Magnet: Bringing People Together

CS 30700 Team 18

Yuanqi Cao
Natalie Evoniuk
Pratim Moulik
Esteban Richey

- CS 30700: Product Backlog Grading Rubric 1 Problem Statement (0.5 points)
  - **(a) Clear and well-defined problem statement.**
  - **(b) Be as detailed as you can in 2-4 sentences.**
  - As social creatures, humans seek out communities to be a part of, whether that community shares a common hobby, religion, location, political affiliation or more. When you move to a new city or gain interest in a new hobby, it can be difficult to find people in the area with similar interests, and if that interest or hobby is inherently something that is best done with others, not finding others to do it with can ruin your desire to do it. Our app, called Magnet, is a social media that brings people together based on their interests. No matter how niche your hobby, interest, or sport is, you can find a person or group in the area that want to participate in that activity with you.

- Background Information (2.5 points)
  - **Explain background information about the problem, the domain, and targeted users.**
  - **Mention whether there are any applications or systems that are similar to your planned work.**
  - **Discuss the limitations of other solutions and how you address each limitation.**
  - The internet can bring people together who have common interests. You can find a Facebook group, Discord server, or a GroupMe for almost every interest or hobby, and connect with others around the planet that share that interest. However, some hobbies are better done in person, and connecting with others with that interest around the globe might not be as enjoyable because you can't do anything in person with those people. If you have such an interest, connecting with others in person that also have that interest might be difficult as it requires an established group that already has that interest in common; for example, if you live in a college town, you can join a local club for your interest, or if there is a community center in your city, they might have a club that match your interest. Our app is for users that have hobbies that do not already have an established group in their area. Some hobbies are uncommon enough to not be guaranteed to have an established group in any city, like some board games, obscure sports, and other lesser known hobbies. Our application, Magnet, is for users that want to connect to people in their area that share a common interest, while that common interest doesn't already have an established local presence. Or, in the

    case where an obscure hobby or interest already has an established group somewhere else, we want our app to direct users with that interest to that group.

    - Any messaging application has similar capabilities as our app. Discord, GroupMe and Facebook messenger allows you to send messages to people and create group messages. The existing applications that allow for group chat creation also have very limited ability to find people to create a group. However, our app is different as it will aid in the process of finding people to message, as well as help in the process of making those groups. The ability to find others in a match-based style is already present in apps like Tinder, or Bumble, but their domain solution is for users looking for casual dating, and sometimes friendships. Our app is solely for building friendships around a hobby or interest.
    - One limitation we might have is how we will go about implementing a robust match-based system. First of all, we will try to implement an automatic and accurate system for basic matching. With a large database, many of the matching processes may become unorganized. This can either be solved through an accurate AI system or through a simpler algorithm, but grasping the idea of “matching efficiency” and “user satisfaction after match” should be the main focus. The second limitation could be how we will implement an accurate and private location service for all users. Not only do we need to utilize a secured database, but we also need to implement a fast and efficient “radar” to scan nearby people. This could be improved by matching the database with the coding language for example if we are coding in python we could explore mysql or postgreSQL. We would also need to improve connection utilizing google maps or other map services API. Lastly, we might be limited by the vast competition of other similar apps. To overcome this barrier, we could try limiting our app services to niche subjects such as sports or outdoor travel focused. We could also make our GUI very friendly and easy to use.

- 3 Requirements (Backlog) (10.0 points)
    - **(a) Divide this section into two subsections, “Functional” and “Non-Functional”.**
    - **Functional:**
        - **User Stories:**
        - **Dont Include Real World Examples. Be Technical and On-point.**
            - Story 1: As a user, I want to be able to create a profile that communicates which sports I play
            - Story 2: I'm a fan of a specific topic, I want to be matched with other similar fans/users of that topic
            - Story 3: As a user, I want the app's menus and navigation to be intuitive and easy to use
            - Story 4: As a user, I want to find people who also like one or more of them so we can make a group chat and talk
            - Story 5: As a user, I want to find people to play activities with in my local community

- Story 6: As a user, who travels a lot, so I want to find people with similar interests in multiple different areas
- Story 7: As a user, I want to list out my hobbies and activities preferences in order to get better match results
- Story 8: As a user, I want to block unwanted messages and activities
- Story 9: As a user, I want to increase and decrease my location search radius casually and easily
- Story 10: As a user, I want to be matched for online activities (such as online meetings or gaming) easily
- Story 11: As a user, I want to report other bad users easily to improve the overall environment of the app
- Story 12: As a user, I want to recommend groups to other of my friends
- Story 13: As a user, I want to be able to manage my existing group such as being a chat moderator
- Story 14: As a user, I want to be able to see my reports and customer service receipts to see whether i am being fairly treated
- Story 15: As a user, matches with groups that do not fit my interests so that other incorrect matches do not happen in the future.
- Story 16: As a user, I want to have the right to freely share my interests either in the app or outside the app
- Story 17: As a user, I want to have knowledge of what permissions this app requests on my device
- Story 18: As a user, I have no interest in inactive groups and would like for them to be reported so that they're not recommended by the app.
- Story 19: As a user, i want to have a real time tracker during my match for safety reasons
- Story 20: As a user, I would like to be able to enable multiple locations for a group profile since members do not live close to each other.
- Story 21: As a user I would like to merge groups with similar interests.
- Story 22: As a user, I would like to have the option of deleting other members from groups if they decide to be inactive or leave.
- Story 23: As a user, I want to be able to rate/review the groups that I have joined for other users to see.
- Story 24: As a user, I would like to create a new genre/ activity by myself
- Story 25: As a user, I would like to show off photos and videos of our group activities to other groups/users in the app

        - Story 26: As a user, I would like to be matched specifically for competitive activities such as tournaments
        - Story 27: As a user, I would like to turn on and turn off my notifications easily
        - Story 28: As a user, I would like to keep my data private and not be tracked constantly by the app when i'm not using it
        - Story 29: As a user, I would like to choose who to open my profile to
        - Story 30: As a user, I would like to have other users be authentic and at least somewhat verified in age or other records in order to have a safe and positive matching experience
        - Story 31: As a user, I would like to share and promote info from inside the app to outside the app
        - Story 32: As a developer or user, I would like to be able to file for potential app improvements or questions and receive consistent feedbacks from the customer support or developers
        - Story 33: As a freelance developer, I would like to make open contributions and improvements to the software (with moderators review) from Github
        - Story 34: As a user, I would like to have a personal page, I would like to have multiple fun features including uploading HD pictures, mini blog, and writing other info
        - Story 35: As a user, I would like to have extra paid privileges where I could access more matches, secret location services, and other extra digital decorations
        - Story 36: As a user, I would like to filter age limits
        - Story 37: As a user, I would like to be able to filter gender limits
        - Story 38: (**If time allows)** As a user, I would like to user multi-lingual version of the app
        - Story 39: **(If time allows)** As a user, I would like to use a multi-country version of the app
        - Story 40: **(If time allows)** As a sponsor, I would like to insert customized, paid advertisements into the app
    - **Languages/Environment:**
        - We will be creating our application using the XCode IDE. The languages we use will include C and Swift for the front-end of the app, including the application itself. The back-end of our application will use Python, which will coordinate the user profiles and user data online.
- **Non-Functional:**
    - **Response time:**
        - Since only 79% of users retry an app only once or twice if it fails to function the first time, a key feature of Magnet is fast response time. Upon opening the app, the login page should be displayed

within 500ms. If the user is already logged in, the objects of their dashboard/home page must be loaded and interactable within 2000ms. These time goals are arbitrary, but reaching them will achieve the feeling of an application that is quick to load
        - Power Consumption/Memory: We want our app to work efficiently, mobile devices have relatively less memory and slower processors, so our code has to be designed with efficiency in mind. With efficiency also comes lower power consumption, which is another consideration that is important for mobile devices
        - Interoperability: Since we allow users photo album access, we will need an IOS framework for sharing this data
    - **Scalability:**
        - Our server-side implementation must be able to handle hundreds of users, potentially more if the app becomes more popular.
        - With the optimistic expectation that our app will achieve limited nationwide use, we should aim to have our back-end system to host at least 100,000 user accounts. We should also expect
    - **Usability:**
        - The app has to be usable for all ages, so high-level software knowledge is not required to use the app. We will also avoid using slang in the wording of our application, so that anyone of any cultural background can use the app without confusion
        - If time allows, we will try to implement localization options, such as changing the language of the application to whatever language you prefer
        - Users will be able to access the app and see all available group options 24 hours per day.
    - **Security:**
        - Have the option to turn location tracking on and off for user/s privacy concerns
        - Secure login for users, including a secure way to store credentials in the back-end (not plaintext!).

- **(b) Format your user stories properly (e.g. “As a , I would like to so that .”) and you may mark some of the user stories with “(if time allows)” tag, when appropriate. Provide as much detail for each user story as you can right now.**
- **(c) In the Non-Functional section, please include performance requirements such as response time, scalability, usability, security, and etc. and quantify your requirements, e.g. within 500ms, 24 hours per day, 10,000 simultaneous requests, and etc. This can be either in a “user story” format with detailed explanations or just a discussion for each requirement.**
- **(d) Create as many user stories as you can. Even if there is not enough time to finish all of them, it is better to have too many user stories than too few.**

- **(e) All user stories but the ones marked as “(if time allows)” should contain enough work for this semester so that each team member will spend around 10 hours/week for the project. If not, points will be deducted and the team will be asked to resubmit the backlog with adequate amount of work**

Total: 14 points (14% of your project grade)