

Matchmaking Bot

Finds people with similar interests to chat with
HSE CS AMI 227

Roles

Conversation search frontend: Engel Danila

Communication frontend: Mishchenko Alexander

Conversation search backend: Artem Abrosimov

Communication backend: Goibov Dadojon

Description

The database stores information about participants who are looking for a suitable companion. Each participant has a unique set of qualities, such as gender, age, hobbies, and more. Based on these characteristics, some range of "quality spread" is used, for example, to connect an 18-year-old participant with those whose ages range from 16 to 20.

When interacting with the bot, messages from one user are stored in a database and then forwarded to another user in their chat with the bot based on a unique identifier (ID). In addition, we are considering implementing additional functionality to improve the participant experience.

Parts

1. Set up a bot in Telegram:
 - Create a new bot via BotFather and get a token.
 - Create a new chat bot in Telegram.
2. Customize the development environment:
 - Install the required packages such as python-telegram-bot, Pandas, etc.
 - Create a new project in the selected IDE.
3. Create a data model for the questionnaire:
 - Design a data structure to represent the completed questionnaire. For example, you can use a class or dictionary to store information about the questionnaire.
4. Create a database to store the questionnaires:
 - Select and configure a database (e.g., SQLite or PostgreSQL) that will be used to store the questionnaires.
 - Create a database schema that defines a table or collection to store the questionnaires.

5. Create and process bot commands:
 - Create bot command handlers for registering, filling out a questionnaire, and joining a group chat.
 - Customize the handling of messages from users so that the bot can receive and process data from the questionnaire.
6. Implement an algorithm for gathering like-minded users:
 - Develop an algorithm that will compare user profiles to determine their similarities.
 - Create functionality to find like-minded users based on given criteria and form a group chat for discussion.
7. Adding topics for discussion:
 - Define a list of discussion topics for group chat.
 - Implement functionality to select a random topic or provide a list of available topics to chat participants.
8. Implement commands to manage group chat:
 - Create commands to add and remove participants from group chat.
 - Implement functionality to check the activity of chat participants and remove inactive participants.
9. Testing and debugging:
 - Test the functionality of the bot at different stages to make sure it works properly.
 - Perform debugging to fix possible bugs or issues in the bot code.
10. Deploy and publish the bot:
 - Set up a web server or hosting to host the bot.
 - Publish the bot so that other users can use it and fill out surveys.