# 

# 

Кафедра вычислительной техники

**ПРОЕКТНАЯ РАБОТА**

**по дисциплине «Введение в разработку интеллектуальных систем»**

**Тема: «Разработка системы принятия решений»**

| Студенты |  |  |
| --- | --- | --- |
| Преподаватель |  | Неверов Е.А. |

Санкт-Петербург

2022 г.

# **NEKKO BOT**

Проектная работа по ВВРИС.

Система рекомендаций музыкальных произведений.

Разработчики:

Угрюмов М.М.

Таланков В. Р.

**Основание для разработки**

Изучение и создание СППР. Получение знаний в разработке “умной” системы.

# **Назначение**

По полученным, при оценивании рекомендованных треков, параметрам: настроения и скорости, система должна сделать наиболее подходящую для пользователя подборку разной музыки. Также система должна уметь подбирать новые музыкальные композиции близкие к вкусу пользователя, для самообучения и расширения границ поиска.

# **Требования к программному изделию**

2.3.1 Требования к функциональным характеристикам

2.3.1.1 Система должна представлять совокупность методических и программных средств решения задач:

* Подбор музыкальных композиций в соответствии с вкусами пользователя
* Оценивание рекомендованных песен, создания списка понравившихся.
* Рекомендации музыкальных композиций из разных отраслей и жанров, с разными параметрами скорости и настроения, для расширения разнообразия проигрываемой музыки.

 2.3.1.2 Для задачи принятия решений должен быть реализован алгоритм подбора композиций по темпу и валентности (в будущем имеется возможность увеличить количество параметров для подбора).

2.3.1.3 Интерфейс должен быть интуитивно понятен для клиента сервиса. Пользователь будет указывать свои предпочтения с помощью оценок каждой композиции (нравится/не нравится). Согласованность данных и корректная их передача пользователю. Система должна работать на IBM совместимых персональных компьютерах и смартфонах.

2.3.3.2 Минимальная конфигурация:

ОС: Windows 7/ IOS 13+ / Android10+

Процессор: Intel Pentium 4 or later

Оперативная память: 512 МБ ОЗУ

**Требования к информационной и программной совместимости**

Язык программирования – Python 3.9.

База данных – API Spotify

Основные библиотеки: TelegramBotAPI, Pandas, Numpy

Базирование GUI интерфейса: Telegram Bot

# **Требования к программной документации**

2.5.1 Разрабатываемая система должна включать справочную информацию о работе системы и подсказки пользователю по ходу работы с программой.

**Этапы разработки**

| № | **Название этапа** | **Срок** | **Содержание** |
| --- | --- | --- | --- |
| **1** | Организация | 5  Дней | Создание страницы для кода на Github  Создание среды общей разработки |
| **2** | Разработка ядра системы | 1  месяц | Создание первостепенного интерфейса  Совмещение системы с Spotify API  Создание алгоритма подбора |
| **3** | Организация базы данных пользователей | 2 недели | Создание базы данных предпочтений пользователя  Хранение данных на сервере |
| **4** | Тестирование программного продукта и составление программной документации | 2 недели | Тестирование и последующая отладка ошибок. |

# **Выбор алгоритма при обработке данных**

**Входные параметры**: бодрость и скорость музыкальных композиций, оценка музыкальных композиций пользователем.

**Основной алгоритм**: Входные данные представлены на координатной плоскости, где ОСЬ X - валентность (настроение) муз. композиции, ОСЬ Y - скорость (BPM) муз. композиции. При использовании программы пользователь оценивает рекомендованные муз. композиции, вследствие чего программа определяет радиус музыкальных вкусов пользователя. Пользователь имеет возможность отмечать любимые композиции, которые будут сохранены в отдельный список, и всегда будут доступны для просмотра. При отметке же о нежелании слушать данный трек будет появлятся в рекомендациях гораздо реже, однако не пропадет совсем.

Далее приведено графическое представление распределения более 5 тысяч муз. композиций (Рис.1) на координатной плоскости (скорость/валентность), а также распределение муз. композиций по жанрам (Рис 2 - 5).

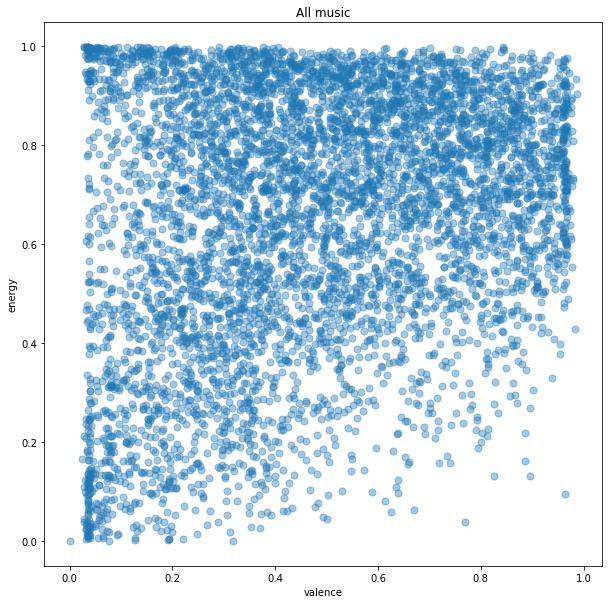


Рис. 1

# 

# 

# 

# Рис. 2

# 

# Рис. 3

# 

Рис. 4

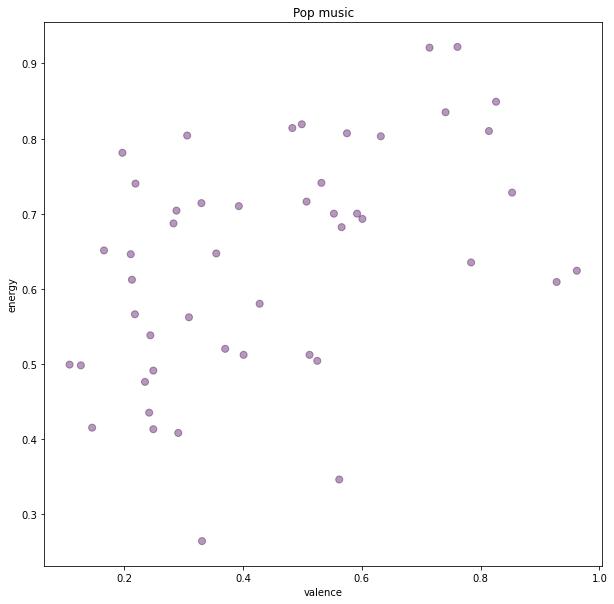


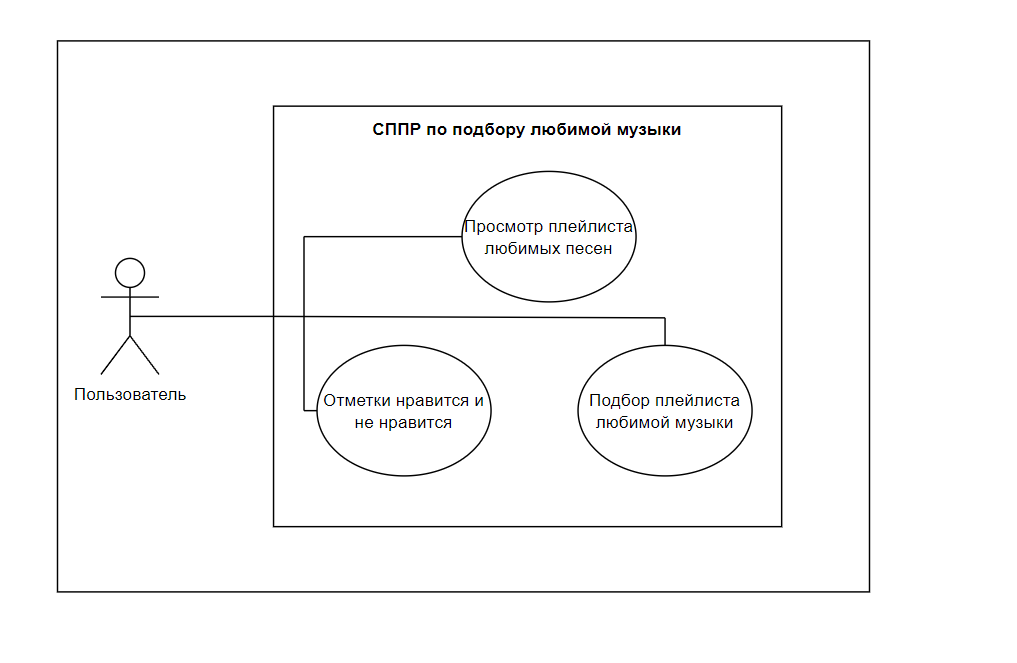
Рис.5

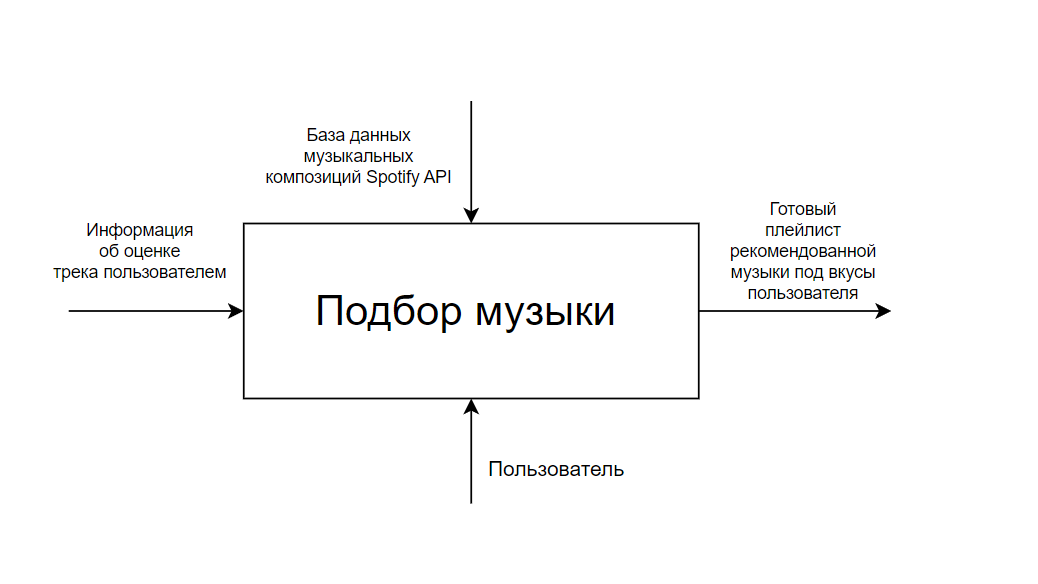
# **О нашем боте**

NEKKO BOT – это бот, позволяющий подобрать плейлист любимой музыки.

Это удобно – использование Telegram бота удобно как на телефоне, так и на компьютере.  
Это быстро – чтобы собрать нужный плейлист достаточно оценить 20 муз. композиций, тогда радиус вкусов пользователя уменьшится и новые подборки будут близки по жанру, скорости и настроению.

# **Диаграммы и структуры проекта**

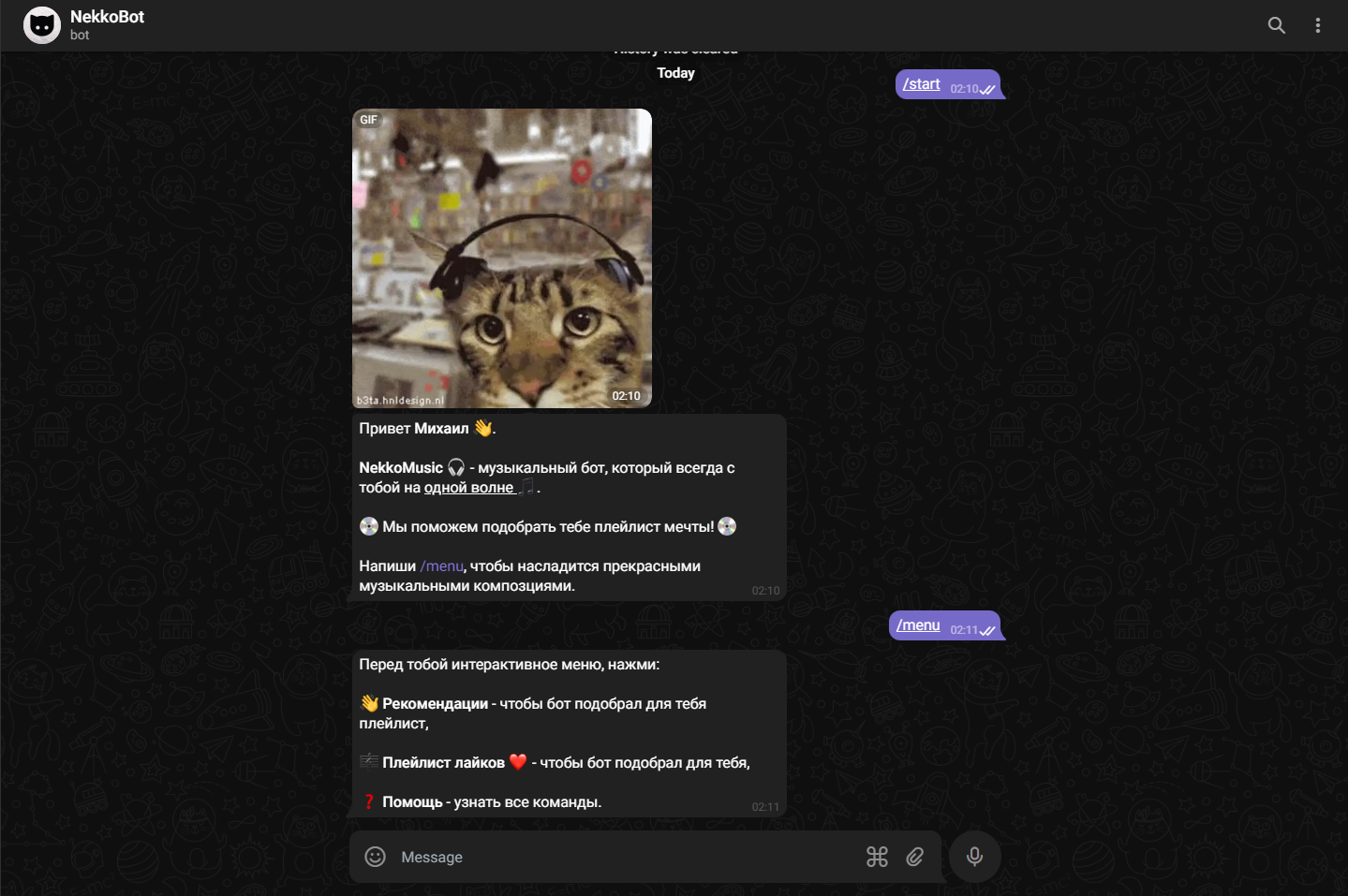
*Диаграмма прецедентов для СППР*

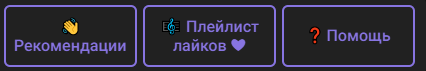
*Контекстная диаграмма СППР*

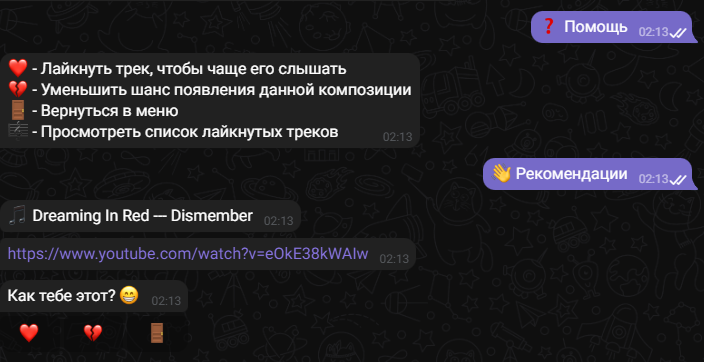
**Функции пользовательского бота**

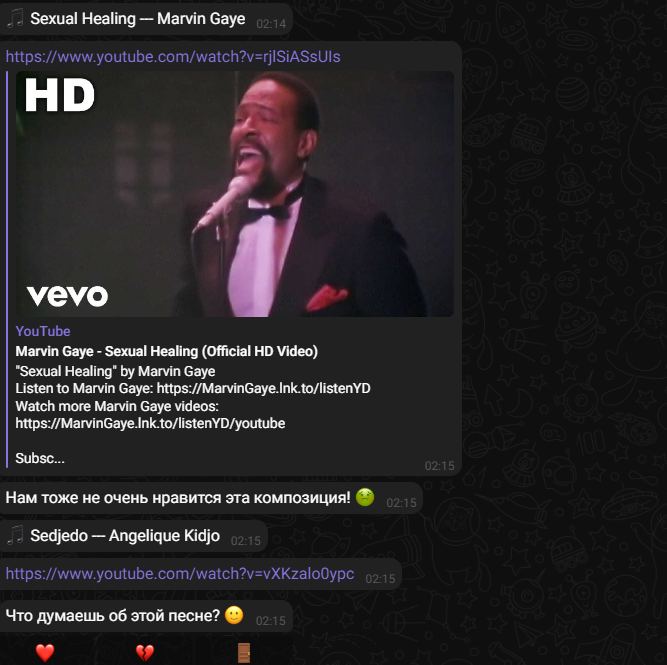
1. Получение списка муз. композиций из базы данных
2. Оценка каждой муз. композиций
3. Просмотр списка понравившихся муз. композиций

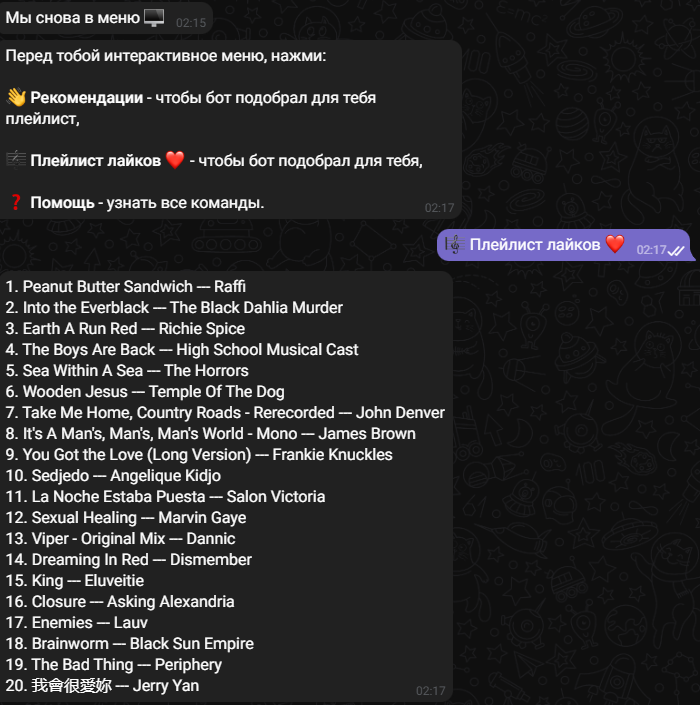
**Примеры работы программ**

*Примеры работы на ПК*

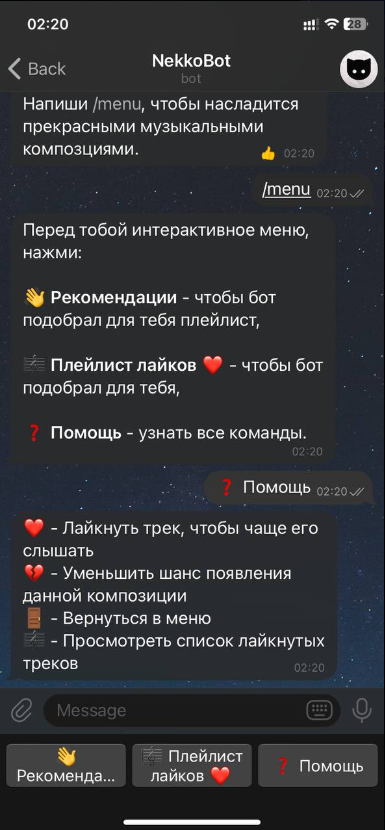
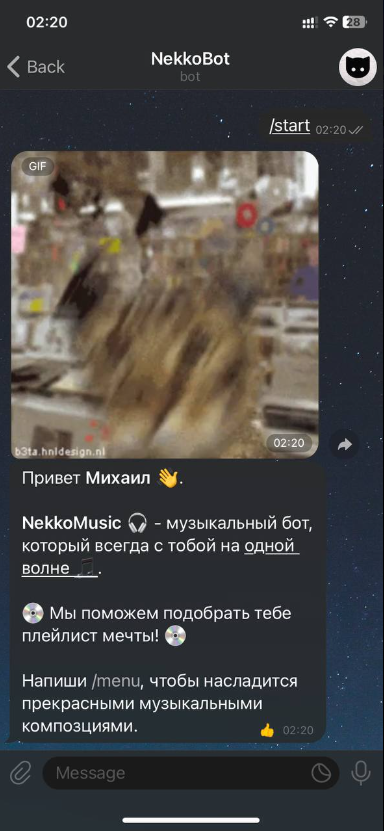


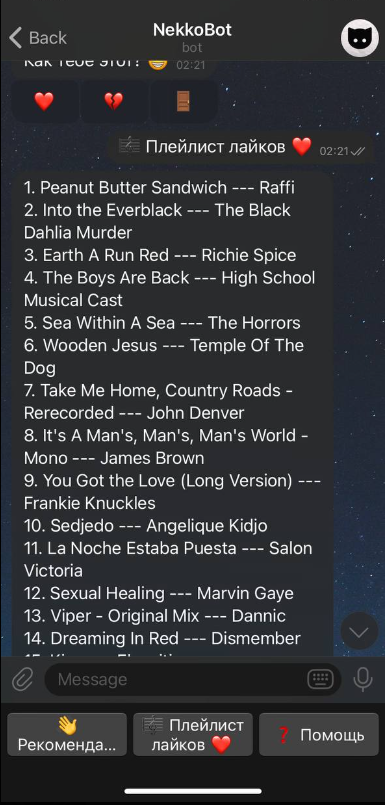
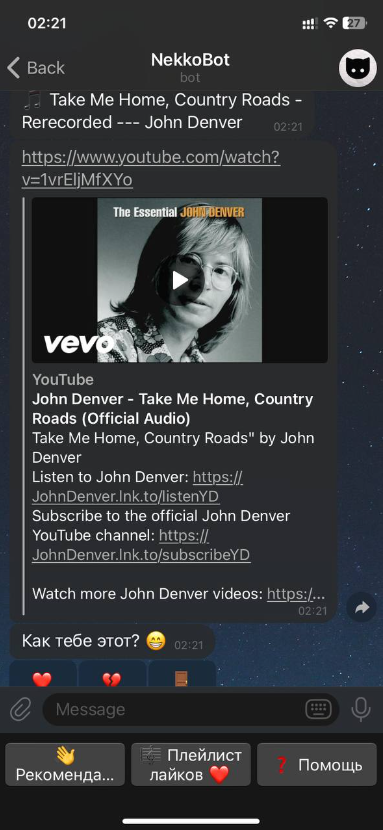






*Примеры работы на IOS*





**Полезные ссылки**

1. /Презентация к защите проекта/

[ВВРИС - NEKKO MUSIC BOT.pptx](https://docs.google.com/presentation/d/1voFAH2jgxcPZD5Lqnr7dtwjo_SkewyUk/edit#slide=id.p18)

1. /Ссылка на GitHub/

<https://github.com/nekoshaurman/mus_project.git>

1. QR-code на бота

