Team Project Report

1. Title of our project

Will You Match Me? UGRP! (Matching program for UGRP Team Members)

2. List team members:

|  |  |  |
| --- | --- | --- |
| **name** | **Email address** | **Student ID** |
| 고낙헌 | [skrgjsdl23@dgist.ac.kr](mailto:skrgjsdl23@dgist.ac.kr) | 201911010 |
| 김주형 | [kimmold@dgist.ac.kr](mailto:kimmold@dgist.ac.kr) | 201911031 |
| 이의정 | [dmlwjd317@dgist.ac.kr](mailto:dmlwjd317@dgist.ac.kr) | 201911131 |
| 한현영 | [hyhan@dgist.ac.kr](mailto:hyhan@dgist.ac.kr) | 201911189 |

3. Motivation for our project

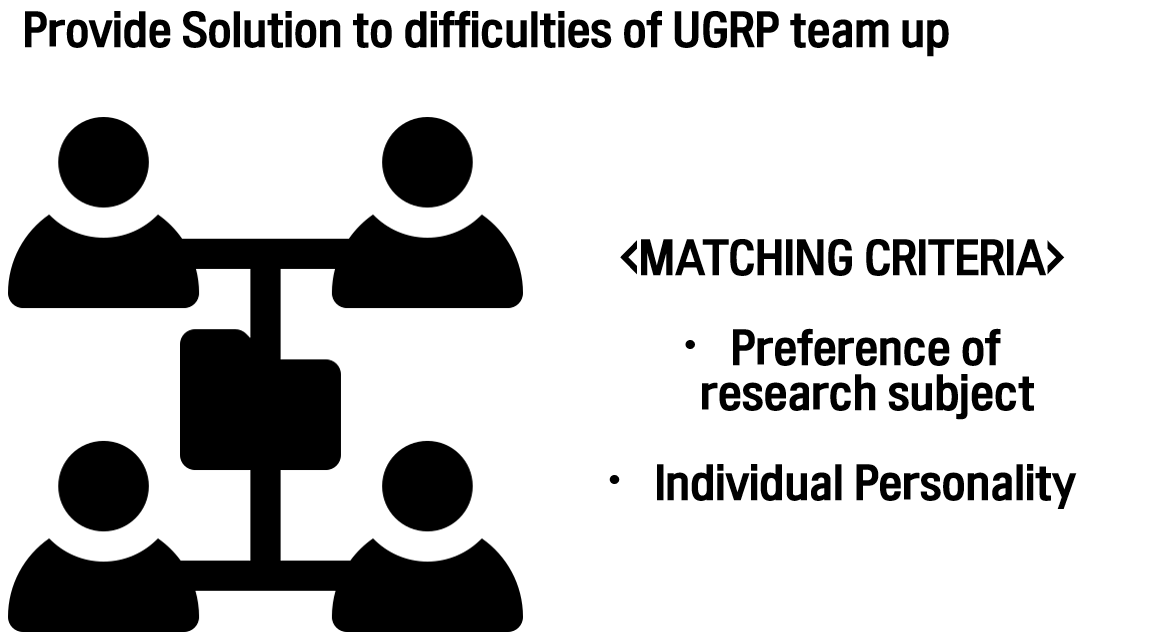


fig 1. motivation

This project is designed for convenient team matching of UGRP, a regular course of DGIST. The purpose of this project is to recommend students suitable for each student according to their input data. In addition, information from various professors will be provided to enable students to use the program to facilitate UGRP.

4. Summarizing what we planned to do and what have done and haven’t done.

|  |  |  |
| --- | --- | --- |
| **Planned to do** | **Implementation State** | **Remarks** |
| Sign-up, Sign-in | ○ |  |
| Enter user information | ○ | Save to in-program database |
| Matching function based on Students’ interests | ○ | Use Euclidean distance |
| Matching function based on Students’ personality(MBTI) | ○ |
| Print out students who fit | ○ |  |
| Matching function that link user with professor(based on user’s interest) | ○ |  |
| Print out list of professors | ○ |  |
| Serving to multiple users using a Web-based database | △ | Deploying in-program database |

5. Describe the implementation design of your project,

a. Provide a class diagram which includes classes with their major member function and other functions.

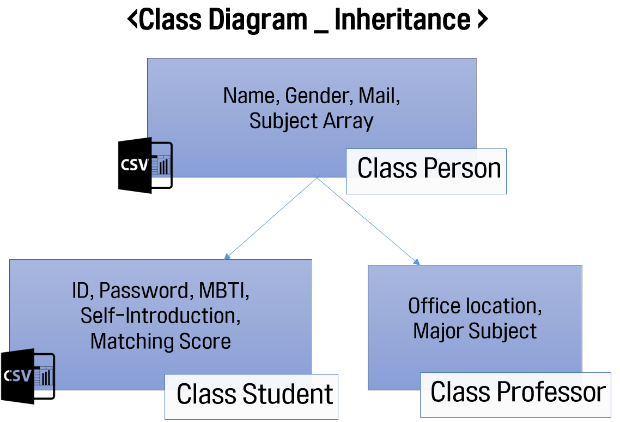
The program is largely divided into three classes. A structure in which a parent class named Person is inherited by a Student and a Professor, and the Person class basically contains member variables such as Name, Gender, Mail, and Subject Array. In the case of a Student class, it inherits the member variables of the Person class and additionally has the ID, Password, MBTI, Self-Introduction, and Matching Score as the member variables. The Processor class inherits the Person class and has Office location and Major subject as member variables.

fig 2. Class Diagram

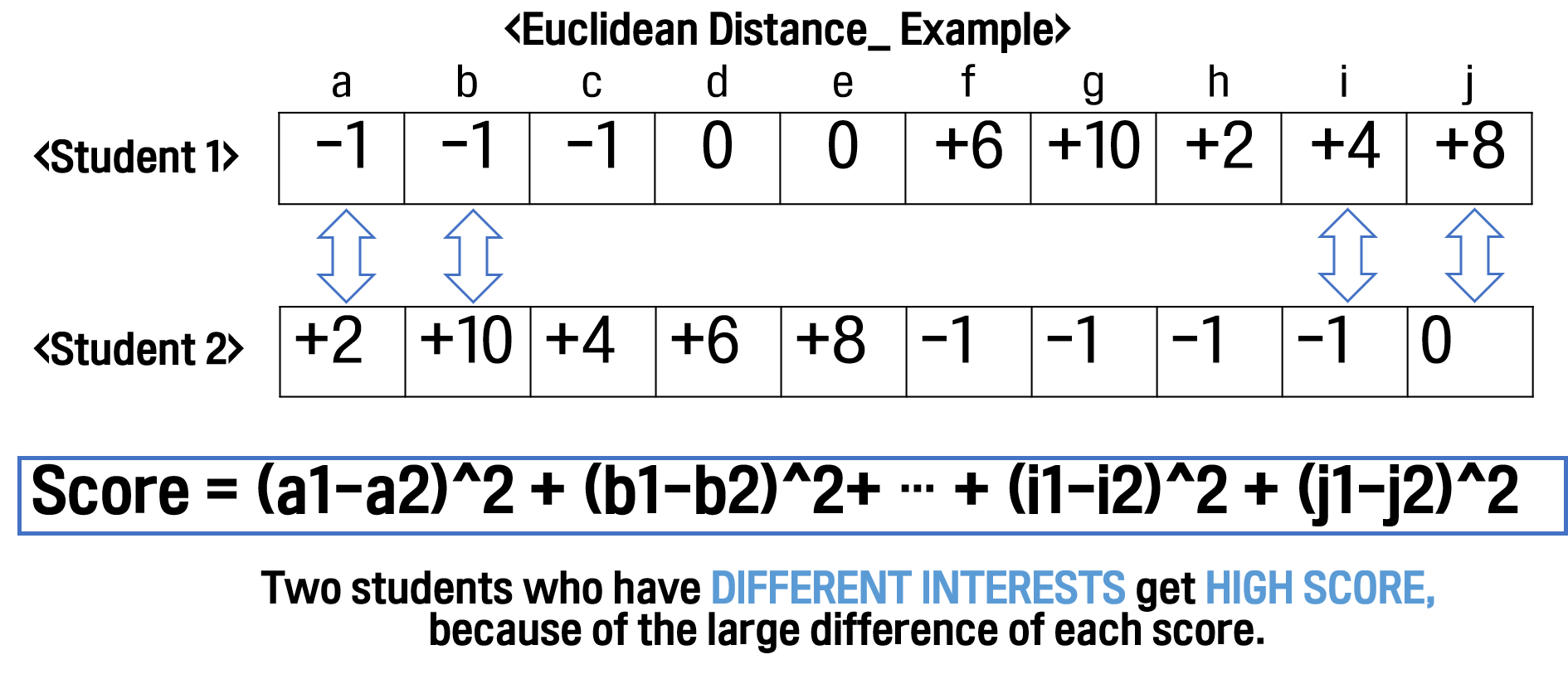
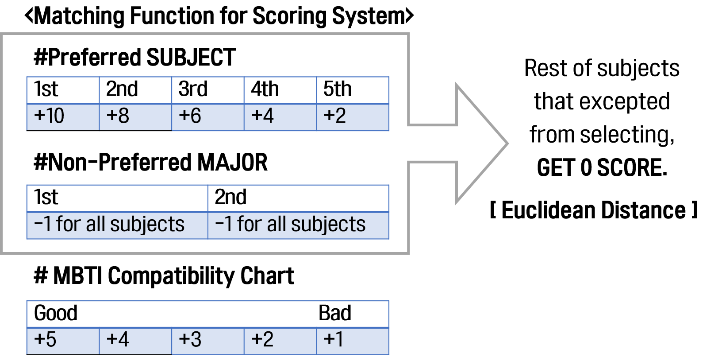
b. Provide details about what we are each in charge of.

fig 3. Euclidean Distance

The method based on the matching algorithm of this program is Euclidean Distance. For Euclidean Distance, priority of preferred subjects can be considered and used as an efficient method for recommendation. It's a more effective way than the Hamming Distance we've been trying to use before. Calculate the scores of each student based on their own score, as shown in the figure above.

Algorithms were designed based on this method, and ultimately reflected in the scores are preferred or non-preferred subjects, MBTI. Based on this, the scores are calculated and based on this, the students are given a list of suitable students. This is similarly used in professor recommendation algorithms.

**fig 4. Matching Algorithm**

6. Explain how to compile our code and how to test.

We have verified that the program is working properly by executing a series of processes we have devised from the user's perspective. During the sign-up process, the user's student number, password, name, gender, school email, MBTI, user’s favorite subject, uninterested subject, and self-introduction were entered. It was subsequently verified that it was well stored in the database. It was then necessary to verify that the matching algorithm based on Euclidean distance measuring technique worked well in practice, and that the results from the entered information were consistent with those from the test process. Since then, we have confirmed that the process of entering and storing the information (name, email, and office location) of professors we randomly generated and matching students with professors who are in the field of interest is working properly.

7. Results section

a. Describe our experiment setup and metrics of success.

b. Present results with discussion.

8. Conclusion of our work and what we learned.

Unlike Python, which used to be used a lot, I felt that programming using C++ was more difficult. Also, unlike Python, which was easy to program by modularizing it based on function, even if C++ is the same function, there were some difficulties in modularizing even though it was a little different for each individual code. However, I could feel the importance of sophisticated programming that Python can't feel well. Furthermore, in the process of realizing common goals, I realized the importance of communication and collaboration.

9. Link of the recorded video

10. Link of the source code repository

<https://github.com/h0han/SE271_project>