2 СПЕЦИАЛЬНАЯ ЧАСТЬ

## 2.1 Общая постановка задачи

В современных условиях развития информационных технологий и популяризации потоковых музыкальных сервисов, таких как Spotify, возрастает необходимость создания удобных инструментов для поиска и прослушивания музыки. Одним из таких инструментов является Telegram-бот, который позволяет пользователям искать и находить музыку на платформе Spotify напрямую из мессенджера Telegram.

Целью данного проекта является разработка Telegram-бота, который будет использовать API Spotify для поиска музыки по запросам пользователей. Бот должен быть удобным в использовании, быстро обрабатывать запросы и предоставлять точные результаты.

В результате выполнения проекта ожидается получение Telegram-бота, который предоставит пользователям возможность быстро и удобно искать и прослушивать музыку на платформе Spotify прямо из мессенджера Telegram. Бот должен быть стабильным, быстрым и обеспечивать точные результаты поиска для удовлетворения потребностей пользователей.

2.1.1 Описание входных данных и выходных данных

Входные данные:

* /start - начало взаимодействия с ботом.
* /search <название\_песни> - поиск песни по названию.
* /artist <имя\_исполнителя> - поиск треков по имени автора, исполнителя трека.
* /album <название\_альбома> - поиск треков по названию альбома.
* Идентификатор пользователя (user\_id).
* Сообщения чата.
* Строки поиска, сформированные на основе команд пользователя.
* Токены доступа для авторизации и выполнения запросов к API Spotify.

Выходные данные:

* + Приветственное сообщение.
  + Список найденных треков.
  + Информация об исполнителе и его треках.
  + Список треков в альбоме.
  + Название трека
  + Исполнитель трека
  + Альбом трека
  + Ссылка на прослушивание трека
  + Имя исполнителя
  + Биография исполнителя
  + Популярные треки исполнителя
  + Название альбом
  + Исполнитель альбом
  + Список треков альбом
  + Дата выпуска альбом

2.1.2 Схема работы комплекса

Схема работы приложений представлена в приложении Г.

2.1.3 Описание программы

Программа представляет собой Telegram-бота для поиска музыки на платформе Spotify, написанного на языке программирования Python с использованием библиотек telebot и spotipy. Программа состоит из нескольких модулей, каждый из которых отвечает за определенные функции и задачи. Основные модули программы включают: Основной модуль программы, который запускает бота и координирует работу всех остальных модулей.

* + Запуск бота.
  + Инициализация основных компонентов.
  + Обработка команд и сообщений от пользователей.

Модуль для хранения конфигурационных данных и настроек.

* + Хранение токенов доступа к Telegram и Spotify API.
  + Параметры конфигурации бота.

Модуль для обработки команд и сообщений от пользователей.

* + Обработка команды /start.
  + Обработка команды /search <название\_песни>.
  + Обработка команды /artist <имя\_исполнителя>.
  + Обработка команды /album <название\_альбома>.
  + Взаимодействие с пользователем через кнопки и меню.

Модуль для взаимодействия с API Spotify с использованием библиотеки spotipy.

* + Аутентификация и авторизация с использованием токенов доступа.
  + Выполнение поисковых запросов по трекам, исполнителям и альбомам.
  + Обработка и форматирование данных, полученных от API Spotify.

Модуль вспомогательных функций, используемых в других модулях.

* + Форматирование строк и сообщений.
  + Обработка ошибок и исключений.
  + Логирование и сохранение логов.

2.1.4 Описание набора данных

* + Команды и сообщения (текстовые команды, запросы на поиск музыки).
  + Интерактивные элементы (выборы через кнопки и меню).
  + Метаданные сообщений (идентификатор пользователя, время отправки и т.д.).
  + Информация о треках (название, исполнитель, альбом, ссылка и т.д.).
  + Информация об исполнителях (имя, биография, популярные треки и т.д.).
  + Информация об альбомах (название, исполнитель, список треков и т.д.).
  + Логи ошибок (записи об ошибках и исключениях).
  + Статистика использования (количество запросов, популярные команды и т.д.).
  + Конфигурационные данные (токены доступа, настройки бота и т.д.).

2.2 Описание проблемных программ

2.2.1 Описание проблемной программы №1

Для проекта Telegram-бота требуется реализация функции поиска музыки на площадке Spotify с возможностью сохранения результатов поиска в избранное для каждого пользователя. Эта задача включает в себя создание механизма поиска треков, сохранение результатов поиска для каждого пользователя и возможность просмотра сохраненных треков. Реализация функционала избранного позволит пользователям сохранять наиболее понравившиеся треки для последующего прослушивания или повторного доступа, что сделает использование бота более удобным и персонализированным. Каждый пользователь сможет управлять своим списком избранных треков, добавляя новые треки или удаляя уже сохраненные, что повысит удовлетворенность пользователей от использования бота.

2.2.1.1 Схема алгоритма проблемной программы №1

Поиск музыки: ─ Бот получает текстовый запрос от пользователя, содержащий информацию о желаемой музыке, такую как название трека, имя исполнителя или название альбома. Этот запрос передается боту через интерфейс мессенджера Telegram. ─ Получив запрос, бот выполняет запрос к API Spotify, используя полученные от пользователя данные. API Spotify позволяет осуществлять поиск музыкальных треков по различным критериям, таким как название трека, имя исполнителя или жанр. ─ После отправки запроса к API Spotify и получения результатов, полученные данные обрабатываются ботом. Он форматирует результаты поиска для удобного отображения пользователю. Обработка включает в себя фильтрацию, сортировку и форматирование полученных данных, чтобы представить пользователю наиболее релевантные и полезные результаты поиска.

Сохранение в избранное: ─ Пользователь может выбрать трек из результатов поиска, который ему понравился, и добавить его в избранное для дальнейшего доступа. ─ Когда пользователь решает добавить трек в избранное, бот сохраняет информацию о выбранном треке в базе данных. Для каждого пользователя создается отдельный список избранных треков, где хранится информация о названии трека, имени исполнителя, альбоме и других релевантных данных.

Просмотр избранных треков: ─ Пользователь может запросить список своих избранных треков, отправив соответствующую команду или взаимодействуя с интерфейсом бота. ─ Бот загружает сохраненные данные о избранных треках для конкретного пользователя из базы данных и отправляет пользователю список треков. Каждый трек представлен с соответствующей информацией, позволяя пользователю легко просматривать и управлять своим списком избранных треков.

2.2.1.2 Таблица идентификаторов проблемной программы №1

| Идентификатор | Смысловое содержание | Тип | Разрядность |
| --- | --- | --- | --- |
| search\_query | Текстовый запрос пользователя | String | 255 |
| spotify\_results | Результаты поиска в Spotify | JSON | - |
| user\_favorites | Избранные треки пользователя | JSON | - |
| track\_id | Идентификатор трека | String | 255 |
| track\_info | Информация о треке | JSON | - |

2.2.2 Описание проблемной программы №2

Для удобства пользователей и более интуитивного поиска музыки на площадке Spotify в Telegram-боте будет реализована функция меню. Эта функция позволит пользователям выбирать тип поиска (по названию трека, имени исполнителя, названию альбома и т.д.) из предложенного списка вместо ввода текстовой команды. Такой подход значительно упростит процесс взаимодействия с ботом, особенно для пользователей, которые не хотят или не могут вводить текстовые запросы. Пользователи смогут легко выбирать необходимый критерий поиска, что сделает использование бота более удобным и эффективным.

2.2.2.1 Схема алгоритма проблемной программы №2

Отображение меню: ─ При получении запроса на начало взаимодействия от пользователя, бот отправляет пользователю сообщение с кнопками-вариантами поиска. Кнопки представляют собой варианты поиска музыки по различным категориям, таким как по названию трека, имени исполнителя, названию альбома и другим параметрам. Это сообщение содержит интерактивные элементы, которые позволяют пользователю выбрать нужный тип поиска непосредственно из меню, обеспечивая более удобное и интуитивно понятное взаимодействие.

Выбор пользователем типа поиска: ─ Пользователь выбирает один из вариантов поиска, нажимая соответствующую кнопку на экране своего устройства. Это позволяет пользователю указать предпочтительный критерий поиска музыки, соответствующий его запросу или интересам. Выбор производится путем нажатия на соответствующую кнопку, что делает процесс выбора быстрым и удобным.

Обработка выбора: ─ После того как пользователь сделал выбор типа поиска, бот получает информацию о выборе пользователя и начинает обработку запроса. В зависимости от выбранной категории поиска, бот определяет необходимые параметры для выполнения запроса к API Spotify и подготавливает данные для выполнения поиска в соответствующей категории.

Выполнение поиска: ─ Следующим шагом является выполнение поиска музыки по выбранной пользователем категории. Бот использует API Spotify для отправки запроса и получения результатов поиска, соответствующих выбранному типу поиска. Запрос может быть направлен на поиск треков по названию, исполнителю, альбому или другим параметрам, в зависимости от выбора пользователя.

Отправка результатов: ─ После выполнения поиска, бот отправляет пользователю результаты в удобном для чтения формате. Это может быть список найденных треков, информация об исполнителях и альбомах, а также ссылки для прослушивания треков. Результаты представлены в читаемой форме, что обеспечивает удобство и эффективность использования бота пользователем.

2.2.2.2 Таблица идентификаторов проблемной программы №2

| Идентификатор | Смысловое содержание | Тип | Разрядность |
| --- | --- | --- | --- |
| search\_option | Выбранный пользователем вариант поиска | String | 255 |