December 2nd, 2020

CPSC 481 STAGE 5 REPORT

TEAM QUEUE

Prepared by: Brian Du, Pragya Chopra, Jase Pasay, Lougheed Wan, Chuks Nwakanma
TA: Philmo Gu

Executive Summary

Individuals often find themselves looking for peers that share the same passion for gaming as they do. We created Team Queue to cater to this specific need. **Team Queue** is a **web-based application** designed to connect individuals on the basis of their common interests in gaming.

This report describes the approach this project followed in the different phases of designing, researching and evaluating. The first part explores the various design problems and solutions and recognizes the stakeholders. The second part describes the active participation of the users, through the methods described in IDEO cards, which allowed a deep analysis and understanding of their requirements. Lastly, the third part showcases the Low-Fidelity and High-Fidelity prototypes created for this project along with a description of our findings for the Heuristic Evaluation and the corresponding changes made to the prototypes.

This project has taught us how to bridge the gap between the experiences of the developers and those of the population they are designing for.

Introduction

The aim of our project's application is to assist users, primarily video game players, by providing a platform through which they may group up with other users that share a similar interest in video games. Users will be able to search for other players based on a variety of personal and game-related criteria using an automated matchmaking system. Users may also take a more manual approach through the use of preset lobbies and lobby creation. Once grouped, users will be able to communicate with other lobby members via voice or text chat and from there, organize their in game activities.
Our report will be covering the overall project process. Initially, the project began with brainstorming, stakeholder evaluation and idea formation, after which, we moved on to user research and documented our findings. Next, was the prototyping process, this consisted of developing a low-fidelity iteration, which we built upon into a high-fidelity iteration. Lastly, the final stages consisted of heuristic evaluations, feedback gathering and hi-fi prototype improvements.

Design Problem

Many games require/are better played with a group of people. However, many people tend to play by themselves because either their friends don't play games or don't play the same games as them. This app can serve as a great way to talk to like minded individuals and create connections. Many applications out in the market right now are tailored to connecting with friends i.e. require an invite (such as discord) there are very few, if any, applications in the market that allows for strangers with like minded interests (a certain game in this case) to come together and meet up.

Design Solution

Our basic idea is to design a social networking application for the gaming community. The primary goal of this application will be to connect people who play and/or have an interest in a specific game with people that have similar interests. The application will be web-based. The user can log into the system and they can do one of the two things to start connecting:

Firstly they can specify a game, provide some additional details and be automatically connected with individuals who fit these categories or they can manually search a game which will show different groups related to that game and the user can select one of these groups. The user can also connect with individuals and create their own lobby with their friends. Our application will also have additional functionalities to enhance communication such as private/group chats and voice calls.

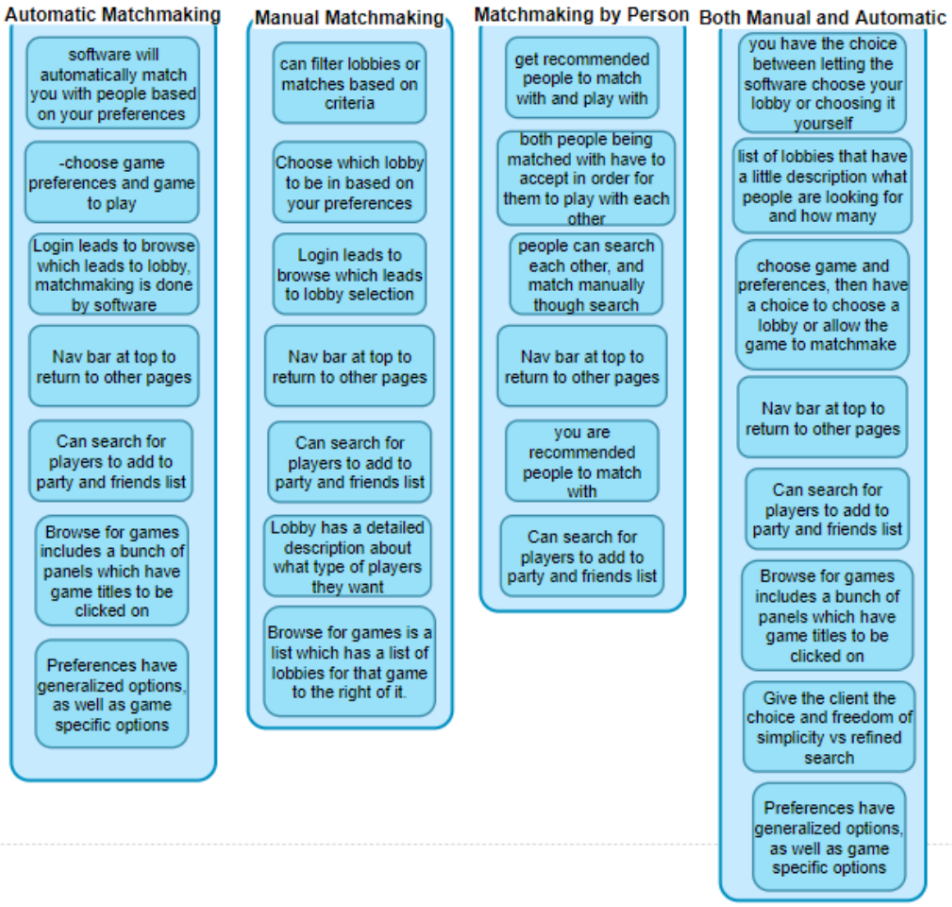


Figure 1. Our Affinity Diagram, which outlines our different ideas on what our system would look like.

Stakeholders and Users

- Gaming community(end user):
  - This is the primary community that we will be targeting. This application will be tailored towards them. If income was made with this application, most of it will come from these people.
- Game company managers
  - Collaboration would help show authenticity and support.
- Server side developers
  - This application requires a server to host all connections between accounts, thus it is very important to have server side developers managing and keeping an eye on databases
- Software Developers
  - As the UI designers we have to work closely with the software developers to create a smooth and efficient application. We have to make sure that the ideas implemented are viable in a developer standpoint
- Sponsorship companies
  - Funding is an important part of any application, pitching to sponsorship companies is a vital part of any application design. Thus these companies play a major role in the funding of our application

User Research Methods and Process

For our user research methods, we decided on using “Survey”, “Scenario Testing” and “Competitive Product Survey” as our 3 methods of choice.
The reasoning behind using this combination of research methods was the idea that we wanted a grand overall snapshot of what our system should look like. We based our survey to target the gaming community, as these people were our ideal audience.
Through the survey, we asked very generic questions about the look and feel of what our software should be like to get a general sense of what the gaming community values in a matchmaking software.
In our “Scenario Testing” we created another survey that once again targets the gaming community. However, this survey consisted of multiple scenario based questions, which targeted on potential issues our system will come in contact with, such as toxic/rude players or age gaps etc. For each of these questions we gave counter solutions and asked the clients to pick the most ideal one, this helped us gauge what implementations we need to safeguard against these problems.

Finally our “Competitive Product Survey” allowed us to research what other similar applications were out on the market as of right now, and it allowed us to define and differentiate our software against other software on the market. This made it easy for us to hammer home our own unique idea, in which no other software on the market is covering.

User Research Findings

Using these research methods, our team was able to gather a broad range of data which correlate to what our potential users would want in the system. A general trend we found was the idea that there must be a preferences selection for each game.
Our users voiced heavily on the importance of limiting toxic and rude behaviour, and the most ideal solution that was chosen was the idea of grouping like minded people together. (casual players with casual players etc) Another important factor that was mentioned greatly was the idea of having lobby chat, direct chat as well as voice comms in lobbies. Our users mentioned how important it was for them to be able to communicate with the people that they were lobbied with. Finally our users also mentioned how important it was to them that they have a customizable account, which can store their game info, name, password, region etc. They emphasized that having this information on the system will make it easier for them to find people that have the same interests as them.

Important Design Choice and Justification

As our team was developing ideas for our UI design, we took into consideration all the information we recovered from our user research. The first implementation we took into consideration was the idea of making our matchmaking completely anonymous. This means that when a user queues up for a lobby, they are not able to choose the people he/she queues up with. We decided to implement our main way of matchmaking as such as we believed that our whole philosophy for this system was to have people meet other people to further growth. If a user can choose who they queue with, then they will just get picky and the overall flow of the system will decrease. However, we also realized in some situations, such as having really specific preferences, it was also beneficial if a user can choose a specific lobby to be in. Thus we also decided to make a “manual matchmake” option, where users are able to specifically choose which lobby they would like, and each lobby would have a description on what the lobby members are trying to achieve. Using this as a basis of our UI design, we went ahead and created our low fi prototype based off of these 2 design choices.

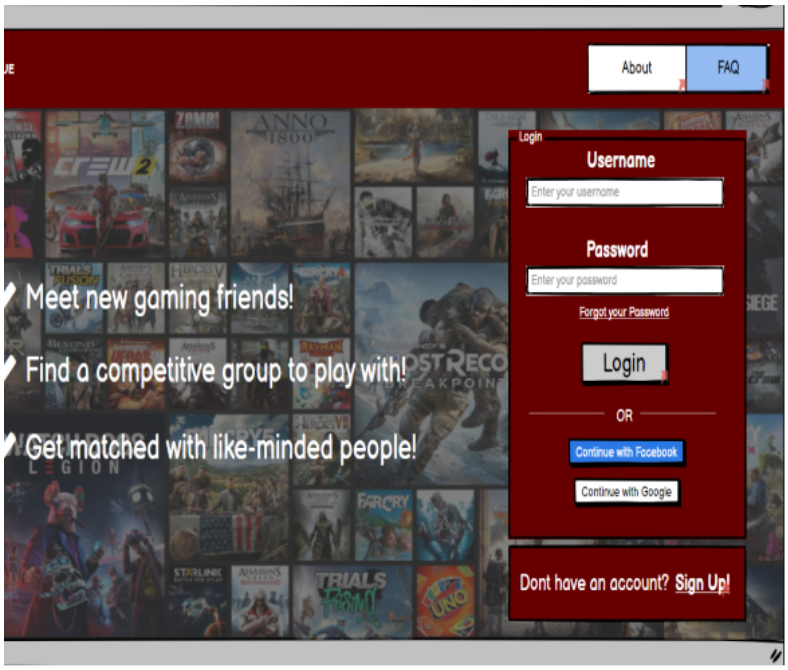
Low-Fi Design Lessons Learned

After working on the Low-Fi design of our prototype, we learned both things that we did well and could improve upon. As a group we were open to each member's ideas and gave good constructive feedback on them, leading to effective discussions and usage of time in meetings, as well as more polished ideas. Something that could have been done better was estimating the time it took to complete some tasks. For example, we had initially thought that creating the Balsamiq pages for our low-fidelity prototype would have taken much more time than it did. This did not cause time pressures because of our effective time management, however had we estimated better we could have seen quicker progress. In addition to this, we sometimes misinterpreted the stage three document about what was expected, which led to some minor delays.

High-Fi Design Lessons Learned

A challenge we faced was in our high fidelity prototype tool. Not all members were experienced with Adobe XD, and its learning curve slowed progress some. This did cause a minor delay in our timeline and shortened our window for error, however we adapted to this well and it did not cause excessive time pressures because of our effective time management. Below are some images of the pages from both our low fidelity and high fidelity prototype.

Figure 2: Login page design (low fidelity (left), high fidelity (right))



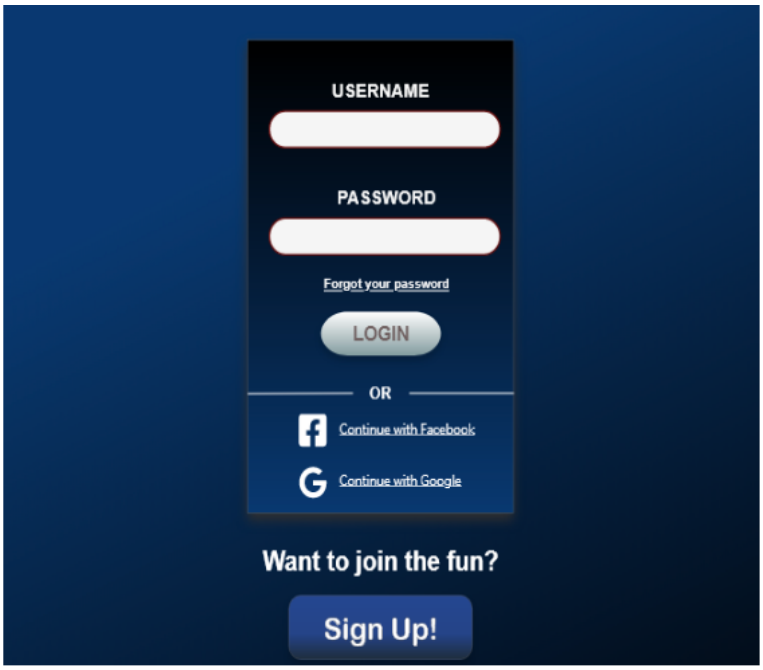


Figure 3: Browse Page (low fidelity (left), high fidelity (right))

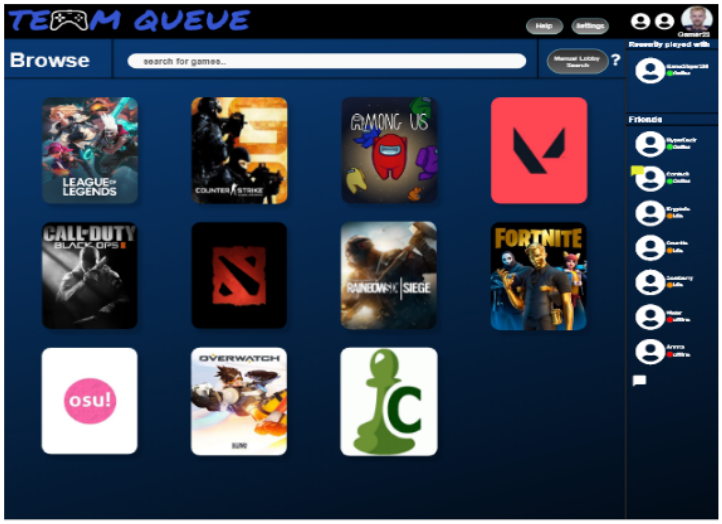
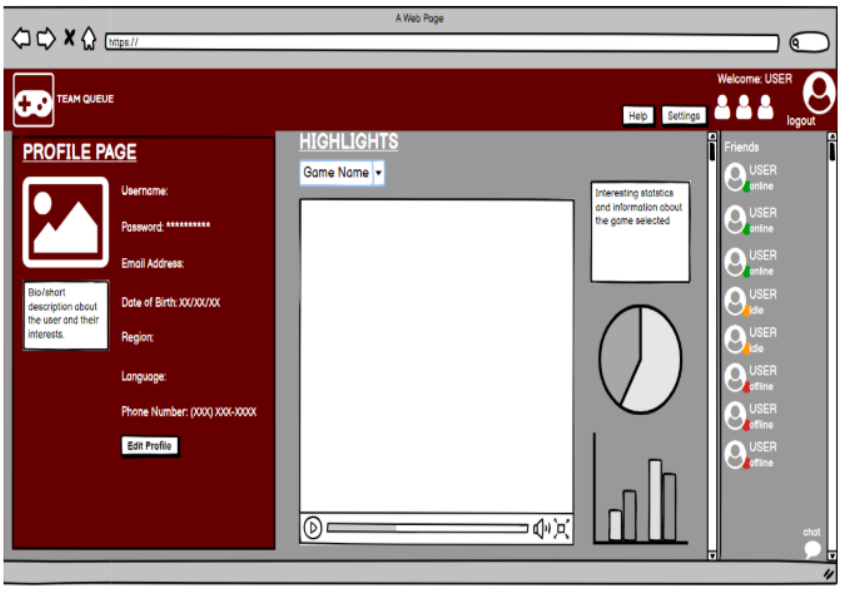


Figure 4: Profile Page (low fidelity (left), high fidelity (right))



Heuristic Evaluation:

Process:

To conduct the heuristic evaluation we split our team into two groups, one of three evaluators and another of two reviewers. Where members Lougheed, Chuks and Jase were the evaluators, and members Pragya and Brian were the reviewers. The evaluators each conducted their own evaluation of the system separate to the other evaluators and then reported their results to the rest of the team. All evaluations were done in accordance to the following ten UI design heuristics:

- Visibility of System Status
- Match between system and real-world
- User Control and Freedom
- Consistency and Standards
- Error Prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Once each evaluator had completed their inspection of the system, the reviewers took the evaluators reports, and again separate to one another, classified each identified problem in the reports based on its perceived severity. The evaluators then met to compare their individual findings and to make some initial decisions about the found issues and how they could be fixed, as well as if they should be fixed in this stage of the project. Lastly, the team met all together to summarize this process, our findings, and decisions into one finalized heuristic evaluation.

Findings and decisions:

From our finalized heuristic evaluation, issues were raised in four of the ten UI design heuristics, being:

- Error prevention
- User Control and Freedom
- Aesthetic and minimalist design
- Help and documentation

The error prevention issue found had to do with a user being able to login without inputting any login information. This was classified as a minimal severity problem and as a team we decided that we did not feel it was necessary to fix this issue in this stage of the project. As our prototype has not included any coding we felt that implementing exception handling into our prototype seemed unreasonable. However, we still felt it was a notable error and it is something to be acknowledged once we are at an implementation stage in development.

Further, the lack of a “back” button on some pages we felt was a minor severity user control and freedom problem, and as a team we felt this too was unnecessary to implement now. Since our application will be a web application it makes more sense to simply use the browsers own back button as users will be guaranteed to be familiar with it and its functionality. Also, similar to the exception handling, in our prototyping tool we are unable to track the page before, and thus a functional back button was not feasible.

A minimal severity issue that was raised in regards to aesthetic and minimalist design, more so on the minimalist design, had to do with some pages feeling cluttered. This issue felt easily solvable to the team and thus we decided to focus some of the pages to better serve a single purpose (eg. splitting our support page into a separate FAQ and contact page), and take away some extra items where possible (eg. extra text on login/signup pages). Lastly, it was felt that there was a lack of help and documentation for the add friends feature in our application and was deemed a minor severity error.

As a team we decided it was something worth implementing in this phase of the project and thus we created a tutorial in our support section to better document this feature.

Figure 5: shows the changed aesthetics of the system. Left is before, Right is after

Figure 6: shows the changed FAQ Page. The top page is before, the bottom pages are after

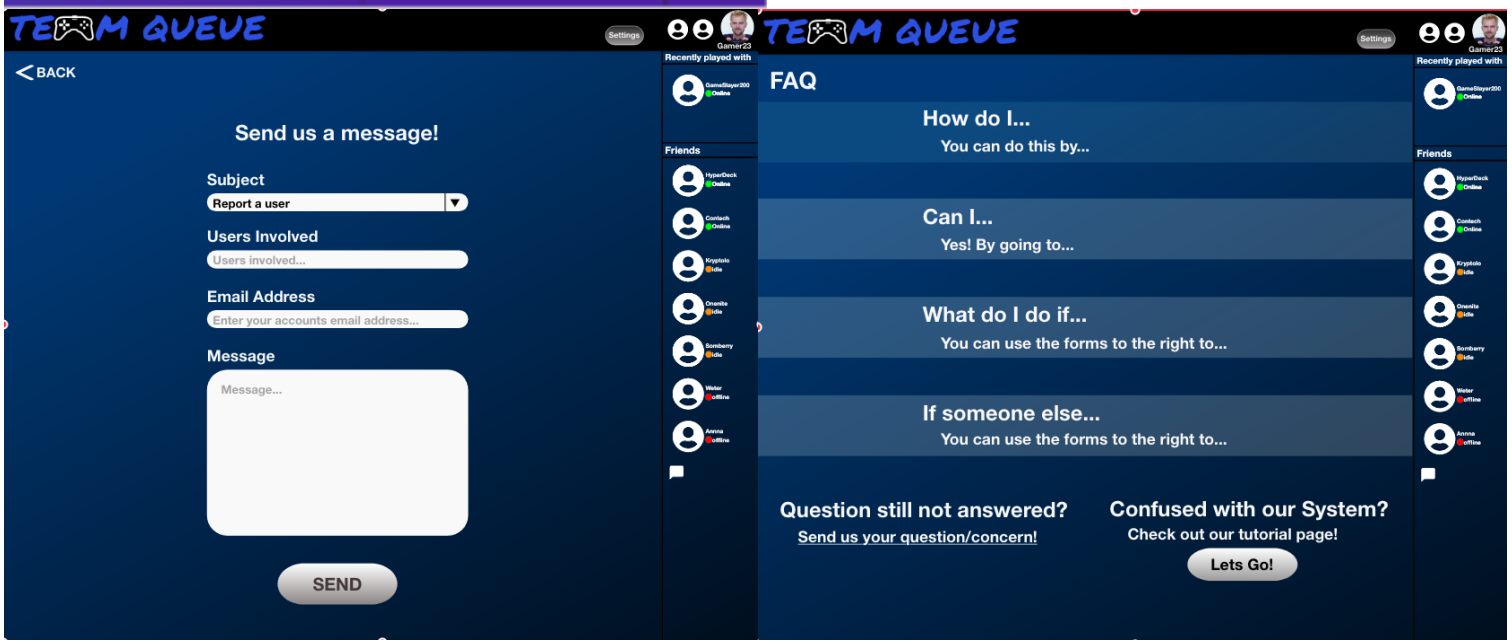


Figure 7: Shows an example of the added tutorial/documentation pages

Be Critical

- Changes:
  - Color scheme change
  - Consistency with design
  - Lines
  - Size
  - Font
  - FAQ Layout
  - Added back button
  - Login validity
- Should Be Done In The Future:
  - Implement remaining games
  - Increased useable friend list
  - Redesign layout keeping in mind changes made
  - Check for valid inputs
- Could Be Done In The Future:
  - Additional games
  - Voice chat
  - Input text
  - Private lobbies
  - Inviting friends
  - Mood
  - Implement light and dark themes
  - Profile Picture

Conclusion

The whole project process proved instrumental towards our application's design development.
With our project idea chosen, narrowing down design problems and brainstorming solutions allowed us to better understand our idea and develop steps towards implementation. Evaluating how our application related to stakeholders and end users allowed us to analyze and review our relationship with specific target audiences. Performing user research provided us with valuable information regarding our end users, furthermore, we gained additional insight by analyzing available competitive products. Consistent design choice re-evaluations, in light of given feedback, allowed us to refine our product design. Lastly, a robust heuristic product evaluation concluded the design refining process, by facilitating a structured review of our products overall user experience.

Appendix:

Vertical Tasks	Horizontal Tasks
Matchmake Automatically	Login
Matchmake Manually	Signup
Edit Profile Page	Access About Page
Settings Page	Direct Messaging Between Friends
(In Lobby) Text Chat	Browse Games
Tutorial/Documentation	Change Settings
	Access FAQ Page
	Support Ticket/Report System
	Text Chat
	Voice Chat
	Set Search Criteria
	Create a Lobby

Changes

- Preference Page

Text Chat

- Simulate sending messages and images
- More user chats

Tutorial/Document

- Add friends
- Making a lobby
- Automating Queue
- Edit Profile
- Chatting

Jase:

- Portfolio

Chuks:

- Demo
- Introduction
- Conclusion

Pragya:

- Executive Summary
- Description of design problem
- Description of design solution
- End-user and stakeholders
- Format Report
- Make design changes

Lougheed:

- User research methods and process
- User research findings
- Important design choice and justification
- Tutorial Page

Brian:

- Text Chat
- Preference Page
- Low-Fi
- High-Fi
- Heuristic Eval
- Design Changes
- Be critical
- Need to see what everyone else changed

DUE DATE: 1st Dec

First Meeting (Thursday during Tutorial)

- Check up on how everyone is doing

Hi-Fi done by the **28th**

Final Meeting (Monday 30th at 4pm-5pm)

Finish Before meeting

- Demo
- Report
- Portfolio
- Go over everything and finalize

Browse page spacing
Browse page line thickness
Images aren't stretched
FOCUS SIMPLICITY
Stay consistent in sizes and style

Fix login logic
Lower word count