

Co-op Queue

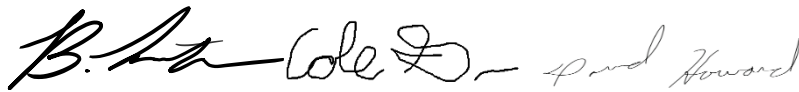
by

Brianna Liston, Cole Glaser, David Howard

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2020 Brianna Liston, Cole Glaser, David Howard

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.



Brianna Liston, Cole Glaser, David Howard

4/23/20

4/23/2020



Abdou Fall

4/23/20

4/23/2020

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2020

Table of Contents

<u>Section</u>	<u>Page</u>
Abstract	1
1. Problem Statement	2
1.1. Introduction	2
1.2. Project Description	2
1.3. Problem	3
1.4. Solution	3-4
1.5. User Profile	4
1.5.1. Project Title	4
1.5.2. Potential Users	4
1.5.3. Software and Interface Experience	5
1.5.4. Experience with Similar Applications	5
1.5.5. Task Experience	5
1.5.6. Frequency of Use	5
1.5.7. Key Interface Design Requirements That the Profile Suggests	6
1.6. Use Case Diagram	6
2. Project Management	7
2.1. Budget	7
2.2. Objectives/Deliverables	7-8
2.3. Project Schedule	9-10
3. Technical Elements	11
3.1. Networking	12
3.2. Application	11-12

Table of Contents (cont'd)

<u>Section</u>	<u>Page</u>
3.3. Security	12
4. Application	13
4.1. Application Architecture	13
4.2. Screenshots	14
4.2.1. Login Page	14
4.2.2. Newsfeed Page	15
4.2.3. Game Details Page	16
5. Test Plan	17
5.1. Overview	17
5.2. Scope	17
5.3. Objective	17
5.4. Entry and Exit Criteria	18
5.5. Test Logging and Reporting	18
5.6. System Testing	18
5.7. Testing Procedures	19
5.8. Pass/Fail Conditions	19
5.9. Schedule of Team Member Testing	20
5.10. Test Cases	20-25

Table of Contents (cont'd)

<u>Section</u>	<u>Page</u>
6. Conclusion	26
6.1. Fall Semester 2019	26
6.2. Spring Semester 2020	26
6.3. Problems Encountered and Analysis of Problems Solved	27
6.4. Future Recommendations/Recommendations for Improvement	27
Appendix A. References	28
Appendix B. Project Poster	29

List of Illustrations

Tables

<u>Number</u>		<u>Page</u>
Table 1.	Project Budget	7
Table 2.	Project Objectives/Deliverables Due Dates	7
Table 3.	Schedule of Team Member Testing	20
Table 4.	Test Case Table	20-25

Figures

<u>Number</u>		<u>Page</u>
Figure 1.	Use Case Diagram	6
Figure 2.	WBS	8
Figure 3.	Fall Project Schedule and Gantt Chart	9
Figure 4.	Spring Project Schedule and Gantt Chart	10
Figure 5.	Application Architecture	13
Figure 6.	Login Page	14
Figure 7.	Newsfeed Page	15
Figure 8.	Game Details Page	16

Abstract

Gaming has become less social than ever, with 67% of gamers strictly playing solo, while the rest are constantly connecting to games full of random players (Forbes). We want to create a way for gamers to reconnect with each other by finding out they have a similar taste in games as their friends. With the influx of new game releases, it can be difficult to keep track who wants to play what. Co-op Queue will be an application that has all new games imported into it, allowing users to tag which games they like. This marks whether or not they're interested in playing that game. Then, once the user indicates that they are interested in a game, it will automatically inform them of their friends who are also interested in the game and give the user an opportunity to message them. Helping more people to fill their co-op queue.

1. Problem Statement

1.1 Introduction

Through the rise in popularity that gaming has endured, there are several side effects that have taken place in the industry. Primarily, there has been a drastic increase in the amount of games that have come out or been announced. These are spread across all genres, and range in quality and size. However, another trend that has been noticed is that gamers are becoming less and less social, with the majority now playing solo. We hypothesize that these two trends are correlated, and that people are having more trouble wading through the massive amounts of games to find ones both they and their friends like. Co-op Queue's goal is to create a platform that allows users to easily sort through games to find not only games they like, but people that also like those games.

1.2 Project Description

Co-op Queue is a web application that will allow users to register for an account. Once a user is registered, they will be able to view information on all recently released and upcoming games. This includes trailers, reviews, news articles, and even screenshots of the game. If they decide they like it they can indicate that, at which point they will be presented with a list of other users who liked that game. They can then send a friend request to another user, and if accepted, message each other about the game and potentially set up a time to play upon the game's release.

1.3 Problem

There are a few applications that allow users to see what games are coming out, but they are usually never that organized. You usually have to search by genre or know the name of the game already, making the discovery of new games a challenge. There are also several places where you can add and talk to friends, but there is no way to know what games they like when you initially add them. We aim to provide a solution to both problems at once.

1.4 Solution

In a modern age where people are constantly connecting to games with a sea of random players, we want to create a way for gamers to connect with each other. Whether it be making new friends or discovering the interests of their current friends, we plan to help guide gamers in reaching out to others. Co-op Queue will be a mobile friendly web application that has all newly released games imported into it. This application will be able to be used seamlessly across Apple, Android, and web browsers, allowing a wide range of access to as many users as possible. A user can mark whether or not they're interested in playing that given game. When the user is interested in a game it will automatically inform them of other people who are also interested in the game and give them the opportunity to either message them or set up an event. This creates a commonality between players and allows them to forge more meaningful relationships that they might not otherwise have.

Co-op Queue's target audience will be the gaming community as a whole. We want this web application to be useful to gamers across all platforms in an effort to bring all gamers together and keep them informed about all that the gaming industry has to offer at any given moment. We think that this will be more valuable than other options due to the fact that it not

only shows what new games are out, it tells you the interests of your friends. While similar products, namely the Steam marketplace, show some new games and show what your friends are currently playing. This does not solve the problem of forming relationships over common interests, due to the fact that it doesn't connect a new game to a user's interests. It allows you to see the games that people are currently playing, but it doesn't not facilitate a connection and makes the end user do a lot more work for a worse result.

1.5 User Profile

There will only really be one type of user interacting with our application, the End User. The target audience of this application will primarily be people who are interested in finding new games, and secondarily those trying to find people to play those games with. These users will be able to choose which games they like, and then find other people who like those games. Then, users can set up a time to play the new game with their newly made friends.

1.5.1 Project Title

Co-op Queue

1.5.2 Potential Users

- Anyone who is interested in discovering new games
- Anyone who would like to find new friends to play games with.

1.5.3 Software and Interface Experience

This application will be geared towards people who have used gaming store fronts before, due to the similar layout and purpose of looking at a game's information. Co-op Queue is also meant for people who have used online chat services, more specifically those where you add friends before you can chat. It is a requirement to understand how friend request sending/accepting works to use the platform effectively.

1.5.4 Experience with Similar Applications

- Steam
- Epic Games Store
- Xbox Live
- Snapchat (Chat Service Specifically)

1.5.5 Task Experience

- Using a web browser or mobile phone to navigate a web application.
- Comfortable adding and messaging people through an online service.
- Understanding of how to navigate aggregates to view different game materials.

1.5.6 Frequency of Use

Whenever users want to discover new games or find new people to play games with. This occurrence can be daily, weekly, monthly, or yearly.

1.5.7 Key Interface Design Requirements That the Profile Suggests

- Easy to pick a game and look at relevant information.
- Easy to choose which games you like and add people who also like that game.
- Simple to see your friends and message them.

1.6 Use Case Diagram

Figure 1: Use Case Diagram presents the application's use case diagram. Due to the fact that Co-op Queue only focuses on functionality for one type of user, we decided that only one-use case was necessary. That user type is the end user, who can use all of the features of the application.

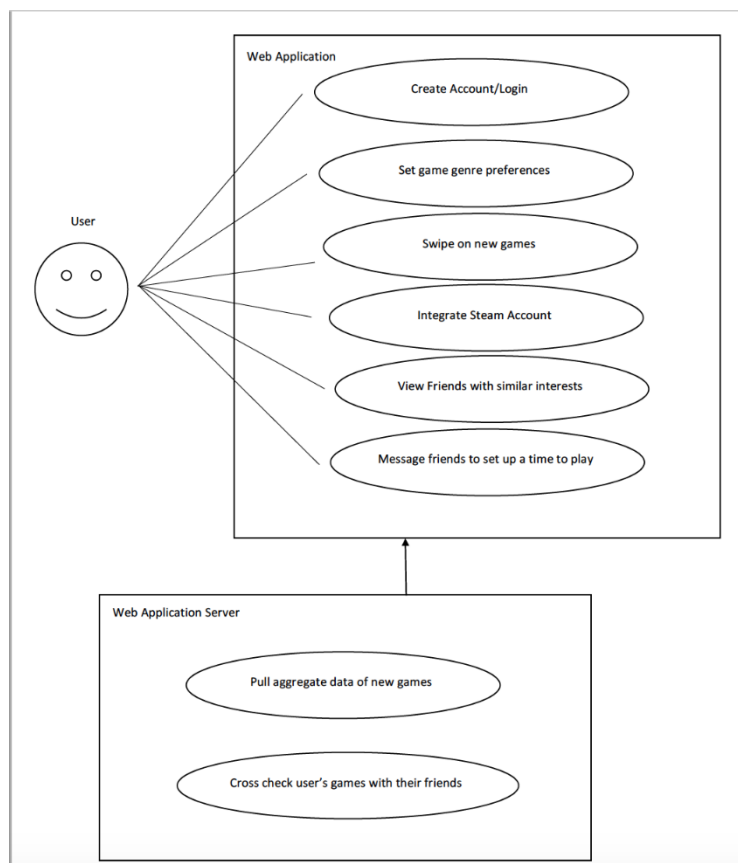


Figure 1: Use Case Diagram

2. Project Management

2.1 Budget

Table 1: Project Budget contains the budget for Co-op Queue. It shows the hypothetical cost for the initial development for the project, if this was a real-world scenario. It contains the labor cost of three developers as well as the costs that are associated with hosting the application and database. Due to the fact we are working for free for this project, the labor costs are not entirely accurate. Also, due to the fact that development was done locally, the hosting costs were lower than they would otherwise be.

ITEM	UNIT #	COST	TOTAL
Labor	600 Hours	\$22	\$13,200
Server Hosting	12 Months	\$8	\$96
TOTAL			\$13,296

Table 1: Project Budget

2.2 Objectives/Deliverables

Table 2: Project Objectives/Deliverables Due Dates contains all of the information in regard to major milestones and their corresponding due dates.

Major Project Milestones					
Fall of 2019 Milestones					
Brainstorming	10/3/2019		Create Data Structure	10/11/2019	
Front End Setup	11/4/2019		Test Case Creation	12/5/2019	
Spring of 2020 Milestones					
Data Postbacks	1/28/2020		Front to Back Connections	2/15/2020	
Microsoft Identity	3/1/2020		SignalR Messaging	3/28/2020	

Table 2: Project Objectives/Deliverables Due Dates

Figure 2: WBS a visual of our plan to complete Co-op Queue.

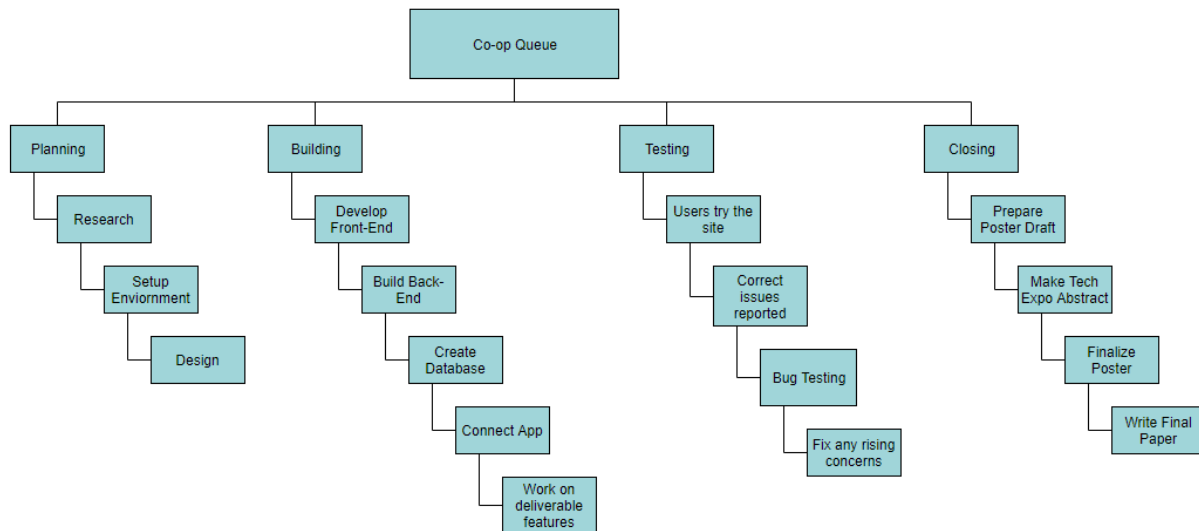


Figure 2: WBS

2.3 Project Schedule

Figure 3: Fall Project Schedule and Gantt Chart is our Fall Semester project schedule with all project milestones and goal dates listed.

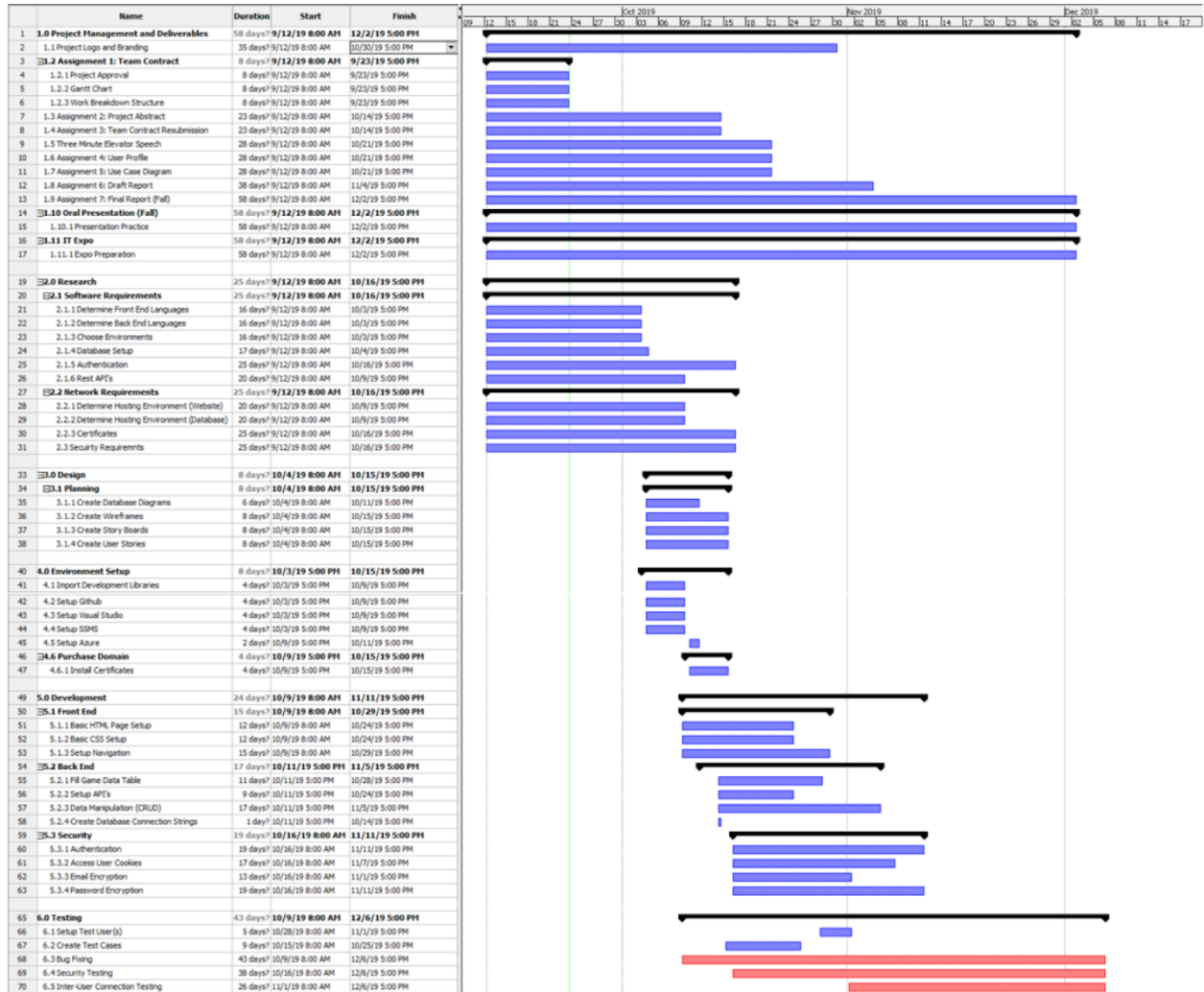


Figure 3: Fall Project Schedule and Gantt Chart

Figure 4: Spring Project Schedule and Gantt Chart is our Spring Semester project

schedule with all project milestones and goal dates listed.

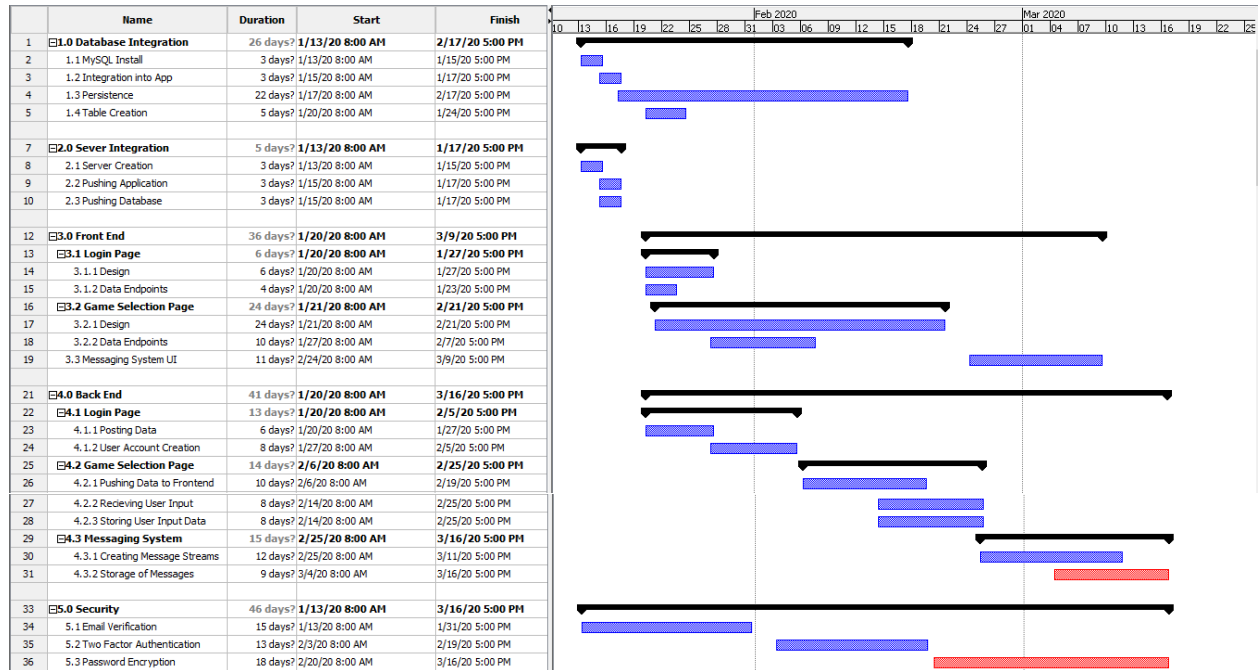


Figure 4: Spring Project Schedule and Gantt Chart

3. Technical Elements

3.1 Network

Our data aggregation will come from other sources, and then be reformatted to fit our own internal data structure. We feel this will make the system more lightweight as well as less taxing, as requesting data to external sources every time a game's information is needed will result in a lot of data being transferred relatively quickly, potentially causing issues. We will be hosting the application on Azure servers and will be using SQL Server for our user and game data storage. We chose Azure because of its reliability being a Microsoft service, as well as its popularity among other projects of variable sizes. Also, due to the fact that we are using Microsoft for nearly all of our application, which makes integration to Azure easier. As for SQL Server, we decided that due to our database knowledge mostly pertaining to relational databases, and our needs for a database primarily being lightweight and functional, we think that SQL Server is the perfect balance between features and being lightweight.

3.2 Application

The Co-op Queue application will be utilizing .Net Core and the Model-View-Controller (MVC) software design pattern for the backend. The main reason for choosing .Net Core, and the .Net framework in general is due to our experience in the framework. Across our team, we have around three years of experience in .Net, making it an easy and comfortable option. As for the MVC design pattern, we chose this due to the data driven nature of our application. The application mainly revolves around displaying data to the user and them making decisions based off it, in return giving us more data. Due to this, we decided that the MVC pattern makes

handling all of these data transfers clean and simple, improving not only the backend code itself but the speed of which the application functions as a whole.

As for Co-op Queue's front end, we will be using Razor to build our pages. Razor pages allow the developer to inject an object into the page through the controller to be manipulated by the front end directly. This makes it easy to create dynamic webpages through C# injection into the front end.

3.3 Security

We will be creating a login feature that allows users to sign into their accounts using the .Net Core Identity. We will be using the default password encryption that is already provided by the .Net Core Identity to encrypt the passwords. And for the password security requirements, we will require a user to have at least 6 or more characters, an uppercase letter, a number, and a special character (for example: @, #, \$, *). As for security with sending friend requests and adding friends, a user will not be able to friend anyone unless a user who receives the friend request adds them as a friend.

4. Application

4.1 Application Architecture

Figure 5: Application Architecture shows the structure of our application, from the data storage and hosting side to the front end that the user will interact directly with.

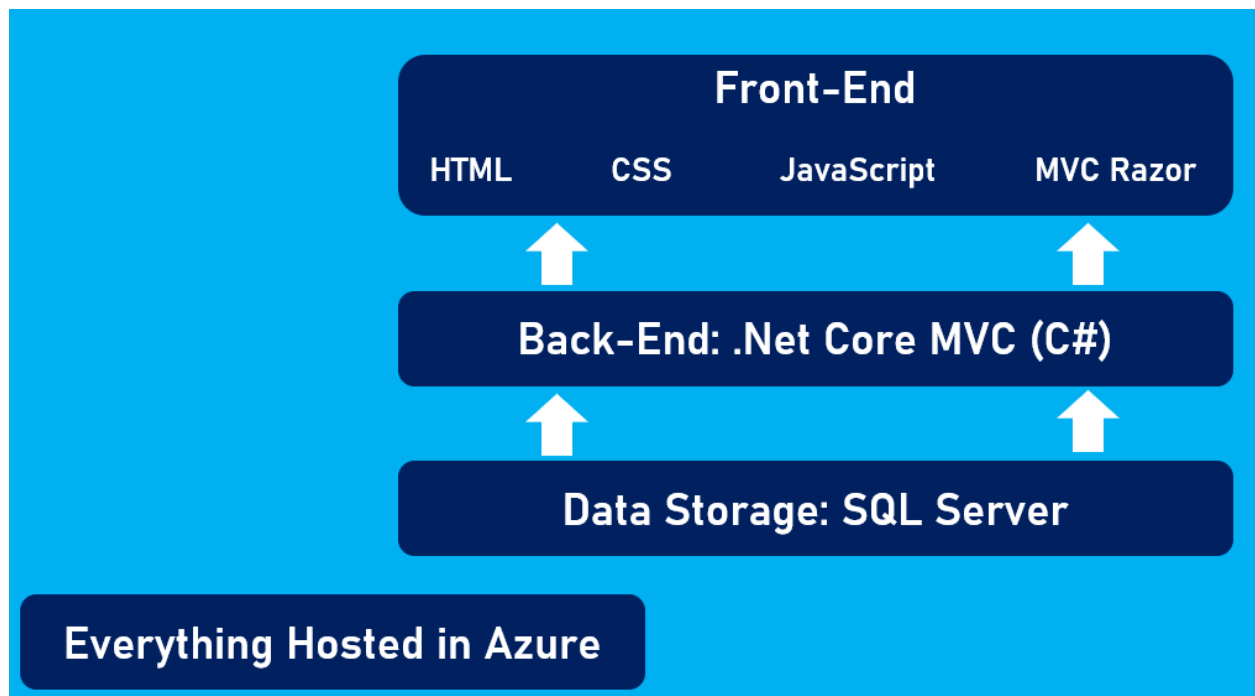


Figure 5: Application Architecture

4.2 Screenshots

4.2.1 Login Page

The login page is the first page the user will see upon reaching our website. Every person has access to this page, as it does not need personal information to use. This page takes a user's information to login, or if they have not signed up before, register to create a new account, as seen in *Figure 6: Login Page*.

Welcome to CoopQueue. Please Login or Register to continue!

The image displays two side-by-side forms for user authentication, separated by a vertical line. The left form is titled 'Login' and contains fields for 'Username:' and 'Password:', each with a corresponding input box. Below these fields is a blue 'Login' button. At the bottom, it says 'Not a user? Click [here](#) to register!'. The right form is titled 'Register' and contains fields for 'Username', 'Email address', and 'Password', each with a corresponding input box. Below the 'Username' field, it states 'Username can only contain letters and numbers'. Below the 'Password' field, it states 'Password must contain the following:' followed by a list of requirements: '6 or more characters', 'An uppercase letter', 'A special character', and 'A number'. At the bottom of the right form is a blue 'Register' button.

Figure 6: Login Page

4.2.2 Newsfeed Page

The newsfeed page contains all of the new games that the user has not declared whether or not they are interested in that game, as shown in *Figure 7: Newsfeed Page*. Each game consists of an icon and a name, to make it super simple for users to figure out which game is which. Clicking on a game's card will take the user to that game's details page.

Queued Games (7)



Figure 7: Newsfeed Page

4.2.3 Game Details Page

As *Figure 8: Game Details Page* displays, the game details page shows in depth information of a specific game. This information consists of a description, screenshots, news articles, trailers, and if the game is released, reviews.

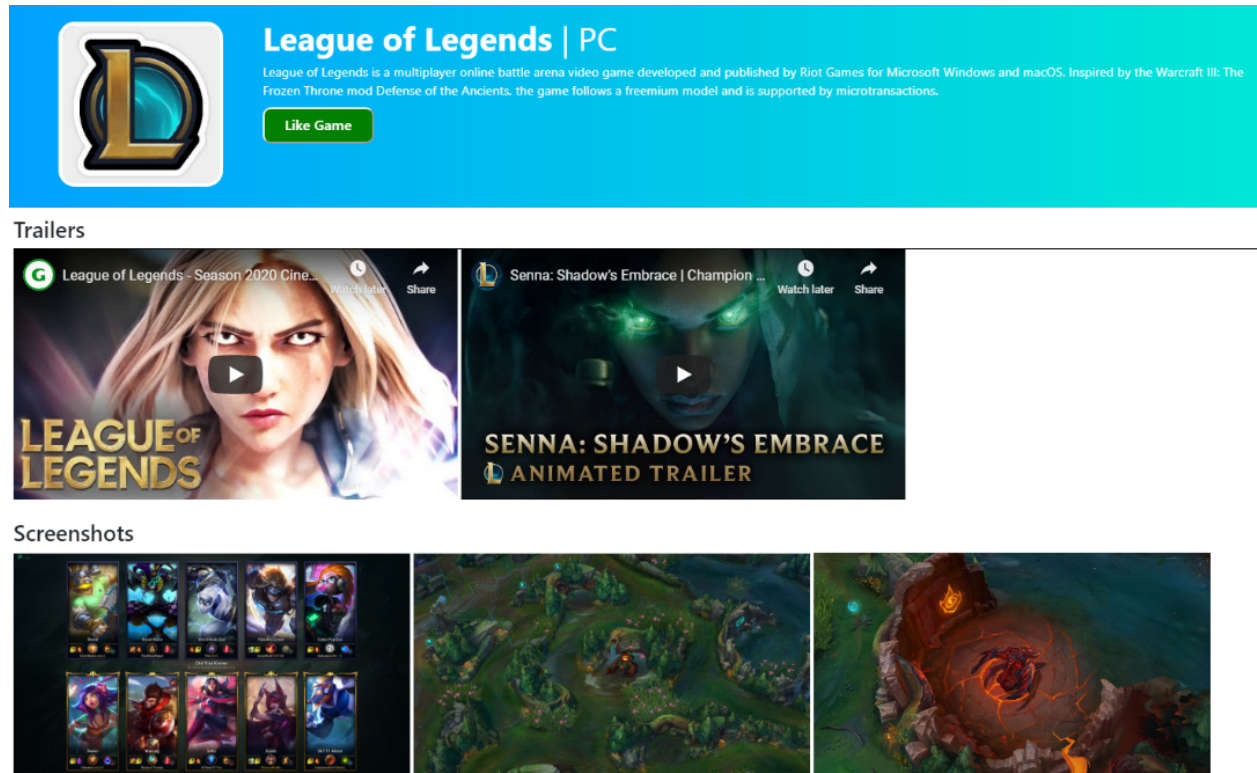


Figure 8: Game Details Page

5. Test Plan

5.1 Overview

Testing is a crucial step to any product being developed during the pre-production phase. In this section, we will explain our test plan procedure for Co-op Queue, and should be used by the following individuals:

- Project Managers
- Developers
- Beta Testers
- Quality Assurance Analysts

5.2 Scope

Testing of all the operations on each page, as well as ensuring that all data retrievals and post backs are working as expected. Testing will be organized by page and pages will be tested in order of how the user is expected to reach them.

5.3 Objective

Verify that each page works as expected. Each button on each page needs to call the function/page that is expected. Each method needs to return/post the correct information from/to the database.

5.4 Entry and Exit Criteria

Entry Criteria:

- Complete build of web application.
- Test Data set up correctly.
- Written requirements to test against.

Exit Criteria:

- All tests pass the correlated requirements.
- Bugs detected and fixed.
- Test cases updated if need be.

5.5 Test Logging and Reporting

Any bugs that are discovered during this process will be documented by whoever discovered them. If it is unclear which portion of the application caused the issue, all developers will meet to discuss the bug and to track down the source. Once completed, the developer responsible with that portion of the application will be delegated to fix the issue. If multiple bugs are assigned to the same developer, they will be addressed in order of severity.

5.6 System Testing

System testing will be performed by the developers of Co-op Queue once while a given portion of the application is being developed, and then again once all major requirements are completed. All developers will test all portions of the app, hopefully allowing for developers not familiar with that portion to catch otherwise easy to miss bugs.

5.7 Testing Procedures

The following are the requirements needed prior for testing to begin:

- Create and document test cases for all major paths.
- Create and document test cases for all edge case scenarios that could be thought of.
- Set up all test data when needed.

The following are the tests that will be used against the Co-op Queue application:

- **Unit Testing** – Developers will unit test portions of the application as they are being developed. This will ensure that each portion in isolation works and will not be documented in the test document.
- **Functional Testing** – These tests will be performed against the application to check to make sure that all requirements are met and that all business rules are implemented.
- **Regression Testing** – This test will be performed at the end of Co-op Queue's development to ensure everything that has been implemented on the application is working correctly.

5.8 Pass/Fail Conditions

A test is marked as successful when the result of performing the test matches the expectations that were set when the test was created. A test has been failed when the expectation does not match the actual result.

5.9 Schedule of Team Member Testing

The schedule of developer testing is presented in *Table 3: Schedule of Team Member Testing*.

Team Member	Timeline	Frequency
Cole Glaser	2/1/2020 - 3/18/2020	Weekly
Bri Liston	2/1/2020 - 3/18/2020	Weekly
David Howard	2/1/2020 - 3/18/2020	As Needed
Beta Testers	2/1/2020 - 3/18/2020	Bi-Weekly

Table 3: Schedule of Team Member Testing

5.10 Test Cases

An exhaustive list of all of our test cases is presented in *Table 4: Test Case Table*.

Tester	Test Case	Feature	Input	Expected Output	Actual Output	Pass/Fail	Reasons for Failure	Date
QA Analyst and Project Manager	TC1	Login/Registration page	As a user, when registering my account, I will see an error message when I do not meet the password requirements.	An error message should come up.	An error message comes up.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC2	Login/Registration page	As a user, when registering my account, I will see an error message when I do not meet the Username requirements.	An error message will show up	An error message comes up	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC3	Login/Registration page	As a user, when registering my account, I will see an error page when username	An error message will show up	An error page showed up	Fail	“Argument Null Exception: Value cannot be null. Parameter name: password”	4/5/20

			and email text box is filled, but not password.					
QA Analyst and Project Manager	TC4	Login/Registration page	As a user, when registering my account, I will see an error message when username and password text box is filled, but not email.	An error message will show up	An error page showed up,	Fail	“Argument Null Exception: Value cannot be null. Parameter name: email”	4/5/20
QA Analyst and Project Manager	TC5	Login/Registration page	As a user, when registering my account, I will see an error message when email and password text box is filled, but not username.	An error message will show up	An error message comes up.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC6	Login/Registration page	As a user, when registering my account, I will see an error message when email text box is filled, but not username and password.	An error message will show up	An error page showed up,	Fail	Argument Null Exception: Value cannot be null. Parameter name: password”	4/5/20
QA Analyst and Project Manager	TC7	Login/Registration page	As a user, when registering my account, I will see an error message when username text box is filled, but not email and password.	An error message will show up	An error page showed up,	Fail	Argument Null Exception: Value cannot be null. Parameter name: password”	4/5/20
QA Analyst and Project Manager	TC8	Login/Registration page	As a user, when registering my account, I will see an error message when password text box is filled, but not username and password.	An error message will show up	An error page showed up,	Fail	Argument Null Exception: Value cannot be null. Parameter name: email”	4/5/20
QA Analyst and Project Manager	TC9	Login/Registration page	As a user, when logging in my account, I will see an error message when username text box is filled, but not password.	An error message will show up	An error page showed up,	Fail	Null Exception: Value cannot be null. Parameter name: providePassword”	4/5/20
QA Analyst and Project Manager	TC10	Login/Registration page	As a user, when logging in my account, I will see an error message when password	An error message will show up	An error page showed up,	Fail	Null Exception: Value cannot be null. Parameter name: userName”	4/5/20

			text box is filled, but not username.					
QA Analyst and Project Manager	TC11	Login/Registration page	As a user, when registering my account, I will be sent to the login page.	A login page will come up	I was sent to the login page	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC12	Login/Registration page	As a user, when logging in my account, I will see an error message when password and username text box is not filled.	An error message will show up	An error page showed up,	Fail	Null Exception: Value cannot be null. Parameter name: userName”	4/5/20
QA Analyst and Project Manager	TC13	Login/Registration page	As a user, when logging in my account, I will see an error message when I use a non-existing username	An error message will show up	An error message comes up	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC14	Login/Registration page	As a user, when logging in my account, I will be directed to the homepage.	I should be directed to the homepage	I was sent to the homepage	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC15	Coop-Queue, Homepage (Newsfeed)	As a user, from the Homepage, can navigate through a of the tabs located on the right side of the page: Newsfeed, Profile, Friends, Likes, Messages.	I should be able to navigate through all of the tabs.	I navigated through all of the tabs.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC16	Coop-Queue, Homepage (Newsfeed)	As a user, from the Homepage, I will be able to select one of the queued games.	I can select all of the queued games	All the queued games can be selected.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC17	Coop-Queue, Homepage (Newsfeed), Queued Games	As a user, after selecting all of the queued games, I could select and view all contents provided	I should be able to select all of the queued game and view all of the contents, such as news, reviews, videos, and screen shots.	It could all be viewed,	Pass	N/A	4/5/20
QA Analyst and	TC18	Coop-Queue, Profile	As a user, I should be able to see my profile	Profile name should be visible	Profile name is visible.	Pass	N/A	4/5/20

Project Manager			name under profile.					
QA Analyst and Project Manager	TC19	Coop-Queue, Profile	As a user, I should be able to edit my profile by changing profile picture and edit my profile bio.	The profile picture and bio can be edited	The profile picture and bio can be edited.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC20	Coop-Queue, Homepage (Newsfeed), Queued Games	As a user, I should be able to click “Like Game” on a game I am interested in and see who else is also interested in the game.	I can click the “Like Game” button for the game I am interested in and be directed to a page showing other people who are interested in your chosen game	I liked the game I am interested in and been brought to a page that shows people who are also interested in the game.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC21	Coop-Queue, Homepage (Newsfeed), Queued Games, [Chosen Liked Game]	As a user, I should be able to send a friend request by clicking “Add Friend”	I should be able to send a friend request to a person I’m interested in being friends with.	I clicked “Add Friend” and the requested should be send to the person I am interested in being friends with.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC22	Coop-Queue, Friends	As a user who received a friend requested from a game I am also interested in, I should be able to accept his/her friend request.	I should be able to add pending friend requests.	I added a pending friend request.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC23	Coop-Queue, Friends	As a user, I should be able to go to friends and click “Send Message” to message a friend.	I should be able to send a message to a friend.	I click send message, but it doesn’t do anything.	Fail	I cannot send a message to a friend.	4/5/20
QA Analyst and	TC24	Coop-Queue, Friends	As a user who wants to delete a friend, I should be able to go their	The friend should be removed	The friend gets removed.	Pass	N/A	4/5/20

Project Manager			profile and click “Remove Friend”	from the friends list.				
QA Analyst and Project Manager	TC25	Coop-Queue, Likes	As a user, under “Likes” tab, I should be able to view my “Liked” games.	I should be able to view my “Liked” games.	I can view my “Liked” games.	Pass	N/A	4/5/20
QA Analyst and Project Manager	TC26	Coop-Queue, Messages	I can send messages to friends	I should be able to send messages to friends	I can’t send messages to friends	Fail	Can’t post Messages	4/5/20
Beta Testers	TC27	Coop-Queue, login	As a user, I should be able register an account and login into the website with no difficulty.	Register an account and login with ease		Pass	N/A	4/5/20
Beta Testers	TC28	Coop-Queue, Homepage (Newsfeed)	As a user, I should be able to easily navigate through the website with ease.	Easily navigate through the website with problems.	A few errors: send message doesn’t work	Fail	A button, “Send Message” doesn’t work.	4/5/20
Beta Testers	TC29	Coop-Queue, Homepage (Newsfeed)	As a user, from the Homepage, I will be able to select one of the queued games I am interested in and “like” the game	I can select any of the queued games, and “like” it.	Can select interested games and like it.	Pass	N/A	4/5/20
Beta Testers	TC30	Coop-Queue, Homepage (Newsfeed), Queued Games, [Chosen Liked Game]	As a user, I should be able to send a friend request to a person who I am interested in being friends with.	I can send a friend request and be added as a friend on the friends list.	Can send friend requests.	Pass	N/A	4/5/20
Beta Testers	TC31	Coop-Queue, Profile	As a user, I should be able to edit my profile by changing profile picture and edit my profile bio.	The profile picture and bio can be edited	Can edit profile	Pass	N/A	4/5/20
Beta Testers	TC32	Coop-Queue, Homepage (Newsfeed), Queued Games	As a user, I should be able to view the games and check out the news, review, trailers, screen shots.	I can review the contents of each game.	Can check out all of the contents.	Pass	N/A	4/5/20
Beta Testers	TC33	Coop-Queue, Likes	As a user, under “Likes” tab, I	I should be able to view	I can view my	Pass	N/A	4/5/20

			should be able to view my “Liked” games.	my “Liked” games.	“Liked” games.			
Beta Testers	TC34	Coop-Queue, Messages	I can send messages to friends	I should be able to send messages to friends	I can’t send messages to friends	Fail	Can’t post messages	4/5/20
Beta Testers	TC35	Coop-Queue, Friends	As a user who received a friend requested from a game I am also interested in, I should be able to accept his/her friend request.	I should be able to add pending friend requests.	I added a pending friend request.	Pass	N/A	4/5/20
Beta Testers	TC36	Coop-Queue, Friends	As a user, I should be able to go to friends and click “Send Message” to message a friend.	I should be able to send a message to a friend.	I click send message, but it doesn’t do anything.	Fail	Button doesn’t work.	4/5/20

Table 4: Test Case Table

6. Conclusion

6.1 Fall Semester 2019

The main focus up until this point has been feeling out all of our options and potential issues that could occur down the line. We now know what frameworks and technical aspects we need to use and are currently educating ourselves on the security aspects as well as the hosting aspects. We will start off with standing up the project in Azure and connecting all the pieces together. Once that is done the plan of attack is to work on the pages in order of how the user will encounter them, from login page to game selection page to messaging functionality for both the front and back end. This will help stay organized as we employ a front to back approach for the main application development. The security aspect will run alongside this main development pattern so that we can work independently, yet together. We feel like this will make our project development time efficient while still working together in a productive manner.

6.2 Spring Semester 2020

Once spring semester hit, the development portion of the project went into full swing. The stored procedures, database calls, controller methods, and front-end functions have been tweaked and improved constantly. As our knowledge evolved, so did our application, causing some code to go unused and deprecated. The front-end pages had considerable improvements, starting with the removal of ReactJS. We decided that due to compatibility issues with Razor pages, it would be best to go with a more traditional route as far as styling goes. Along with this, there was a significant redesign on the front end to make Co-op Queue more mobile friendly.

6.3 Problems Encountered and Analysis of Problems Solved

Our first major problem is what platforms to support. We were originally going to integrate Xbox, PlayStation, and Steam friends lists into our application, but quickly found out both Xbox and PlayStation do not have public API's. Due to this, we decided to make our system internal only, with no integrations at all. We feel that this was a good solution for the time, but in the future, we would gear the application towards Steam and implement that. We also had issues setting up Microsoft's identity user for user accounts, but finally got that settled. Due to technical issues and time, we weren't able to set up email confirmation and two-factor authentication. The last of our major issues was setting up SignalR for the user messaging. SignalR allows the application to set up a live stream with the database and receive updates without making a call to it directly. This makes it so that messages show up in real time. However, we could not get it working properly in time.

6.4 Future Recommendations/Recommendations for Improvement

If we were going to continue this in the future, we would start off with refactoring the backend. As we learned throughout the project, we changed the structure of the backend, but left some artifacts of the original structure that are still there. we would also refactor it further to improve on the new code. On top of this, we would probably gear the application towards Steam, and integrate Steam's friends list. That way it allows for a more seamless transition from our application to the platform the game will be played on.

Appendix A. References

[1] “Investing in the Soaring Popularity of Gaming,” 2018 [Online]. Available: https://www.reuters.com/sponsored/article/popularity-of-gaming?utm_source=reddit.com.

Paul, Ian. “What Is Two-Factor Authentication, and Which 2FA Solutions Are Best?” *PCWorld*, PCWorld, 5 June 2019, <https://www.pcworld.com/article/3225913/what-is-two-factor-authentication-and-which-2fa-apps-are-best.html>.

Appendix B. Project Poster

Figure 9: *Coop-Queue Poster* is our poster of the project that will be used to show case our project for the Expo.

Team 24: Cole Glaser, Bri Liston, David Howard
Technical Advisor – Abdou Fall
College of Education, Criminal Justice, and Human Services
School of Information Technology

Co-op Queue

University of CINCINNATI

About

Co-op Queue is a mobile friendly web application that connects people who have similar interest in games.

- ✓ Co-op Queue **allows users to pick** which video games they like.
- ✓ We **connect users** based off those choices to other users who also are interested in those games.
- ✓ Users can message each other to **set up events** to play the games they connected through.

Technologies used by Co-op Queue


Hosted with Azure


Database


Written in C#


.NET Framework

Problem

There is no current platform that **both informs users** of game releases and **connects them**.

Solution

Co-op Queue provides a platform for users to not only **be informed** of game releases, but **connect** off those games

Implementation

Co-op Queue uses Microsoft Azure to host the database and website and uses the following technologies to improve the experience

- Bootstrap to aid in mobile friendliness and scalability
- Model-View-Controller design pattern to keep the data clean and the business logic segregated

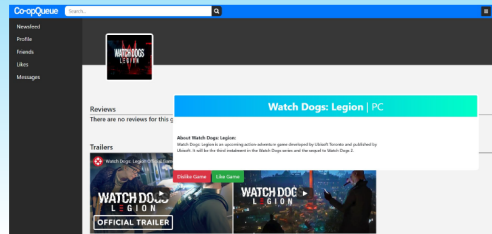
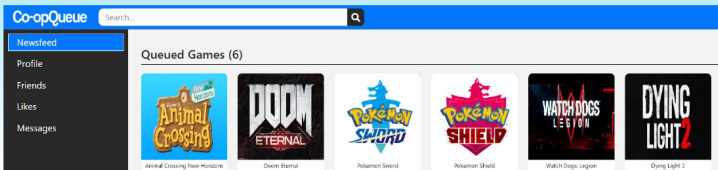


Figure 9: *Coop-Queue Poster*