ 5](#_Toc161781508)

[1.1. Наименование программы 5](#_Toc161781509)

[1.2. Документ, на основании которого ведется разработка 5](#_Toc161781510)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc161781511)

[2.1. Назначение программы 6](#_Toc161781512)

[2.1.1. Функциональное назначение 6](#_Toc161781513)

[2.1.2. Эксплуатационное назначение 7](#_Toc161781514)

[2.2. Краткая характеристика области применения 7](#_Toc161781515)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 8](#_Toc161781516)

[3.1. Постановка задачи на разработку программы 8](#_Toc161781517)

[3.2. Описание и обоснование архитектуры программы 12](#_Toc161781518)

[3.2.1. Описание архитектуры программы 12](#_Toc161781519)

[3.2.2. Обоснование архитектуры программы 19](#_Toc161781520)

[3.3. Описание и обоснование алгоритма работы программы 23](#_Toc161781521)

[3.3.1. Описание алгоритма работы программы 23](#_Toc161781522)

[3.3.2. Обоснование алгоритма работы программы 23](#_Toc161781523)

[3.4. Описание и обоснование выбора метода организации входных и выходных данных 24](#_Toc161781524)

[3.5. Описание и обоснование выбора состава технических и программных средств 25](#_Toc161781525)

[3.5.1. Состав технических и программных средств 25](#_Toc161781526)

[3.5.2. Обоснование выбора технических и программных средств 25](#_Toc161781527)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 27](#_Toc161781528)

[4.1. Ориентировочная экономическая эффективность 27](#_Toc161781529)

[4.2. Предполагаемая потребность 27](#_Toc161781530)

[4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 28](#_Toc161781531)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 31](#_Toc161781532)

[ПРИЛОЖЕНИЕ 1 32](#_Toc161781533)

[ПРИЛОЖЕНИЕ 2 34](#_Toc161781534)

[ПРИЛОЖЕНИЕ 3 37](#_Toc161781535)

[ПРИЛОЖЕНИЕ 4 41](#_Toc161781536)

# **АННОТАЦИЯ**

Данная программная документация представляет собой пояснительную записку к программному проекту «Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий», предназначенного для обеспечения и упрощения работы преподавателей на серверах приложения Discord, повышения интерактивности проведения онлайн-занятий с учетом запросов пользователей бота-ассистента. Помимо этого, можно также подстроить бота “Mugiwara” под иные нужды сервера и пользователей благодаря расширенному, легко адаптируемому и интуитивно понятному функционалу, который реализован в данном Discord-боте.

Раздел «Введение» включает в себя наименование программы и документ, на основании которого ведётся разработка, с указанием утвердившей данный документ организации.

В разделе «Назначение и область применения» содержатся функциональное и эксплуатационное назначения программы и ее краткая характеристика области применения.

В разделе «Технические характеристики» содержатся следующие подразделы: постановка задачи на разработку программы, описание функционирования программы, описание и обоснование алгоритма работы программы, описание и обоснование выбора метода организации входных и выходных данных, описание работы с базой данных, описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указываются предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными приложениями с такой же функциональностью.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов[1];
2. ГОСТ 19.102-77 Стадии разработки[2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов[3];
4. ГОСТ 19.104-78 Основные надписи[4];
5. ГОСТ 19.105-78 Общие требования к программным документам[5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом[6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению[7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78[8], ГОСТ 19.604-78[9].

**ВВЕДЕНИЕ**

* 1. **Наименование программы**

Наименование программы – «Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий».

Наименование программы на английском языке – «Discord bot-assistant “Mugiwara” as a tool for conducting online classes».

* 1. **Документ, на основании которого ведется разработка**

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта «Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий».

Документ был утвержден организацией: Национальный исследовательский университет «Высшая школа экономики», Факультет компьютерных наук, образовательная программа «Программная инженерия».

1. **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**
   1. **Назначение программы**
      1. **Функциональное назначение**

Discord бот-ассистент “Mugiwara” реализует возможность интерактивного взаимодействия студентов между собой и с преподавателями, помощь в организации и проведении онлайн-занятий, предоставляет решение актуальных проблем и не только. С его помощью можно будет получить актуальную информацию о серверной статистике, выполнять продвинутую модерацию, организовывать и решать тесты, переводить сообщения, и это далеко не весь функционал ассистента “Mugiwara”.

Основной целью бота является организация и проведения онлайн тестов в формате квизов и упрощение модерации сервера с помощью интеллектуального определения. В нем пользователи с ролью Студент смогут проходить тесты, а преподаватели и модераторы – организовывать данные тесты и проводить модерацию.

Приложение «Discord бот-ассистент “Mugiwara”» состоит из двух частей: клиентской и серверной.

Клиентская часть реализует пользовательский интерфейс приложения. С помощью неё пользователь может выполнять команды известные для бота-ассистента и погрузиться в процесс онлайн-обучения благодаря красивому оформлению, интерактивности взаимодействия и интуитивно понятному для пользователей интерфейсу с описанием.

Серверная часть предназначена для хранения данных пользователей, серверной статистики, а также обрабатывания и исполнения клиентских запросов в виде команд.

В качестве дополнительного функционала будет возможно проигрывание видео/музыки с платформы YouTube. Также в дополнительном функционале будет реализована функция запроса к системам OpenAI и получения ответов на запросы, что поможет ученикам в изучении материала с использованием интеллектуального агента.

* + 1. **Эксплуатационное назначение**

Бот “Mugiwara” будет применяться в сфере образования и развлечений. Он предоставит возможность продвинутой модерации текстовых каналов сервера с помощью системы умного определения нарушений и специальных команд, организацию тестов в формате игровых квизов, проигрывать видео с youtube, собирать серверную статистику, а также смогут с помощью умного ассистента OpenAI отправлять запросы и получать ответы на них.

Данный бот является универсальным решением и облегчает дальнейшее взаимодействие между преподавателями и учениками, поскольку повышается интерактивность проведения онлайн занятий и интерес к учебе, также будет интерес к данному боту среди различных развлекательных серверов приложения Discord.

* 1. **Краткая характеристика области применения**

«Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий» – продвинутый бот платформы Discord, который позволит пользователям данной платформы с помощью рабочих команд умного ассистента упростить работу на серверах и повысить интерактивность во время проведения занятий. Благодаря данному продвинутому боту сильно упрощается организация преподавательского процесса, проведение онлайн-занятий и заинтересованность в учебном процессе, исходя из прорывного и продвинутого функционала ассистента “Mugiwara”.

Целевой аудиторией Discord бота-ассистента “Mugiwara” являются различные преподаватели, школьники и другие пользователи, для которых будет полезен функционал продвинутого бота-ассистента, а также лица, которые будут нуждаться в автоматизации модерации сервера в приложении Discord и повышении интереса и интерактивности в участии в онлайн-занятиях и мероприятиях серверов Discord. Основная цель разрабатываемого продвинутого бота-ассистента – автоматизация и упрощение процесса модерации серверов платформы Discord, помощь и обеспечение организации и проведения учебных квизов в виде интерактивных тестов, реализованных при помощи средств библиотек discord и dhooks.

1. **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**
   1. **Постановка задачи на разработку программы**

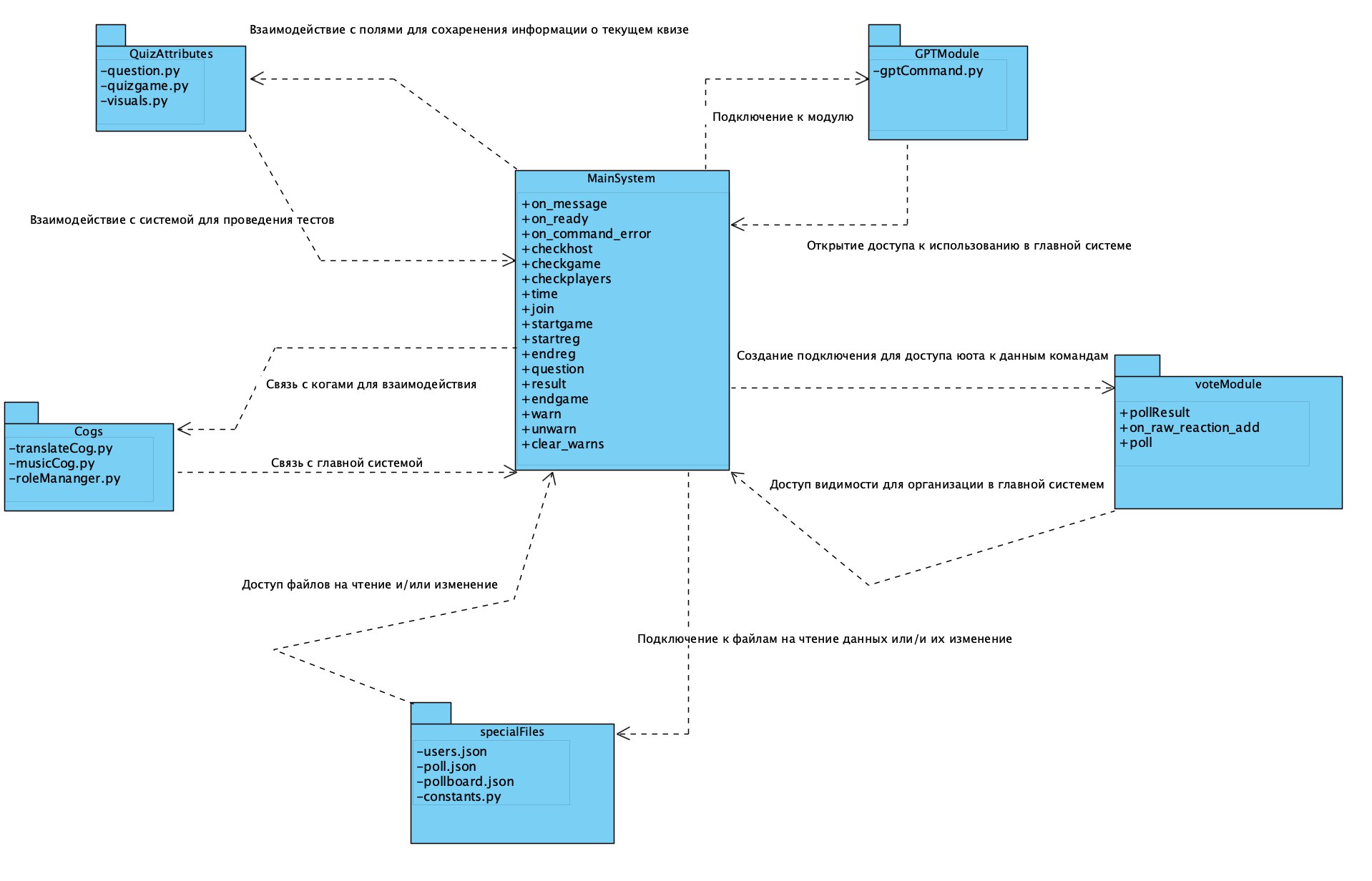
Разрабатываемая программа должна выполнять следующие поставленные задачи:

* Выдача ролей при входе на сервер с помощью интерактивного выставления реакции на сообщение сервера с отображением кастомизированых ролей данного сервера.
* Проверка знаний учеников в различных категориях с помощью организации и проведения интерактивных тестов/квизов. Команды, отвечающие за зону ответственности в составлении и участии в мероприятиях квизов:
  + **startgame** – команда для того, чтобы начать интерактивный квиз в игровой форме;
  + **startreg** – команда для начала регистрации на участие в интерактивном тесте-квизе игрового формата;
  + **endreg** – команда завершения регистрации на участие в игровом тесте, не позволяя другим пользователям сервера приложения Discord и продвинутого бота-ассистента “Mugiwara” присоединиться к текущему проводимому квизу;
  + **question** – команда отображения и показа следующего вопроса текущего интерактивного квиза в красивом оформлении с помощью средств библиотеки dhooks;
  + **result** – команда отображение и показа результатов последнего заданного вопроса из текущего интерактивного теста;
  + **endgame** – команда завершения текущей сессии интерактивного учебного квиза, проводимого в игровом формате, и отображения интерактивной таблицы лидеров-участников данной завершенной сессии теста-квиза в красивом оформлении с помощью средств dhooks;
  + **checkhost** – команда проверки наличия у игрока роли организатора, составителя и ведущего тривиальных учебных квизов в игровой форме;
  + **checkgame** – команда проверки активности и статуса игрового интерактивного квиза-теста знаний в текущий момент времени данного сервера приложения платформы Discord;
  + **checkplayers** – команда просмотра того, какие пользователи-ученики принимают участие в грядущем интерактивном квизе-тесте знаний в игровой форме;
  + **join** – команда для присоединения к участникам текущей сессии теста и принятии участия в данном проводимом учебном квизе в игровом формате;
  + **time** – команда получения информации об оставшемся времени, отведенного на показ и сдачу ответа на текущий вопрос.
* Возможность голосования в кастомных созданных на сервере приложения Discord опросниках с помощью реакций-эмодзи цифр и интерактивное отображение заполнения и результатов по мере проведения голосования. Также в связи с данным функциональным требованием реализуется команда создания опросников голосования **poll** с кастомизированным количеством минут времени, отведенного на опрос, количеством максимально возможного голосов для варианта ответа в опросе, текстовые варианты ответов.
* Организация продвинутой модерации серверов Discord с помощью умного детектирования запрещенных слов и публикации ссылок, удаления сообщения и выдачи предупреждения за определенное нарушение правил сервера и при превышении 5 нарушений выдача бана пользователю с последующим киком-удалением с данного сервера. Команды ручной модерации сервера:
  + **unwarn** – команда снятия и удаления одного предупреждения у выбранного пользователя сервера, при отсутствии предупреждений или ошибки в написании команды/поиска пользователя будет выведено сообщение, соответствующее содержанию и смыслу недочета в использовании данной команды;
  + **warn** – команда выдачи одного предупреждения у выбранного пользователя с указанием причины выдачи его такими, как BADWORDS или LINKS, при ошибке написания команды выводится сообщение об ошибке пользователю и повторе запроса с корректным написанием команды. При получении максимально допустимого количества предупреждений пользователь получает бан на данном сервере Discord и кикается с него до ручного снятия бана модераторами или владельцем сервера;
  + **clear\_warns** – команда снятия всех выданных ранее предупреждений у выбранного пользователя, при ошибке написания команды или отсутствии предупреждений у выбранного пользователя на сервере выводится соответствующее сообщение с содержанием недочетов в использовании данной команды продвинутого бота-ассистента “Mugiwara”.
* Организация возможности перевода текстовых сообщений при помощи использования команды translate с указанием нужного языка и текста для перевода. При этом обязательно текст на перевод не является пустым или символом пробела, при несоблюдении данного требования гарантированно выкидывается исключение, которое вызывает срабатывание средств защиты и отправляет сообщение об ошибке. При запросе перевода на неизвестный для бота-ассистента “Mugiwara” язык выкидывается соответствующее содержанию исключение, которое перехватывается обработчиком, после чего отправляется сообщение в тестовый канал приложения-платформы Discord с описанием ошибки и требованием повтора запроса с корректным написанием языка или его кода. Также будет присутствовать возможность перевода текстовых сообщений с помощью реакций-эмодзи флагов, известных для продвинутого бота-ассистента “Mugiwara” в виде кодов языков, сопоставленных с доступными для использования на сервере эмодзи флагов различных известных стран, при отсутствии известного для бота флага страны или языка перевод не будет произведен без нарушения и функционирования работы бота.
* Проигрывание музыки, подкастов и видео с онлайн платформы видеоконтента YouTube с использованием библиотек язык Python такими, как youtubesearchpython, ffmpeg и youtube-dl, и библиотеки взаимодействия интернет http-протоколов с приложением платформой Discord со схожим названием discord. Команды продвинутого бота-ассистента “Mugiwara” для организации работы с платформой видеоконтента YouTube:
  + **play** – команда для проигрывания выбранного видео платформы YouTube, доступного по Интернет-ссылке. При первом использовании команд play бот-ассистент проверяет наличие пользователи в голосовом канале сервера, после чего подключается к нему и начинает проигрывать песню, при повторном использовании команды бот добавляет видео в очередь на проигрывание. При отсутствии пользователя в голосовом канале бот-ассистент “Mugiwara” выдает соответствующее сообщение об ошибке;
  + **queue** – команда отображения и показа текущего списка видеоряда в очереди на воспроизведение в голосовом канале сервера, при отсутствии очереди бот отображает сообщение о пустой очереди и отсутствия видеоряда для воспроизведения;
  + **skip** – команда прерывания проигрывания текущего воспроизводимого видео в голосовом канале сервера и переход к следующему в очереди, при отсутствии и незаполненности очереди для воспроизведения бот перестает проигрывать видео в голосовом канале;
  + **pause** – команда приостанавливания воспроизведения ботом-ассистентом текущего видео в очереди в голосовом канале сервера;
  + **resume** – команда продолжения и возобновления воспроизведения текущего по очереди видео в голосовом канале сервера с помощью продвинутого бота-ассистента “Mugiwara”;
  + **remove** – команда удаления последнего добавленного в очередь на воспроизведение видео;
  + **clear** – команда остановки воспроизведения текущего видео в очереди и полное очищение всей очереди воспроизведения видеоряда с платформы YouTube;
  + **leave** – команда изгнания и удаления бота из голосового канала сервера Discord.
* Возможность использования команды вызова интеллектуального агента-ассистента ChatGPT компании OpenAI для отправки запросов и получения ответов на вопросы учеников сервера Discord и пользователей продвинутого бота-ассистента “Mugiwara” в любой момент времени и суток, когда преподаватель не сможет ответить точно на поставленный вопрос или будет отсутствовать в сети.
* Использование команды **help** для получения справочной информации о всех доступных командах продвинутого бота-ассистента “Mugiwara”, корректном написании каждой команды и их аргументов, информации о зоне ответственности данной команды и описания того, что делает каждая из доступных команд для использования и какие роли пользователей требуются для вызова определенной команды бота-ассистента.
  1. **Описание и обоснование архитектуры программы**
     1. **Описание архитектуры программы**

Для наиболее эффективного использования функционала бота-ассистента был выбран и использован устаревший и нестандартный для пользования на практике шаблон “Система с подсистемами”, который состоит из главной системы с подключенными к ней подсистемами и пакетов с вспомогательными файлами функций для помощи работы команд бота-ассистента и хранилищами информации.

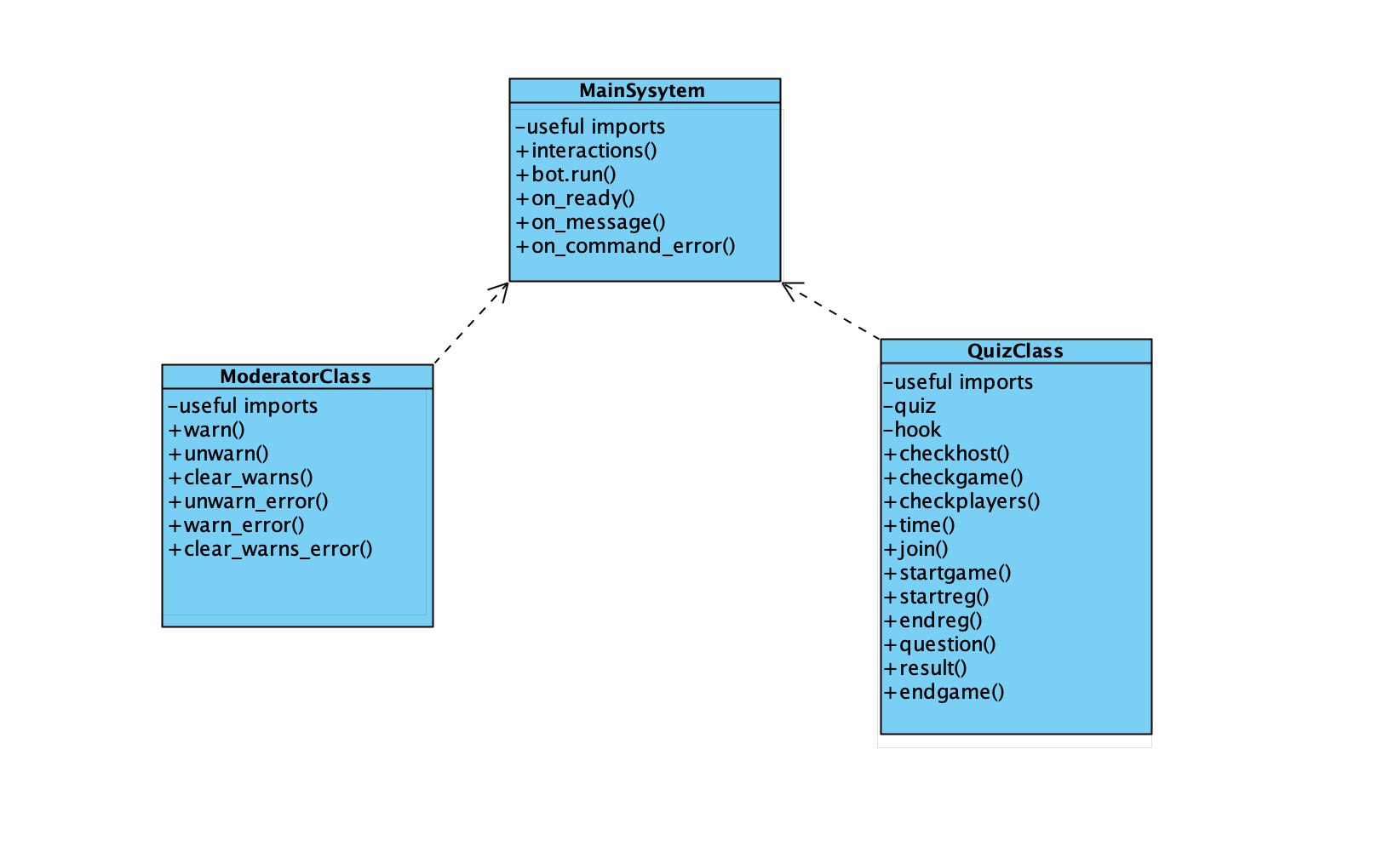
Архитектура приложения продвинутого бота-ассистента “Mugiwara” представлена на рисунке-диаграмме, расположенной ниже.

**Общий вид системы**



Программа состоит из главной системы **MainSystem** и подсистем в виде модулей-пакетов: **GPTModule**, **voteModule**, **specialFiles**, **Cogs**, **QuizAttributes**.

Каждый из модульных пакетов-подсистем отвечает за свою область ответственности и отслеживает корректную работу своей подсистемы и своего предопределенного файлом функционала команд.

**“MainSystem”** – данная главная система, расположенная внутри файла main.py, осуществляет подключение и связь всех подсистем, когов, модулей, вспомогательных файлов и выстраивание, взаимосвязь всех зависимостей и файлов системы и подсистем между собой. Обработка сообщений и ошибок использования команд происходит и осуществляется тут же, в данной системе. Система включает в себя два класса: Moderator и Quiz. Каждый из них снабжен своим функционалом и обработчиком ошибок внутри класса своей области ответственности.

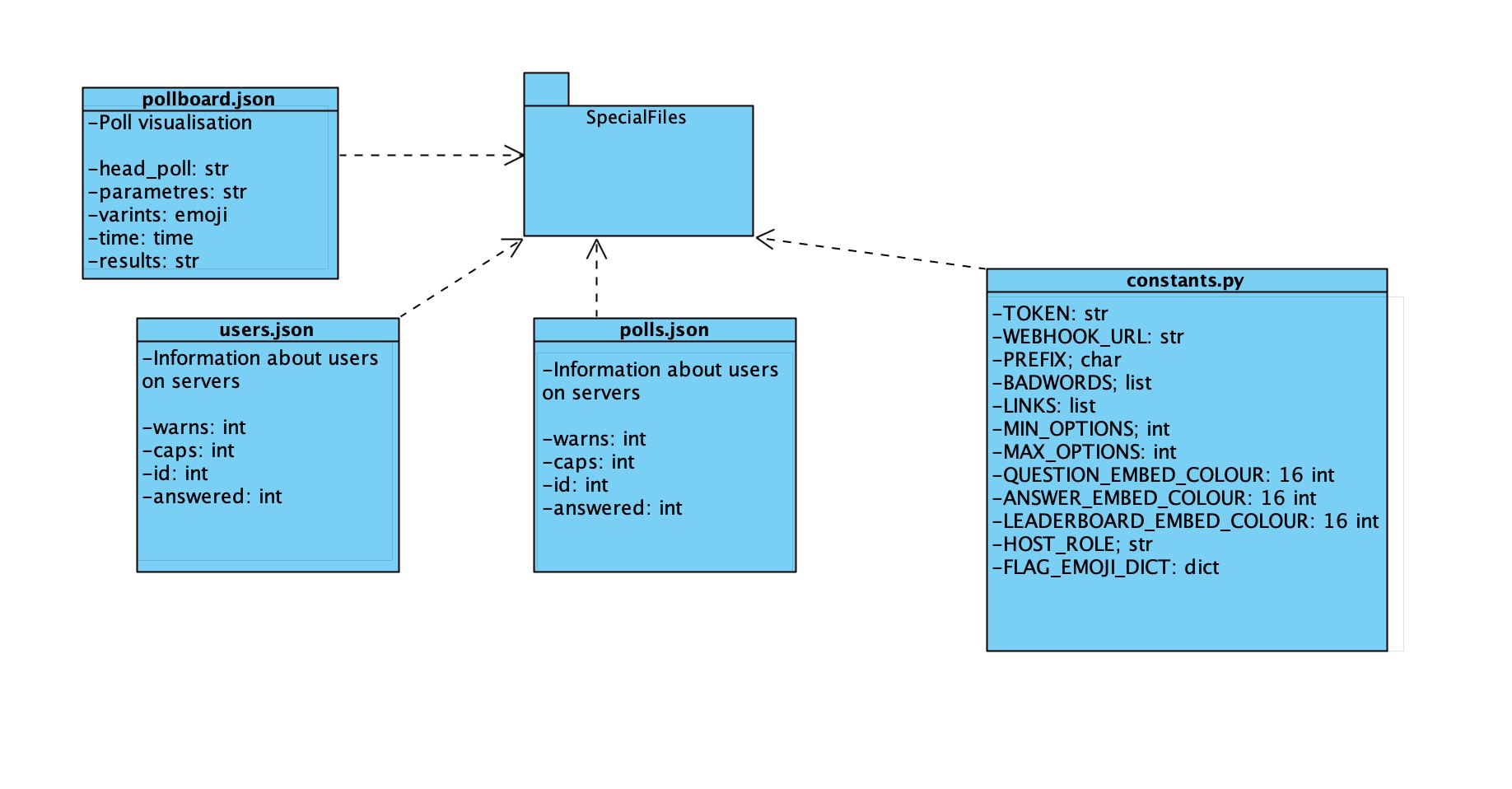
**ModeratorClass:**

* **unwarn** – команда снятия и удаления одного предупреждения у выбранного пользователя сервера;
* **warn** – команда выдачи одного предупреждения у выбранного пользователя с указанием причины выдачи его такими, как BADWORDS или LINKS;
* **clear\_warns** – команда снятия всех выданных ранее предупреждений у выбранного пользователя;

**QuizClass:**

* **startgame** – команда для того, чтобы начать интерактивный квиз в игровой форме;
  + **startreg** – команда для начала регистрации на участие в интерактивном тесте-квизе игрового формата;
  + **endreg** – команда завершения регистрации на участие в игровом тесте, не позволяя другим пользователям сервера приложения Discord и продвинутого бота-ассистента “Mugiwara” присоединиться к текущему проводимому квизу;
  + **question** – команда отображения и показа следующего вопроса текущего интерактивного квиза в красивом оформлении с помощью средств библиотеки dhooks;
  + **result** – команда отображение и показа результатов последнего заданного вопроса из текущего интерактивного теста;
  + **endgame** – команда завершения текущей сессии интерактивного учебного квиза, проводимого в игровом формате, и отображения интерактивной таблицы лидеров-участников данной завершенной сессии теста-квиза в красивом оформлении с помощью средств dhooks;
  + **checkhost** – команда проверки наличия у игрока роли организатора, составителя и ведущего тривиальных учебных квизов в игровой форме;
  + **checkgame** – команда проверки активности и статуса игрового интерактивного квиза-теста знаний в текущий момент времени данного сервера приложения платформы Discord;
  + **checkplayers** – команда просмотра того, какие пользователи-ученики принимают участие в грядущем интерактивном квизе-тесте знаний в игровой форме;
  + **join** – команда для присоединения к участникам текущей сессии теста и принятии участия в данном проводимом учебном квизе в игровом формате;
  + **time** – команда получения информации об оставшемся времени, отведенного на показ и сдачу ответа на текущий вопрос

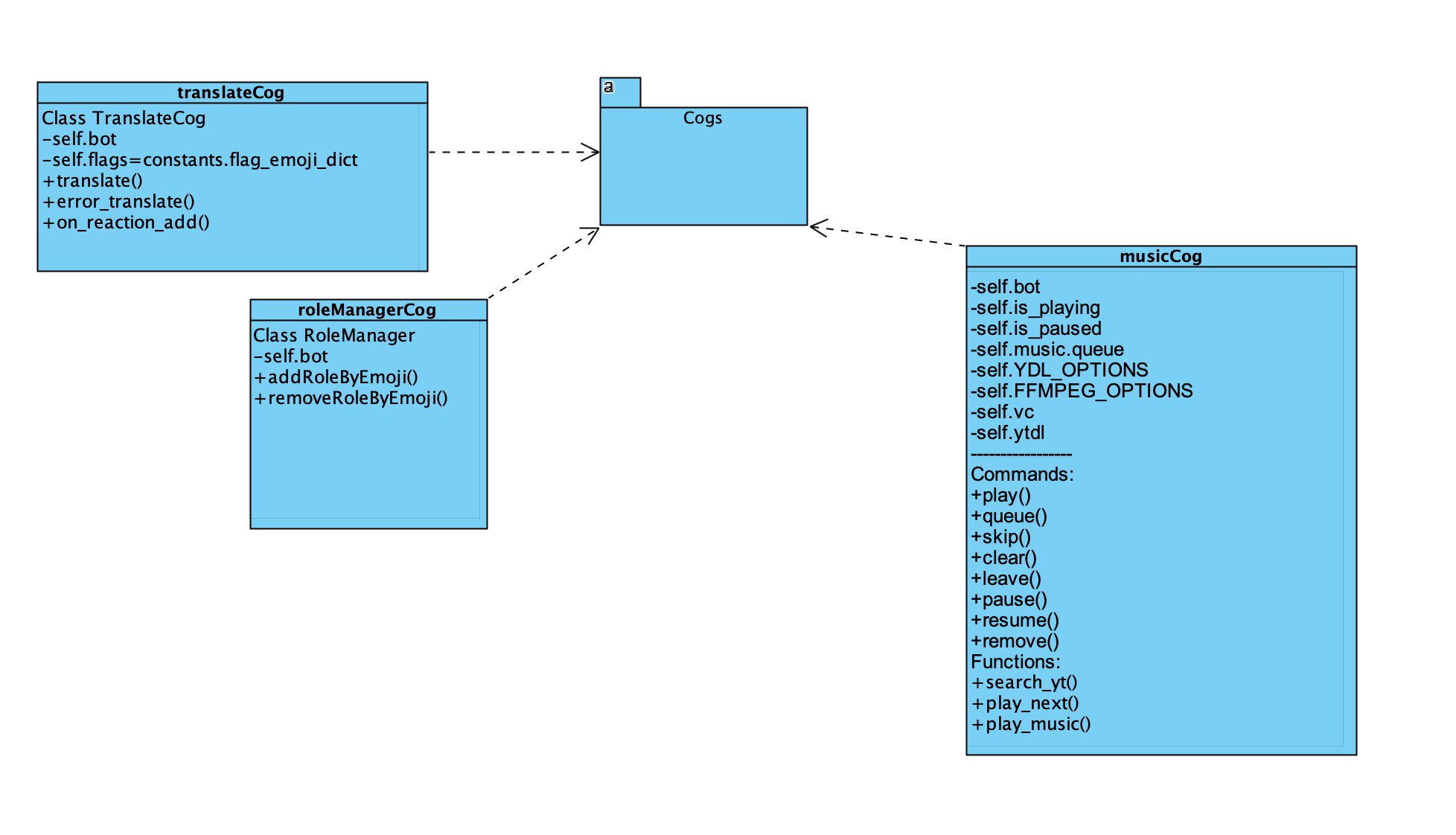
**“SpecialFiles”** – осуществляет хранение файлов, ответственных за организацию и проведение опросов, хранение постоянных переменных/констант, информации о пользователях и их статистики на серверах с ботом-ассистентом “Mugiwara”.



**SpecialFiles**:

* **users.json:** все данные о пользователях и их статистики с характеристиками хранятся в данном файле расширения json и подтягиваются в самого бота программы во время его работы и функционирования на серверах;
* **pollboard.json:** отвечает за красивое оформление и заполнение опросника с голосованием;
* **polls.json:** хранение информации об опросниках, его параметрах и модификациях завершения работы и очищения;
* **constants.py:** хранение всех константных значений, используемых в работе бота-ассистента, в том числе и во время организации и проведения тестов/игровых квизов, что является главной функцией “Mugiwara”.

**“Cogs”** – осуществляет взаимодействие когов функционала команд Discord бота, ответственных за команды проигрывания видео с платформы YouTube, перевода текстовых сообщений, организации опросов в виде интерактивных голосовалок с красивым оформлением и заполнением и менеджмент ролей, их выдачу и снятие.



**MusicCog:**

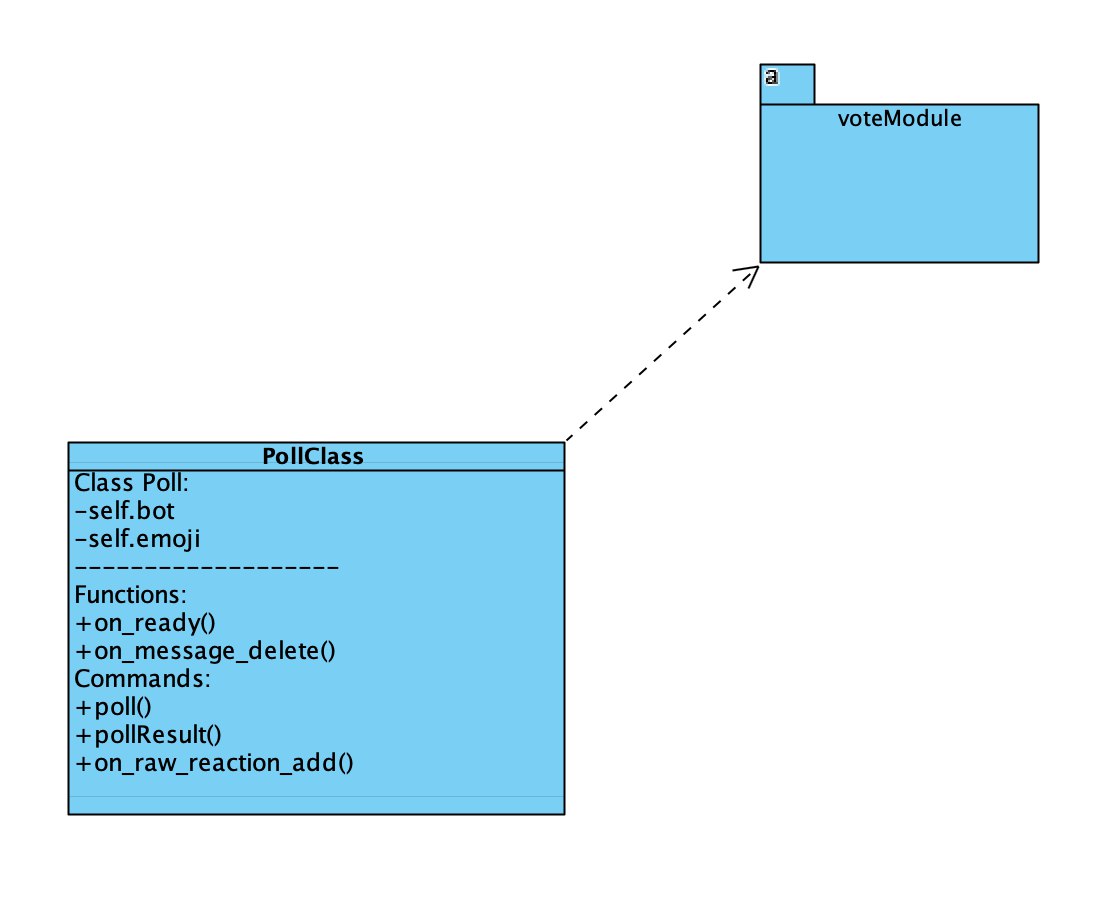
* **clear** - команда остановки воспроизведения текущего видео в очереди и полное очищение всей очереди воспроизведения видеоряда с платформы YouTube;
* **leave** - команда изгнания и удаления бота из голосового канала сервера Discord;
* **pause** - команда приостанавливания воспроизведения ботом-ассистентом текущего видео в очереди в голосовом канале сервера;
* **play** - команда для проигрывания выбранного видео платформы YouTube, доступного по Интернет-ссылке;
* **queue** - команда отображения и показа текущего списка видеоряда в очереди на воспроизведение в голосовом канале сервера;
* **remove** - команда удаления последнего добавленного в очередь на воспроизведение видео;
* **resume** - команда продолжения и возобновления воспроизведения текущего по очереди видео в голосовом канале сервера;
* **skip** - команда прерывания проигрывания текущего воспроизводимого видео в голосовом канале сервера и переход к следующему в очереди.

**TranslateCog:**

* **translate** - команда для перевода текста на определенный язык, после чего отобразится перевод, произношение, точность перевода и с какого языка переведено
* **translate\_error** - обработчик ошибок при работе с командой translate
* **on\_reaction\_add** - при добавлении флага реакции на сообщение отображается информация о переводе схожая с командой translate

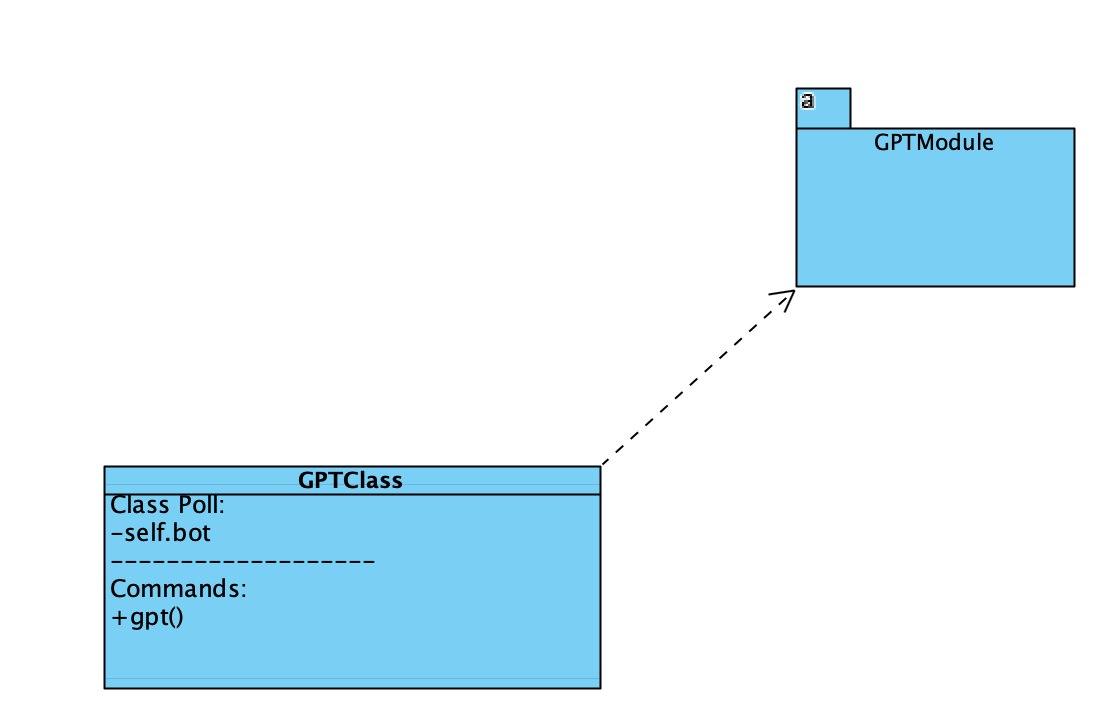
**RoleManagerCog:**

* **addRoleByEmoji** - добавление роли с помощью кастомизированной уникальной эмоции/эмодзи на сообщение бота
* **removeRoleByEmoji** - убрать роль с помощью снятия кастомизированной уникальной эмоции/эмодзи с сообщения бота

****

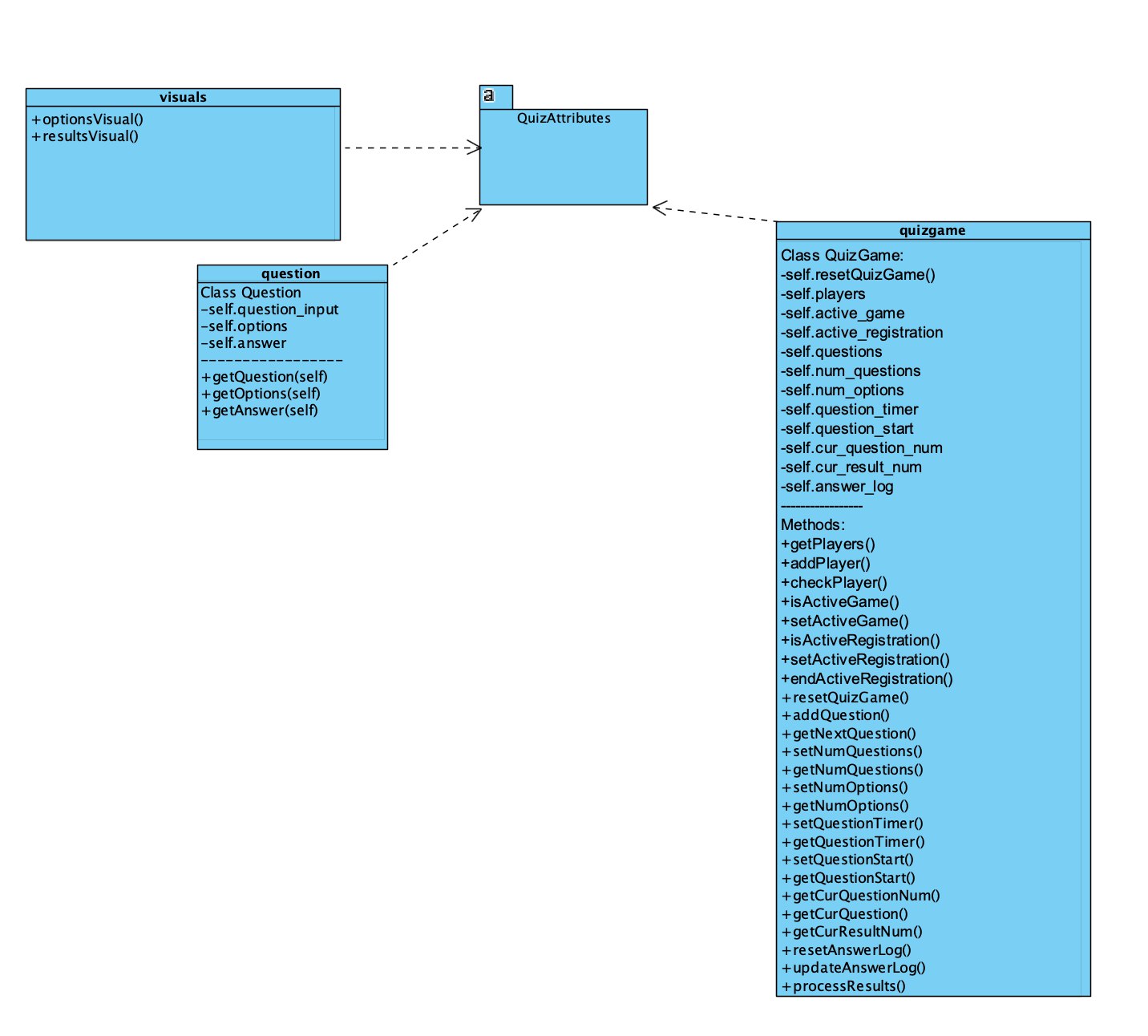
**PollClass aka PollCog:**

* **on\_raw\_reaction\_add** - команда добавления реакции для голосования за определенный пункт нашего бюллетеня в канале Discord;
* **pollResult** - команда отображения результатов голосования;
* **poll** - команда организации голосования с вариантами ответов и временем, отведенным на голосование.

****

**GPTClass aka GPTCog:**

* **gpt** - команда вызова интеллектуального агента-ассистента ChatGPT компании OpenAI для отправки запросов и получения ответов на вопросы учеников сервера Discord и пользователей продвинутого бота-ассистента “Mugiwara”.



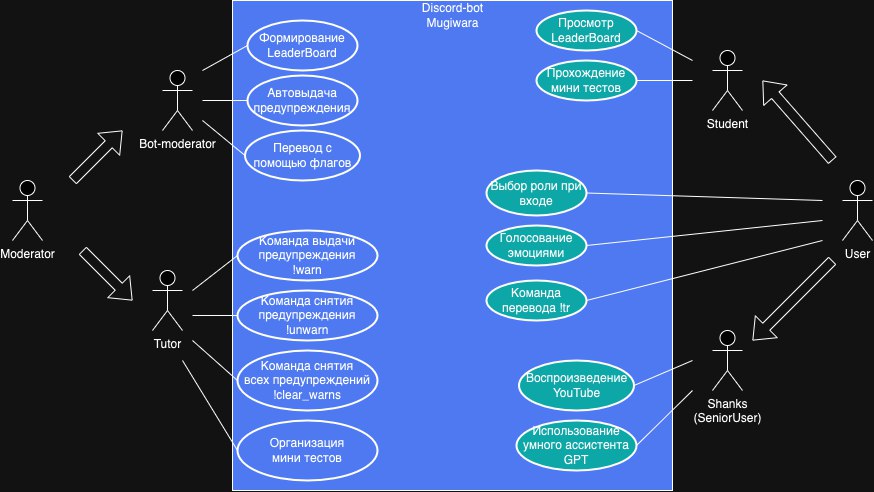
**“QuizAttributes”** – данный пакет с модульными файлами содержит внутри себя три файла с описанием классов **question**, **quizgame**, **visuals**, каждый из которых используется в главной системе **MainSystem** для организации и проведения квизов.

* + 1. **Обоснование архитектуры программы**

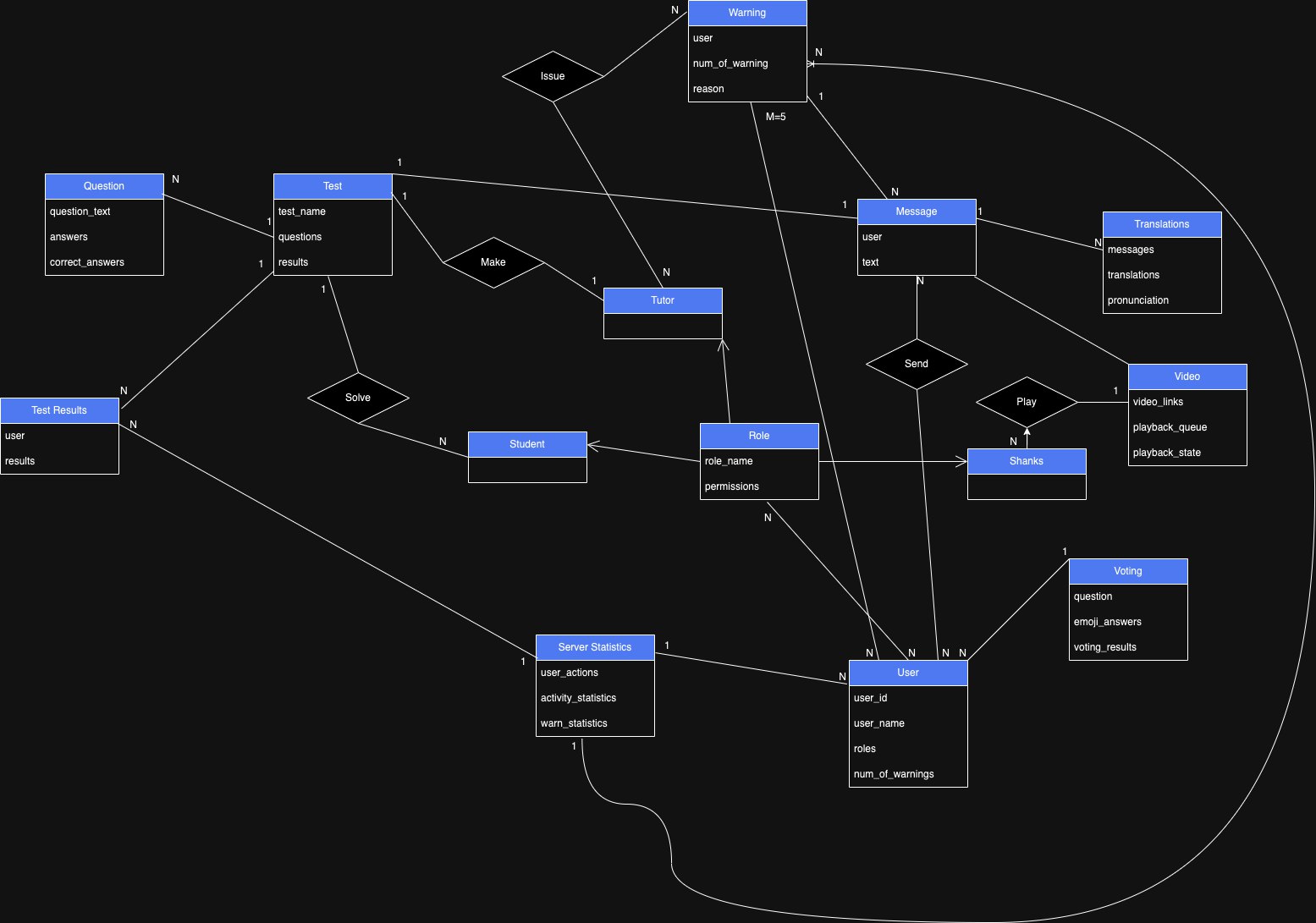
Шаблон “Система с подсистемами” был выбран для создания и проектирования моего разрабатываемого программного продукта «Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий» в связи с его удобством реализации, простотой взаимодействия главной системы с ее подсистемами и практичностью для взаимодействия с приложением Discord.

Данный шаблон также является единственным пригодным для использования и разработки бота-ассистента, поскольку обеспечивает полное покрытие функционала, необходимого для интерактивного и максимально полезного использования всех возможностей для эффективного обучения и взаимодействия на серверах платформы Discord.

**Модель прецендентов:** на ней отображена корректная проработка взаимодействия пользователей с доступными командами бота “Mugiwara”, которая стала возможна благодаря выбору шаблона “Система с подсистемами” и детальному продуманному подходу к проектированию данной разработки.



**Модель предметной области:** на ней отображены взаимодействия сущностей, их атрибуты и связи. Исходя из нее, осуществлялась и проводилась дальнейшая разработка программного продукта 3 курса обучения.



Пояснения:

Сущности и их атрибуты представлены следующим образом:

1. **Пользователь (User):** имя пользователя (**user\_name**), уникальный идентификатор (**user\_id**), роли (**roles**), количество предупреждений (**num\_of\_warnings**).
2. **Роль (Role):** название роли (**role\_name**), разрешения (**permissions**). Роль подразделена на 3 типа, указанных в модели.
3. **Тест (Test):** название теста (**test\_name**), вопросы (**questions**), результаты (**results**)
4. **Вопрос (Question):** текст вопроса (**question\_text**), варианты ответов (**answers**), правильный ответ (**correct\_answer**).
5. **Результаты теста (Test Results):** пользователи (**users**), результаты (**results**).
6. **Голосование (Voting):** вопрос (**question**), варианты ответов (**answers**), результаты голосования (**voting\_results**).
7. **Статистика сервера (Server Statistics):** действия пользователей (**user\_actions**), статистика активности (**activity\_statistics**), статистика предупреждений (**warn\_statistics**).
8. **Предупреждение (Warning):** пользователь (**user**), количество предупреждений (**number\_of\_warnings**), причина (**reason**).
9. **Переводы (Translations):** сообщения (**messages**), переводы (**translations**), произношение (**pronunciation**).
10. **Видео (Video):** ссылки на видео (**video\_links**), очередь воспроизведения (**playback\_queue**), состояние воспроизведения (**playback\_state**).
11. **Сообщение (Message):** пользователь (**user**), текст (**text**).

Взаимоотношения между сущностями выглядят следующим образом:

**“Пользователь”** связан с “**Ролью”** (многие-ко-многим): Пользователь может иметь несколько ролей, а роль может быть назначена нескольким пользователям.

**“Тест”** связан с “**Вопросом”** (один-ко-многим): Тест может содержать несколько вопросов, но вопрос принадлежит только одному тесту.

**“Тест”** связан с **“Результатами теста”** (один-ко-многим): Тест может иметь несколько результатов, но результат теста относится только к одному тесту.

**“Голосование”** связано с **“Пользователем”** (многие-к-одному): Событие голосования связано с несколькими пользователями, но один пользователь может участвовать в нескольких событиях голосования.

**“Статистика сервера”** связана с **“Пользователем”** (многие-к-одному): Статистика сервера связана с несколькими пользователями, но у одного пользователя может быть несколько статистик сервера.

**“Предупреждение”** связано с **“Пользователем”** (многие-ко-многим M=5): Предупреждение связано с несколькими пользователями, у одного пользователя может быть несколько предупреждений.

**“Переводы”** связаны с **“Сообщением”** (один-ко-многим): Переводов может быть несколькими на разных языках, в зависимости от контекста.

**“Видео”** связано с **“Продвинутым пользователем”** (один-ко-многим): Видео может быть связано с несколькими командами и продвинутыми пользователями, но команда или продвинутый пользователь могут быть связаны только с одним видео.

**“Тест”** связан с **“Сообщением”** (один-к-одному): в одном сообщение содержится одно описание теста.

**“Тест”** связан с **“Студентом”** (один-ко-многим): несколько студентов могут одновременно решать один тест, нельзя одновременно два и более тестов решать.

**“Тест”** связан с **“Преподавателем”** (один-к-одному): тест может создаваться только одним организатором одновременно, организатор за раз может создать один тест.

**“Серверная статистика”** связана с **“Предупреждениями”** (один-ко-многим): серверная статистика может содержать информацию одновременно о нескольких предупреждениях каждого пользователя, но предупреждения содержатся в единой статистике.

* 1. **Описание и обоснование алгоритма работы программы**
     1. **Описание алгоритма работы программы**

С момента запуска бота-ассистента на локальном сервере компьютера или удаленном хостинге устанавливается соединение с приложением Discord, происходит так называемый “обмен рукопожатий” (handshake), после чего отображается номер сессии и время подключения бота-ассистента для работы на серверах Discord. Затем подключаются все библиотеки, обеспечивающие работу приложение бота-ассистента, и программные модульные файлы Cogs для обеспечения работы функциональности всех команд бота-ассистента “Mugiwara”, заполняются данные об участниках-пользователях сервера и бота-ассистента в файлы users.json, После чего бот готов и доступен к использованию, работе на установленных серверах и эффективному выполнению всех прописанных в своем функционале команд.

Подробная работа алгоритма взаимодействия Discord бота “Mugiwara” и выполнение функциональных команд ассистента были зафиксированы на видеозаписи с помощью средства OBS Studio. Это было сделано на случай непредвиденных обстоятельств на защите у комиссии.

* + 1. **Обоснование алгоритма работы программы**

Выбранный алгоритм работы и выполнения функциональных команд бота-ассистента был использован по причине его простоты использования и разработки программного кода через библиотеку создания Discord ботов discord.py, которая сделала возможным написание данного продвинутого бота-ассистента и помогла максимально эффективно организовать работу всех команд и их исполнение. Использование библиотеки dhooks обусловлено простотой ее использования и гибкостью оформления красивого интерфейса внутри приложения Discord для отображения вопросов учебных квизов. Библиотека JSON была выбрана в связи с удобством ее использования и хранения данных в компактном формате. Библиотеки openai, yt\_dlp, youtubesearchpython, datetime, googletrans были выбраны, исходя из исключительности их удобного, легко адаптируемого функционала для организации корректной и эффективной работы выполнения команд бота-ассистента и гибкости их внедрения в программный код бота “Mugiwara”. Данные, содержащиеся в JSON файлах и хранящие информацию о пользователях и их статистику, были получены в результате корректного описания работы и поведения бота-ассистента “Mugiwara” с помощью тестирования работы функционала.

* 1. **Описание и обоснование выбора метода организации входных и выходных данных**

В качестве данных, получаемыми на вход ботом-ассистентом, используются:

* Текстовые сообщения, отправляемые во всех текстовых каналах сервера приложения-платформы Discord, в том числе и ответы на вопросы квизов;
* Доступные и знакомые боту для исполнения команды, прописанные в функционале “Mugiwara”.

Данными, которые выдает бот-ассистент на выход после выполнения команд, являются:

* Результаты работы команд бота и их отображение в текстовом формате или визуализированном с помощью технологии dhooks встроенном средстве discord.Embed;
* Аудио-трансляция видеопотока с платформы видеохостинга YouTube в голосовой канал сервера Discord, в котором сидит пользователь, запросивший выполнение команды проигрывания;
* Текстовые сообщения со статистикой и информацией о пользователях, количестве их предупреждений и причин выдачи;
* Результаты тестов и опросов.

Данный выбор метода организации данных, получаемыми на вход и выход, был сделан в связи с самой темой разработки Discord бота-ассистента и исходя из технического задания и требуемого функционала в нем, поскольку именно такая организация данных поможет раскрыть весь потенциал бота “Mugiwara” в сфере образования и предоставит преподавателям возможность максимально эффективно организовать продуктивное онлайн-обучение учеников на серверах платформы Discord.

* 1. **Описание и обоснование выбора состава технических и программных средств**
     1. **Состав технических и программных средств**

Для нормальной работы кода программы продвинутого бота-ассистента “Mugiwara” предъявляются данные требования:

1. 100 МБ свободного дискового пространства;
2. 256 МБ на хосте бота для хранения всех данных о пользователях на сервере;
3. Доступ к сети Интернет для передачи и обработки запросов;
4. Доступ к VPN и стабильное соединение с ним;
5. Доступ к библиотекам, использующихся внутри Discord бота для выполнения команд, и соответствие новым версиям http-запросов;
6. Поддержка хостинга бота и дальнейшего развития программных средств.

Для работы программы после проведенных испытаний и стресс-тестов бота-ассистента был выявлен следующий состав программных средств, необходимых для успешной, качественной и беспрерывной работы продвинутого бота “Mugiwara”, обеспечивающее максимально эффективное использование на серверах приложения-платформы Discord:

1. macOS Ventura 13.3.1, OC Windows 10;
2. Библиотека discord.py для взаимодействия с приложением Discord;
3. Совместимость с библиотеками dhooks, json, time, asyncio, openai, yt\_dlp, youtubesearchpython, datetime, googletrans для работы команд бота-ассистента;
4. Поддержка JSON файлов.
   * 1. **Обоснование выбора технических и программных средств**

При несоответствии требований программных средств и отключении библиотек, задействованных для работы бота-ассистента, код может некорректно обрабатывать часть команд, в результате чего программа будет выдавать ошибки с их подробным описанием на сторону разработчика, а в случае критической ошибки бот завершится аварийно и отключится от сети. При иных ошибках, связанных с некорректным написанием команд или отсутствием прав, приложение обработает соответствующие исключения, отправит пользователю соответствующее сообщение о допущенной ошибки и продолжит свое функционирование с обработкой команд в штатном режиме.

Для разработки программного продукта «Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий» был сделан выбор в пользу языка программирования высокого уровня Python с использованием библиотек discord, dhooks, json, time, asyncio, openai, yt\_dlp, youtubesearchpython, datetime и googletrans, так как это является максимально эффективным решением, наиболее полно покрывающим весь требуемый функционал “Mugiwara” и дающим возможность к его дальнейшему расширению и модификации. В качестве хранилища данных был сделан выбор в сторону файлов формата JSON в связи удобным хранением информации, быстрым доступом к данным в любой момент времени и гибкостью редактирования, записи и обновления информации внутри них. При дальнейшей разработке и добавлении нового функционала и увеличении количества пользователей на сервере система бота-ассистента “Mugiwara” будет расширена с помощью базы данных, использующей СУБД, как MySQL или PostgreSQL

1. **ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**
   1. **Ориентировочная экономическая эффективность**

Программный проект Discord бот-ассистент “Mugiwara” разрабатывался в рамках курсовой работы для ОП "Программная инженерия" на 3 курсе обучения, в будущем при дальнейшей доработке и расширении функционала продвинутого бота-ассистента “Mugiwara” он будет использован на серверах приложения Discord такие, как “Школково”, “/r/OnePiece” и “Академия анимации и мультипликации JamClub”, ориентирован на получение прибыли с поддержания хостинга бота-ассистента, поддержки работы бота и расширения в дальнейшем функционала на установленных серверах, а также получение комиссии с покупки токенов ChatGPT и их использования для отправки запросов, получения ответов от умного ассистента OpenAI. Цель экономической эффективности будет достигаться благодаря материальной поддержки и заинтересованности в данном боте-ассистенте со стороны Discord серверов, нуждающихся в продвинутом, уникальном и прорывном функционале “Mugiwara”, разработанного с использованием средств таких библиотек, как discord, dhooks, json, time, asyncio, openai, yt\_dlp, youtubesearchpython, datetime, googletrans, и языка Python.

* 1. **Предполагаемая потребность**

«Discord бот-ассистент “Mugiwara” как инструмент для проведения онлайн занятий» будет интересен различным серверам, расположенным на платформе приложения Discord, чтобы организовывать и проводить онлайн-занятия в максимально комфортном для учеников месте. Также данный бот-ассистент будет интересен серверам, имеющими необходимость в упрощении организации работы рутинных задач и их автоматизации с помощью технологий и инструментов, которые предоставляет приложение Discord для удобства пользования. Помимо этого, благодаря гибкой настройке и легко адаптируемому функционалу продвинутого бота-ассистента “Mugiwara” может быть использован для любых уникальных нужд серверов Discord, включая развлекательную сферу. Таким образом, “Mugiwara” поможет в автоматизации процесса модерации серверов, организации и проведении мини тестов/учебных квизов для проверки знаний, сможет проигрывать видео с платформы видеохостинга YouTube и работать эффективно с умным ассистентом OpenAI ChatGPT, и это далеко не весь потенциал данного бота-ассистента.

* 1. **Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами**

Пользуясь поиском по сайту c Discord ботами, по российским и зарубежным Интернет сервисам поиска и проводя опросы среди друзей и знакомых, были найдены и отбраны такие альтернативы боту-ассистенту «Mugiwara» с частично схожим набором функционала:

1. Dyno Bot:

Dyno Bot – это настраиваемый Discord-бот для модерации сервера Discord с простой и интуитивно понятной веб-панелью. Его функционал характеризуется данным набором ответственности команд:

* Выдача ролей при входе на сервер
* Проверка знаний с помощью тестов
* Возможность голосования с помощью эмоций
* Помощь модерации сервера и статистики

Данное решение позволяет владельцам серверов легко и быстро настроить данного бота под свои нужды. Однако, программный продукт «Discord бот-ассистент Mugiwara» имеет ряд преимуществ по сравнению с данным ботом:

* 1. Полноценная настройка тестов и команд: “Mugiwara” предоставляет возможность создать более подробные и настраиваемые тесты, добавив свои собственные команды и форматы вопросов, в отличие от Dyno Bot.
  2. Кастомизируемые опросники: Бот “Mugiwara” также предоставляет возможность создания голосований с помощью эмоций, что позволяет пользователям создавать опросы, настроенные под свои потребности и предпочтения с выставлением любого количества минут, данных на выбор варианта ответа, до завершения опроса, в то время как Dyno Bot дает создавать голосование строго на 5 минут.
  3. Расширенные возможности модерации: Бот Mugiwara имеет собственную интеллектуальную модель отбора запрещенных слов и публикации ссылок, что облегчает процесс модерации сервера и сбора статистики. Команды ручной модерации также помогут вам контролировать и управлять поведением участников на сервере. В то время как DynoBot предоставляет возможность только ручной модерации.

1. Tatsumaki Bot:

Tatsumaki Bot – это настраиваемый Discord-бот, названный в честь аниме героини из сериала «OnePunchMan», которые обладает частично схожим с ботом-ассистентом «Mugiwara» набором функционала:

* Выдача ролей при входе на сервер
* Возможность голосования с помощью эмоций
* Помощь модерации сервера и статистики
* Перевод сообщений

Данный бот предусмотрел многие функциональные характеристик у других ботов платформы Discord и позволил ускорить работу проведения модерации с помощью специальных расширенных команд модерации. Несмотря на это, «Discord бот-ассистент Mugiwara» выигрывает на его фоне по ряду плюсов в использовании:

* 1. Перевод сообщений: “Mugiwara” предлагает удобную команду “translate” для перевода сообщений с расширенными разделами: перевод, язык, с которого переведено, и точность перевода, а также произношение при наличии и сам текст на перевод. Также предоставляет возможность перевода сообщений с помощью реакций эмодзи флагов. В отличие от “Mugiwara”, Tatsumaki Bot позволяет переводить сообщения только с помощью команды и на выход подается просто перевод без определения языка, точности перевода и произношения. Также в Tatsumaki Bot отсутствует возможность перевода с помощью эмодзи реакций.
  2. Выдача ролей при входе на сервер: “Mugiwara” делает выдачу ролей более интерактивной и интересной, при это позволяя не только выдавать роли по выставлению эмодзи, но и снимать также роли при снятии эмодзи с сообщения.

1. Rhythm Bot:

Rhythm Bot – это один из лучших ботов приложения Discord и лучший музыкальный бот этого приложения, который обладает полноценным функционалом диджея и проигрывания музыки в голосовом канале сервера Discord.

Хоть Rhythm Bot и обладает более обширным музыкальным функционалом по сравнению с ботом-ассистентом “Mugiwara”, но на это весь его функционал заканчивается. Из-за отсутствия иного функционала, помимо музыкального, он остается неконкурентоспособным по сравнению с “Mugiwara” и его функционалом.

Общее выявленное преимущество Discord бота-ассистента «Mugiwara» по сравнению с тремя рассмотренными аналогами-конкурентами:

* Использование продвинутого ассистента-помощника ChatGPT от компании OpenAI: в отличие от трех представленных аналогов-конкурентов, Discord-бот “Mugiwara” интегрирует в свой функционал умного ассистента ChatGPT, что позволяет пользователям общаться и задавать вопросы этому интеллектуальному ассистенту для получения дополнительной информации об интересующем вопросе и помощи по учебе в ситуациях, когда преподаватель занят или не может ответить на поставленный вопрос. Это несравненное преимущество, позволившее конкурировать с аналогами и опередить их.

1. **ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**
2. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. Документация библиотеки discord.py <https://discordpy.readthedocs.io/en/stable/>
13. Сайт разработчиков Discord <https://discord.com/developers/docs/intro>
14. Документация библиотеки ffmpeg <https://ffmpeg.org/ffmpeg.html>
15. Документация библиотеки youtube-dl для взаимодействия с видеохостингом платформы YouTube <https://youtube-dl.readthedocs.io/en/latest/>
16. Документация разработчиков библиотеки-переводчика googletrans <https://pypi.org/project/googletrans/>

# 

**ПРИЛОЖЕНИЕ 1**

**ПОКАЗ И ПОЯСНЕНИЕ НАЗНАЧЕНИЯ ФУНКЦИЙ-КОМАНД В ПРОГРАММНОМ КОДЕ**

|  |  |
| --- | --- |
| **Название** | **Описание данной части** |
| def on\_message(message) | Функция, которая реализует команду автоматической модерации сообщений благодаря лично созданной технологии фильтра сообщений и помогает в приеме ответов на игровые квизы. |
| def on\_command\_error(ctx, error) | Функция обработки ошибок при написании неизвестных команд боту. |
| def warn(ctx, member: discord.Member, reason: str) | Команда ручной выдачи предупреждения пользователю по причине запр-х слов/ссылок от модератора сервера Discord при наличии роли Модератор. |
| def unwarn(ctx, member: discord.Member) | Команда ручного снятия предупреждения пользователю от модератора сервера Discord при наличии роли Модератор |
| def clear\_warns(ctx, member: discord.Member) | Команда ручного снятия всех предупреждений пользователю от модератора сервера Discord при наличии роли Модератор. Если предупреждений нет, выдается сообщение о 0 нарушений |
| def checkhost(ctx) | Команда для проверки наличия роли для организации и проведения квизов |
| def checkgame(ctx) | Команда, позволяющая проверить, активна ли викторина в данный момент |
| def checkplayers(ctx) | Команда, чтобы проверить, какие игроки участвуют в квизе в данный момент времени |
| def t(ctx) | Команда для проверки таймера отведенного времени для ответа на вопрос |
| def join(ctx) | Команда для присоединения к текущему квизу и участия в нем |
| def startgame(ctx) | Команда для создания учебного квиза. Бот-ассистент “Mugiwara” предложит вам ввести все необходимые данные для организации и проведения квиза. |
| def startreg(ctx) | Команда для начала регистрации на участие в учебном квизе. |
| def endreg(ctx) | Команда завершения регистрации на участие в квизе, не позволяющая другим игрокам присоединиться к ней позже во время проведения учебного квиза |
| def question(ctx) | Команда отображения следующего вопроса учебной викторины |
| def result(ctx) | Команда для отображения результатов последнего заданного вопроса в учебной викторине |
| def endgame(ctx) | Команда завершения текущей учебной викторины и отображения таблицы лидеров и результатов учебного квиза |
| def translate(self, ctx, lang\_to, \*args) | Команда для перевода текста на нужный для пользователя бота “Mugiwara” и знакомый для него язык |
| def poll(self, ctx, time: int, vote: int, title, \*options) | Команда для проведения опроса в течение некоторого времени, превышающего 15 минут. Данная команда пригодится для учеников для голосования о времени проведения занятия. |
| def play(self, ctx, \*args) | Команда для воспроизведения выбранной песни/видео с платформы youtube |
| def pause(self, ctx, \*args) | Команда для приостановки воспроизведения текущей песни/видео |
| def resume(self, ctx, \*args) | Команда для возобновления воспроизведения видео с помощью бота-ассистента “Mugiwara” в голосовом канале сервера приложения Discord |
| def skip(self, ctx) | Команда для пропуска текущей песни/видео, воспроизводимого в голосовом канале |
| def queue(self, ctx) | Команда для отображения текущих песен/видео в очереди бота |
| def clear(self, ctx) | Команда остановки музыки и очистки очереди |
| def leave(self, ctx) | Команда для выкидывания бота-ассистента из голосового канала сервера |
| def re(self, ctx) | Команда для удаления последней песни/видео, добавленной в очередь |
| def gpt(self, ctx, \*, args) | Команда для введения запроса и получения ответа от умного ассистента Chat GPT компании OpenAI |

**ПРИЛОЖЕНИЕ 2**

**СПЕЦИФИКАЦИЯ РАБОТЫ ПРЕЦЕНДЕНТА «ПЕРЕВОД СООБЩЕНИЙ****»**

|  |  |
| --- | --- |
| **Название** | Перевод сообщений с использованием команд translate и эмодзи флагов |
| **Аннотация** | Сценарий перевода сообщений с командами translate и отображением результатов с использованием эмодзи флагов |
| **Автор документа** | Дымов Андрей |
| **Рамки применения** | Все текстовые каналы сервера, куда посылаются текстовые сообщения, доступные к переводу |
| **Значимость** | Ключевая задача |
| **Приоритет** | Высокий |
| **Статус реализации** | Сделан рабочий прототип данной функции |
| **Первичный актер** | Любой пользователь на сервере |
| **Вторичные актеры и их требования** | Пользователи ожидают возможности переводить сообщения с использованием команд translate или с помощью доступных эмодзи флагов и получать результаты вместе с информацией о языке перевода, точности и произношении. |

**Базовый поток:**

A. Прецедент начинается, когда пользователь вводит команду translate, указывая текст для перевода и язык, на который нужно перевести;

B. Система переводит введенный текст на указанный язык;

C. Система отображает результат перевода, оригинальный текст, эмодзи флаг языка, произношение и точность перевода;

D. Прецедент завершается.

**Альтернативные потоки:**

**Альтернативный поток 1:**

Условие начала: Пользователь вводит неправильный код языка в команде translate.

A1. Система возвращает уведомление о неправильном коде языка и предлагает корректные варианты;

A2. Прецедент завершается.

**Альтернативный поток 2:**

Условие начала: Пользователь вводит команду translate без указания языка.

A2-1. Система возвращает уведомление о необходимости указания языка для перевода;

A2-2. Прецедент завершается.

**Альтернативный поток 3:**

Условие начала: Пользователь вводит команду с неправильным написанием команды.

A3-1. Система возвращает уведомление о неправильной команде и предлагает использовать правильный формат;

A3-2. Прецедент завершается.

**Альтернативный поток 4:**

Условие начала: Пользователь не вводит текст для перевода в команде translate.

B1. Система возвращает уведомление о необходимости ввести текст для перевода;

B2. Прецедент завершается.

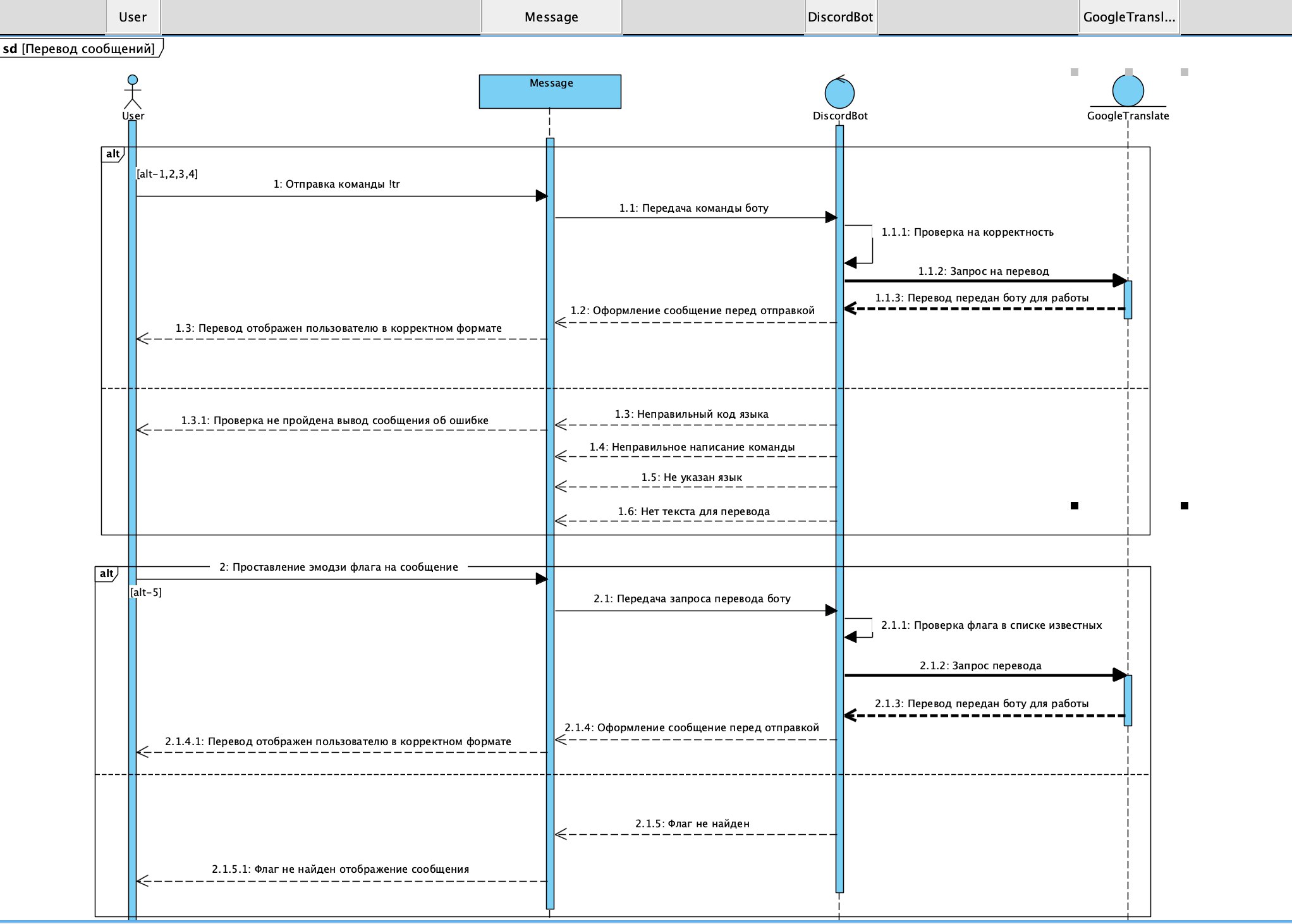
**Альтернативный поток 5:**

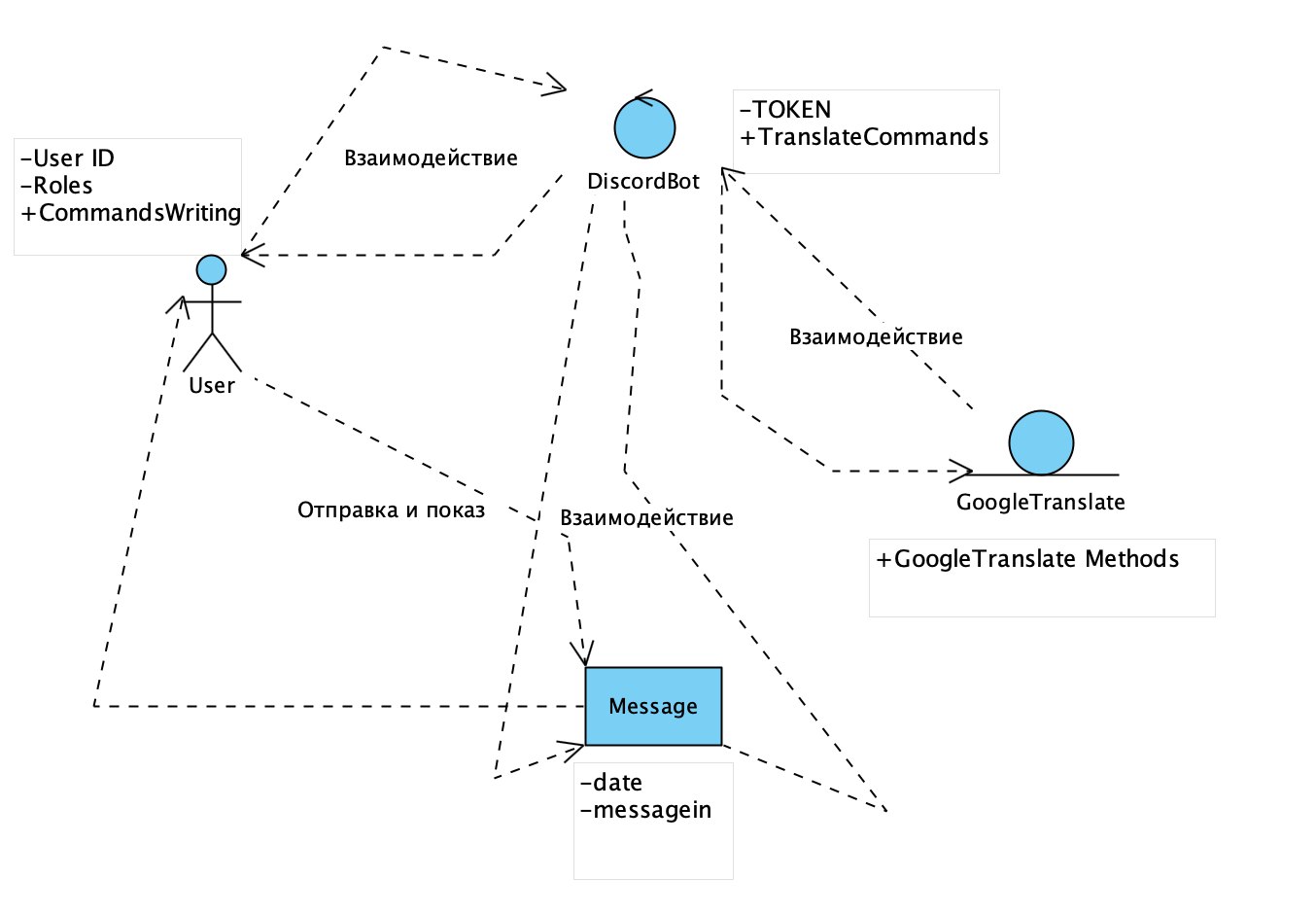
Условие начала: Система не может найти эмодзи флаг для указанного языка.

C1. Система возвращает уведомление о невозможности отобразить эмодзи флаг и предлагает использовать текстовый формат;

C2. Прецедент продолжается с отображением текстового названия языка.

|  |  |
| --- | --- |
| **Предусловия** | Необходимо написать команду перевода в формате !tr (язык/код языка) (обязательно: текст на перевод) или поставить эмодзи флага из известных боту. |
| **Постусловия** | Результаты перевода отображены, и пользователь оповещен о переведенном сообщении. |

**Sequence-диаграмма**

**VOPC-диаграмма**

**ПРИЛОЖЕНИЕ 3**

**СПЕЦИФИКАЦИЯ РАБОТЫ ПРЕЦЕНДЕНТА «ОРГАНИЗАЦИЯ И ПРОХОЖДЕНИЕ МИНИТЕСТОВ»**

|  |  |
| --- | --- |
| **Название** | Организация и прохождение мини-тестов |
| **Аннотация** | Сценарий управления и проведения мини-тестов с использованием команд startgame, startreg, endreg, question, result, endgame, checkhost, checkgame, checkplayers, join, time |
| **Автор документа** | Дымов Андрей |
| **Рамки применения** | Все текстовые каналы сервера, доступные для бота на создание игры и саму игру |
| **Значимость** | Ключевая задача |
| **Приоритет** | Высокий |
| **Статус реализации** | Разрабатывается приоритет |
| **Первичный актер** | Пользователь на сервере с ролями Tutor или Student |
| **Вторичные актеры и их требования** | Организаторы мини-тестов ожидают возможности управлять и проводить тесты с использованием команд, таких как startgame, startreg, endreg, question, result, endgame, checkhost, checkgame, checkplayers, join, time. |

**Базовый поток:**

A. Прецедент начинается, когда организатор вводит команду startgame для настройки игры;

B. Организатор использует команду startreg для начала регистрации игроков;

C. Организатор может использовать команду endreg для завершения регистрации и предотвращения дальнейшего присоединения игроков;

D. Организатор вводит команду question для отображения следующего вопроса;

E. Организатор может использовать команду result для отображения результатов последнего вопроса;

F. Организатор может использовать команду endgame для завершения текущей игры и отображения лидерборда;

G. Организатор вводит команду checkhost для проверки наличия прав на проведение тестов;

H. Организатор вводит команду checkgame для проверки текущей активной игры;

I. Организатор вводит команду checkplayers для проверки списка игроков в текущей игре;

J. Игроки используют команду join для присоединения к текущей игре;

K. Организатор вводит команду time для проверки оставшегося времени на ответ на вопрос.

L. Прецедент завершается.

**Альтернативные потоки:**

**Альтернативный поток 1. Отсутствие текущей игры**

Условие начала: Организатор вводит команду checkgame, и текущей игры нет.

A1. Система возвращает уведомление о том, что активных игр нет.

A2. Прецедент завершается.

**Альтернативный поток 2. Присоединение к игре после начала**

Условие начала: Игрок пытается присоединиться к игре после завершения регистрации (команда join).

B1. Система возвращает уведомление об отсутствии возможности присоединения после завершения регистрации.

B2. Прецедент продолжается с шага J основного потока.

**Альтернативный поток 3. Использование Time не во время теста**

Условие начала: Организатор вводит команду time вне зависимости от текущей активности теста.

C1. Система возвращает уведомление о времени до следующего вопроса или о том, что вопрос неактивен.

C2. Прецедент продолжается с шага K основного потока.

**Альтернативный поток 4. Использование Endgame не во время теста**

Условие начала: Организатор вводит команду endgame вне зависимости от текущей активности теста.

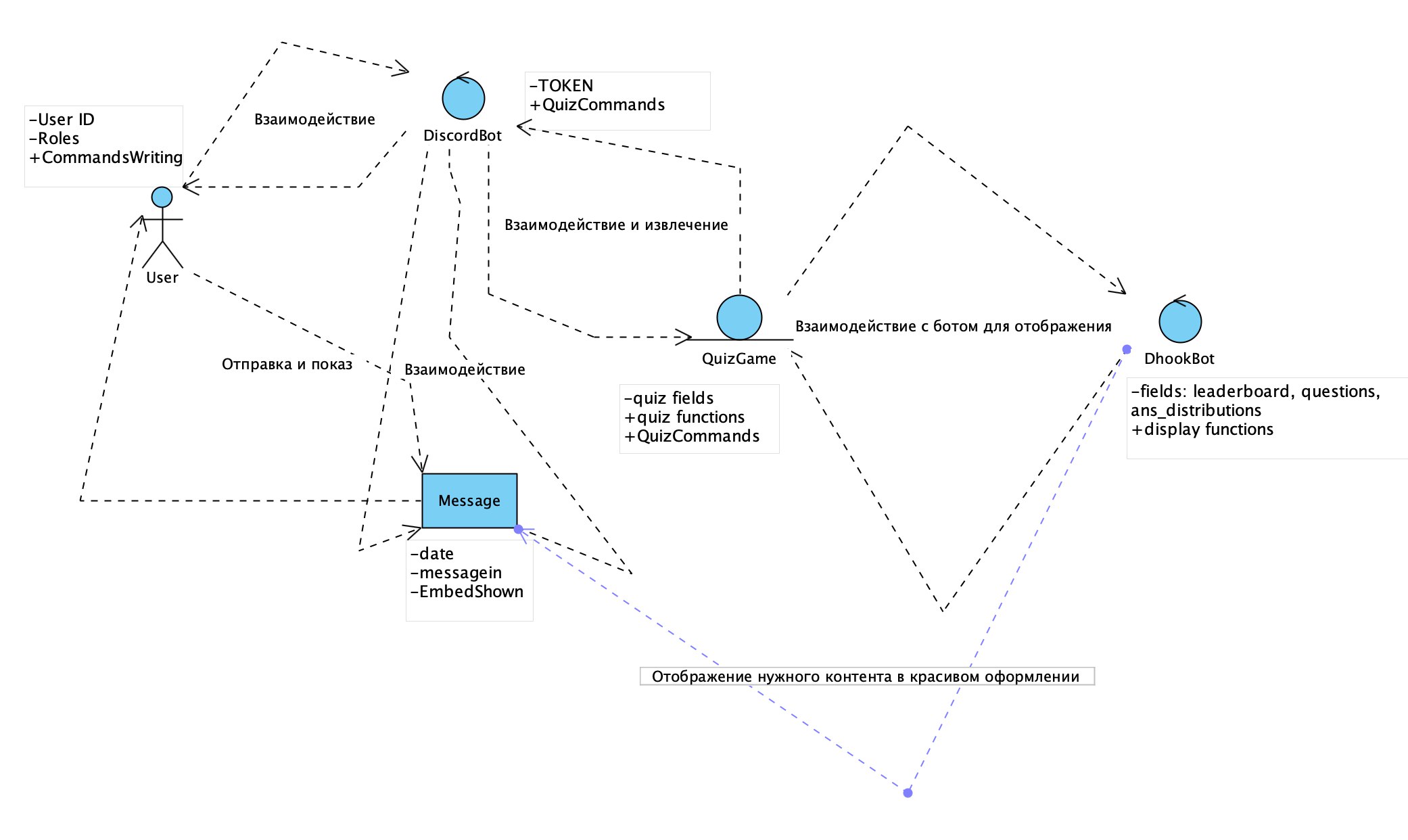
D1. Система возвращает уведомление о завершении текущей игры и отображении leaderboard.

D2. Прецедент завершается.

|  |  |
| --- | --- |
| **Предусловия** | Необходимо наличие текстового канала для проведения мини-тестов, активный статус учеников и активный статус организатора. |
| **Постусловия** | Результаты мини-теста отображены, и leaderboard обновлен. |

**Sequence-диаграмма**

****

**VOPC-диаграмма**

В качестве альтернативного потока для команд !startgame, !startreg, !endreg, !question, !result, !endgame, то есть команды доступных только для хоста квизов, будет один альтернативный поток на этапе проверки роли и потом в случае отсутствия будет переход DiscordBot→Message→User с сообщением о том, что у вас нет прав на использование данной команды, поскольку вы не обладаете ролью хоста.

**ПРИЛОЖЕНИЕ 4**

**СПЕЦИФИКАЦИЯ РАБОТЫ ПРЕЦЕНДЕНТА «ПРОИГРЫВАНИЕ ВИДЕО YOUTUBE»**

|  |  |
| --- | --- |
| **Название** | Проигрывание видео YouTube |
| **Аннотация** | Сценарий управления воспроизведением видео на платформе YouTube с использованием команд play, queue, skip, clear, leave, pause и resume |
| **Автор документа** | Дымов Андрей |
| **Рамки применения** | Все каналы сервера, доступные для бота на вход и проигрывание |
| **Значимость** | Ключевая задача |
| **Приоритет** | Высокий |
| **Статус реализации** | Сделан рабочий прототип данной функции |
| **Первичный актер** | Пользователь на сервере с соответствующей ролью Shanks |
| **Вторичные актеры и их требования** | Пользователи ожидают возможности управлять воспроизведением видео с использованием команд, таких как play, queue, skip, clear, leave, pause и resume. |

**Базовый поток:**

A. Прецедент начинается, когда пользователь вводит команду play, указывая конкретное видео или плейлист для воспроизведения;

B. Система начинает воспроизведение указанного видео или плейлиста;

C. Пользователь может использовать команду queue для добавления дополнительных видео в очередь воспроизведения;

D. Пользователь может использовать команду skip для перехода к следующему видео в очереди;

E. Пользователь может использовать команду clear для очистки очереди воспроизведения;

F. Пользователь может использовать команду leave для выхода из текущего просматриваемого видео или плейлиста;

G. Пользователь может использовать команду pause для приостановки воспроизведения;

H. Пользователь может использовать команду resume для продолжения воспроизведения после паузы;

I. Прецедент завершается.

**Альтернативные потоки:**

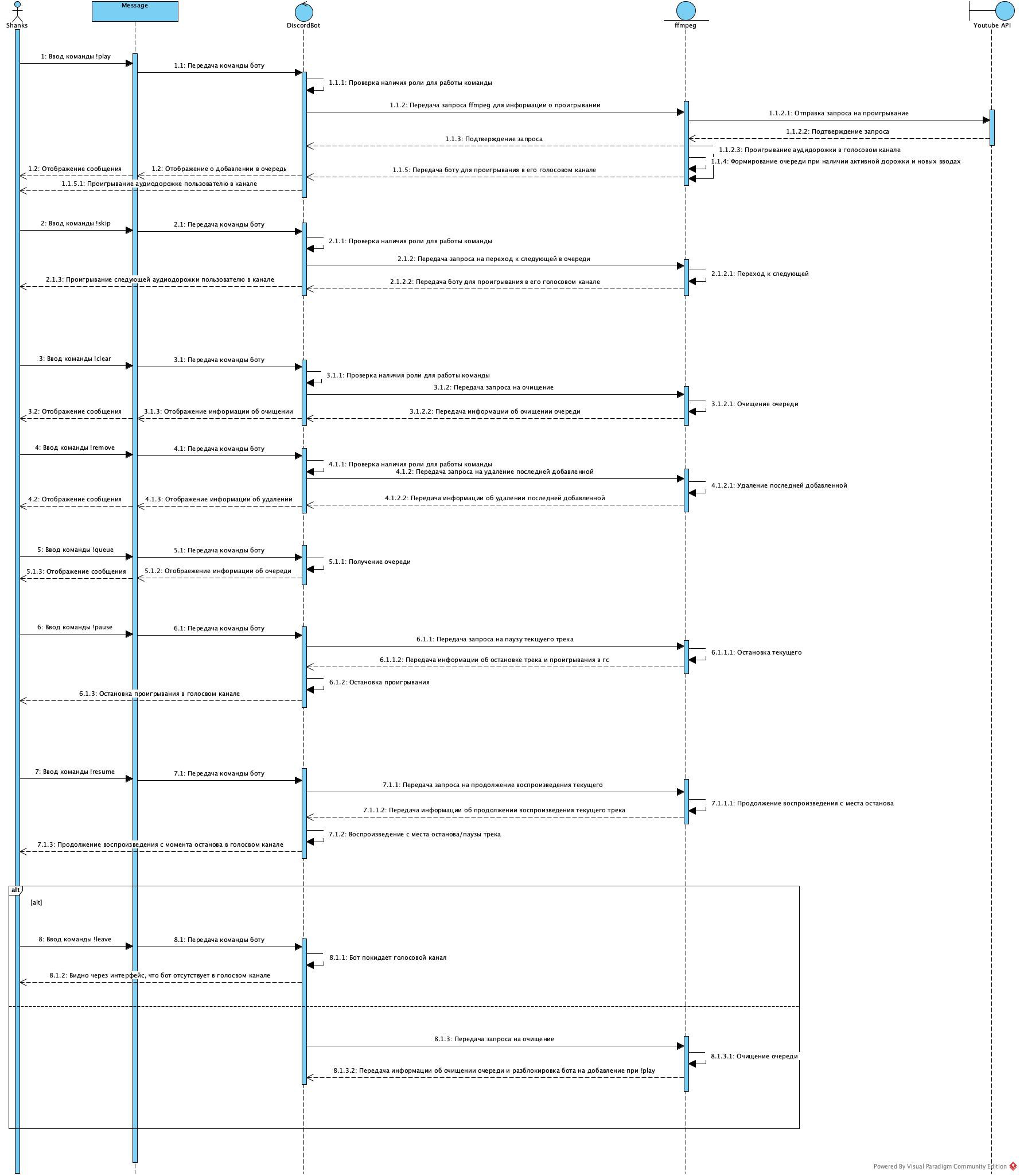
**Альтернативный поток 1. Досрочное завершение пользователем**

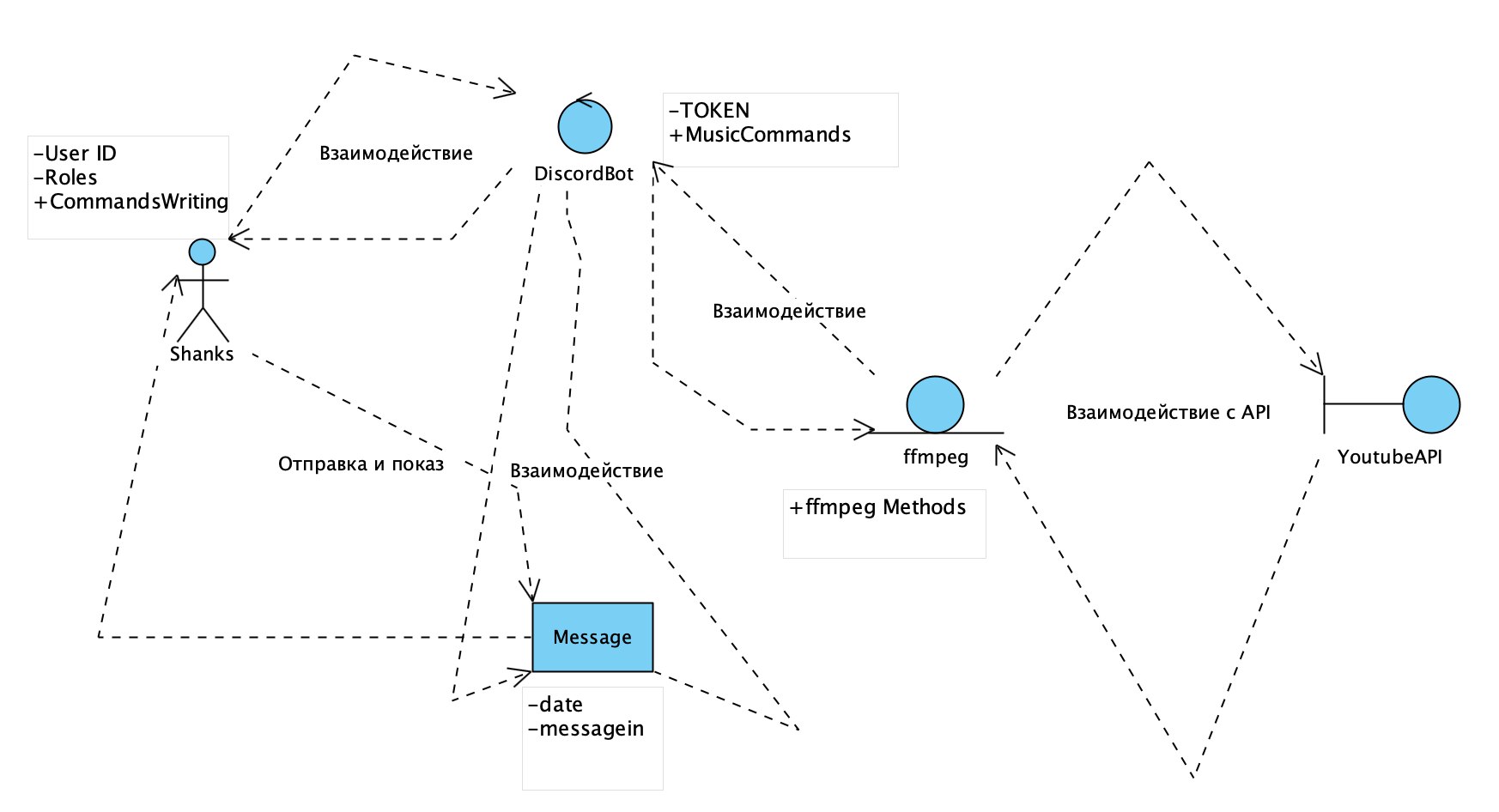
Условие начала: Пользователь вводит команду leave в середине воспроизведения видео или плейлиста.

A1. Система завершает текущее воспроизведение;

A2. Прецедент завершается.

|  |  |
| --- | --- |
| **Предусловия** | Необходимо наличие доступа к интернету, присутствие в голосовом канале сервера и роли Shanks для воспроизведения видео. |
| **Постусловия** | Воспроизведение видео завершено в соответствии с введенными командами. |

**Sequence-диаграмма**

**VOPC-диаграмма**

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |