Stage IIB: Project Proposal and Specifications

**Group Members:**

Forum Modi, github: ForumModi

Rebecca Harris, github: harrir11

Allison Russell, github: allisonrussell18

Mark Boutros, github: <https://github.com/MarkBoutros>

**Preferred Form of Communication:** GroupMe Groupchat

**Group Name:** Coding Cobras

**Project Name:** Dating Compatibility Database

**Github Link:** [**https://github.com/ForumModi/DCD**](https://github.com/ForumModi/DCD)

**Problem Statement:**

Finding and connecting to people in the current digital age is becoming more difficult day by day. All people can be categorized into zodiac signs and Myers-Briggs personalities; however, not everyone knows which Myers-Briggs personalities/ zodiac signs may clash. Therefore, using a compatibility algorithm taking in account of these two parameters we can connect compatible people for long lasting relationships.

**Objective of the Module:**

Collected data from users. (such as their age, location, gender, education level, star sign, and Myers-Briggs type) which will be stored in a database. Compatible matches will be calculated by weighing each data point. The user will be able to obtain a list of these matches.

**Desired End Project:**

The end project will give each user a list of people they are compatible with ranging from high compatibility to mid-range compatibility. The options of matches will be listed out with a yes or no button for people to choose who they want to be with.

**Need for the Module:**

Human connection is vital for every person. During the pandemic people need to connect more than ever. Our database allows people to reach out and connect with each other and hopefully find lifelong relationships.

**Research Plan:**

Our research plan would include researching Myers-Briggs personalities and which ones are compatible and which ones clash. Then we can create an algorithm for calculating compatibility based on our findings and creating a weighted point system using location. In addition, we will have to create a wide array of varying user profiles which can be created using friends as inspiration or our own imaginations. We plan to have at least 20 profiles held in the database; however, that number may change if it is needed.

**Possible Other Applications of The System:**

Another possible application of the system is a friendship setting. People looking for non romantic companionship can still find people they are compatible with. In addition, there could be a networking setting that would alter what data is being collected and use this to find workers for a job.

**Performance:**

By using hierarchical systems and optimized queries in the database, we can efficiently store data and to optimize performance. In addition, we plan to only use binary way relationships instead of n-ary relationships, which should help with performance as well.

**Security:**

Every user will have a username/email address and password. The password must have at least one capital letter, at least one lowercase letter, a number, and a special character (!,<,$, etc.). Any updates to the system will not be made using administrator accounts, instead programmers will make updates to the program and push new updates during an announced “under construction” timeframe. If there is a user who needs to be reported, users can use a report button with an explanation of why the user needs to be reported, which we would review. Possibly to ensure user data will not be leaked, we could also encrypt the data we store using a key with the most protection to decrypt it.

**Backup and Recovery:**

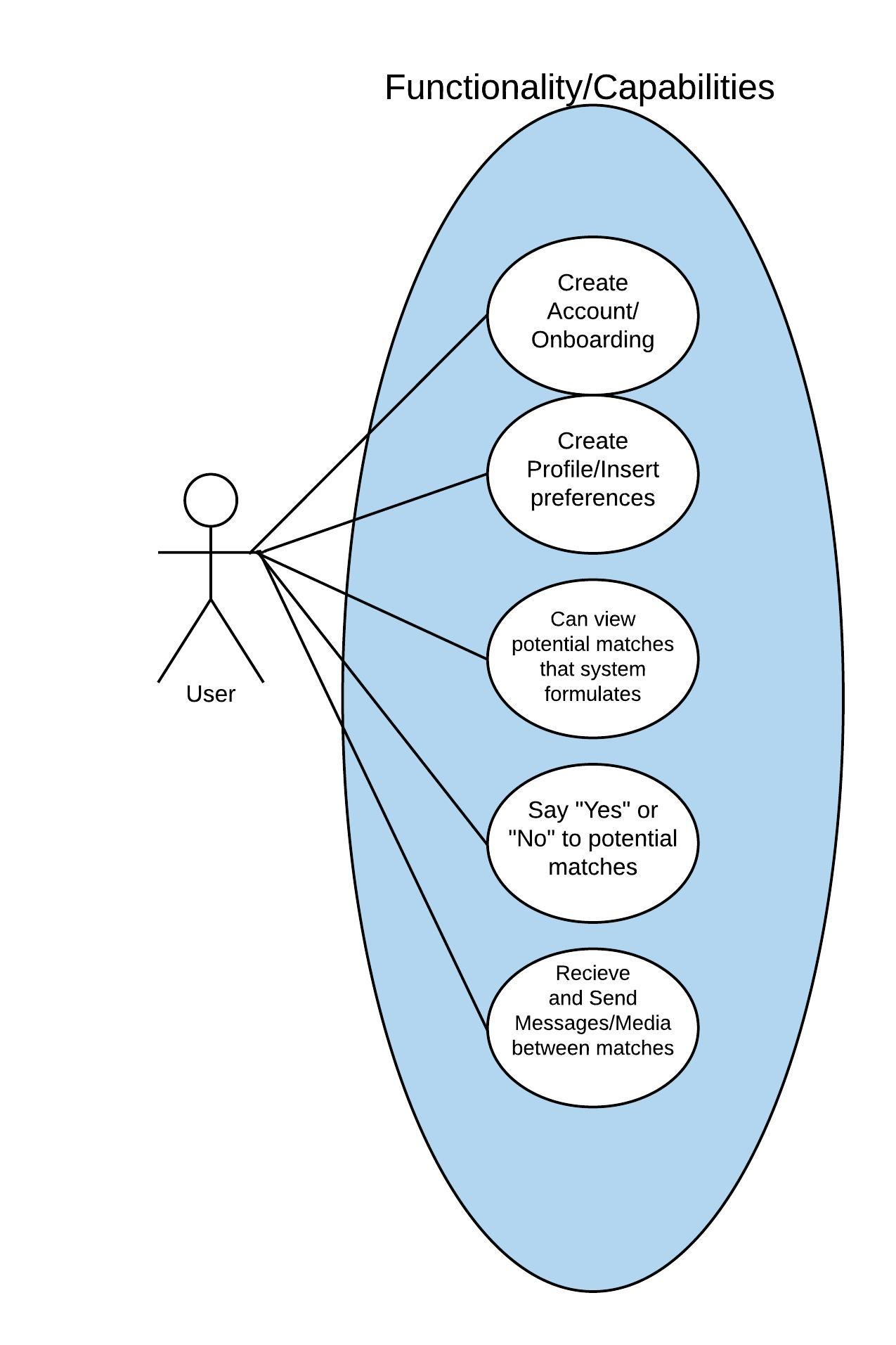
We could create a backup log every 2-3 hours(may change) in case the main system fails. If the system fails, we will revert back to an earlier version in the backup log that does not fail.

**Technology and Concepts to Learn:**

We will learn a database technology such as SQL for inputting user information and compatibility tables. In addition, we will use an object-oriented programming language for calculating compatibility, and will learn how to make an algorithm for this.

Diagram:

Use Cases:



Simplified ER: