CEN 4010 Principles of Software Engineering: Fall 2021

Love and Nerds (LAN) Party

Group 9

Milestone 3

Date:October 25th 2021

Revision History

**Team Lead:**

Trang Leminh - *tleminh2012@fau.edu*

**Team Members:**

Tierney Mathis - *tmathis1@fau.edu*

Raven Ruggs - *rruggs2018@fau.edu*

Bethlyn Joseph - *bethlynjosep2019@fau.edu*

Olivia Razzo - *Orazzo2019@fau.edu*

Kevin Young - [*kevinyoung2018@fau.edu*](mailto:kevinyoung2018@fau.edu)

# Vertical prototype: Account creation - Registration

* Link to vertical prototype:
  + <https://lamp.cse.fau.edu/~cen4010_fa21_g09/vert_prot/registerfinal.php>
* Features implemented
  + Account creation with error checking
    - Username can include numbers, special characters, and letters
    - Name can only consist of letters and/or white space
    - Only valid email format is allowed
    - Passwords must be at least 8 characters long, including at least one number, letter, and caps.
  + All fields are required
    - If any fields are empty when user submits, error will be returned, account cannot be created
  + User login
    - User can log in using same credentials during registration
      * They will be redirected to a simple dashboard, displaying their username
* Functionality enhancements in next sprint
  + Email verification functionality
  + Error checking for re-entering password correctly

# Table of Contents

[**Vertical prototype: Account creation - Registration**](#_pihjls8125mk) **2**

[**Table of Contents**](#_yddo0t5zscnh) **3**

[**Executive Summary**](#_ctphjtj5xjsx) **4**

[**Overview and Use Cases**](#_nvp7bqmjqdtn) **5**

[**Data Definitions**](#_ao8pkpsxjsaz) **11**

[**List of High-Level Functional Specifications**](#_daix246417hi) **13**

[**List of Non-Functional Specifications**](#_qn5br9e2c24t) **16**

[**High-Level System Architecture and Database Organization**](#_ump8inmorzpk) **17**

[**Use-case Diagram**](#_b5ecu67h8vpy) **23**

[**Component and Deployment Diagram**](#_7db2yef156v0) **24**

[**Identify key risks for your project at this time**](#_bfqmq9bcw1zm) **25**

[**Competitive Analysis**](#_6s248lslc84x) **26**

[**Team Roles**](#_c782a7co4f6v) **33**

# Executive Summary

In the past few years the world has gone through trying circumstances that have led to a plethora of mental health issues. One of the aspects of mental health that many individuals experience today due to the Coronavirus pandemic is crippling loneliness. This is generalized as an intense feeling of being sad or unhappy about being socially isolated. With the looming threat that at any moment someone can be infected with a life-threatening disease; many people have avoided any or most in-person activities.

Some individuals may opt-out of leaving the house entirely in favor of staying in and escaping in the virtual world of their favorite video game genre. However, that escapism may not be sufficient enough to give the gamer a sense of filling the hole that loneliness has left. It is possible that even in a co-operative or multiplayer game: the gamer may feel a sense of being alone if not emotionally connected to another player.

That is where Group 9’s idea plays a role. We propose to make a website or application that will bring like-minded individuals together. This will be a social site that is catered towards gamers and will either act as a mostly dating site or a hub where gamers can meet other people to play games with in parallel. The app will have functions such as: making a profile, sorting by distance, sorting by specific game, sorting by realms (dataserver where game characters are on), sorting by genre, sorting by platform (computer, mobile, or console), sorting by class roles (tank, healer, damage), and possibly more.

The overall objective of this site is to potentially cause a spark between gamers and eventually help them feel less alone. Long term goals would be to get the gamers involved with the other person which may lead to in-person dating or meet-ups if/when they feel comfortable.

# Overview and Use Cases

Love and Nerds Party or LAN party is a gamer-centric, social media dating site aimed to bring like-minded individuals together who are interested in meeting potential love-interests or joining collective groups of people who are simply looking for new friends to play with. The platform will serve two primary functions for users; the user can choose to seek out other singles with similar interests or game-centric groups consisting of multiple members. Users should be able to customize their profile with the minimum information of game genre interests, game-specific interests, personal information as well as their profile picture. Users will be suggested matches or interest groups that match either their specific interests, where they can choose to either match and chat with individuals or join the common interest group. Upon joining interest groups, members can post media and content to the group page, with the moderation of the group administrator. Users can also search for groups based on game titles or genres. If certain groups for games do not exist, users can create a new group and invite individuals to join.

## Use Case 1: Finding the one

The user loads into the homepage and wants to start matching with other singles. The user will be shown matches. Then the user will start chatting with matches through a chatroom.

### Description:

This use case will describe how the user will start chatting with other singles matching their interests.

### Actors:

1. User
2. System

### Preconditions:

1. Server is online
2. User is connected to the Internet
3. User has a registered account
4. User is signed into their account

### Primary Flow of Events:

1. On home screen, user is selects one of two buttons, “Looking for the one”
2. System checks if user is currently signed in
   1. If user is not signed in, user will be directed to Alternative Flow 1
3. User is displayed list of matches by profile name
4. User selects a matched profile
   1. If user is displayed empty list, Alternative Flow 3 is followed
5. A chatroom is initiated between the matched pair
6. Terminate use case: Finding the one

### Alternative Flows:

1. User is not signed in
   1. System verifies user is not signed
   2. System requests login credentials from user
      1. If user does not have a registered account, alternative flow 2 is followed
   3. System validates login credentials
      1. If invalid login credentials, user is returned to step 1.2
   4. End of Alternative Flow 1, user is returned to Primary Flow step 3
2. User does not have a registered account
   1. User is displayed “Create an account” option
   2. User presses button
      1. If user does not proceed, user is directed to Primary Flow step 6
   3. System requests user to input new profile information into form
   4. User inputs new login information
   5. User presses submit
   6. System stores new member information to database
   7. System redirects user to profile customization page
   8. User selects custom interests, preferences.
   9. System saves new account information to member profile in database
   10. End of Alternative Flow 2, user is redirected to Alternative Flow 3
3. User is displayed suggested member profiles
   1. System displays profiles matching user
   2. User presses on profile
   3. Suggested profile summary is shown
   4. User selects “Match” with suggested profile
      1. If user selects “Pass”, user is returned to Alternative Flow step 3.1
   5. End of Alternative Flow 3, return to Primary Flow step 3
      1. If no matches, user is returned to Alternative Flow step 3.1

## Use Case 2: Searching for Group

The user loads into the homepage and wants to search for groups to join based on game titles. The user will utilize the search bar to input desired game titles. Then the user will be presented with a list of all results matching search criteria. User can choose to join the page after visiting group homepage

### Description:

This use case will describe how the user will start or continue a search for people or groups matching their interests and then send a request to join.

### Actors:

1. User
2. System

### Preconditions:

1. Server is online
2. User is connected to the Internet
3. User has a registered account
4. User is signed into their account

### Primary Flow of Events:

1. On home screen, user is selects one of two buttons, “Looking for a group”
2. System checks if user is currently signed in
   1. If user is not signed in, user will be directed to Alternative Flow 1
3. User enters game title into search bar
   1. If game title does not exist in database, user is directed to Alternative Flow 3
4. System displays groups containing input titles.
5. User selects group card
6. User is displays group information page
7. User presses “Request to Join Group”
8. System sends request to group owner
9. Terminate Use Case 2: Searching for Group

### Alternative Flows:

1. User is not signed in
   1. System verifies user is not signed
   2. System requests login credentials from user
      1. If user does not have a registered account, alternative flow 2 is followed
   3. System validates login credentials
      1. If invalid login credentials, user is returned to step 1.2
   4. End of Alternative Flow 1, user is returned to Primary Flow step 3
2. User does not have a registered account
   1. User is displayed “Create an account” option
   2. User presses button
      1. If user does not proceed, user is directed to Primary Flow step 6
   3. System requests user to input new profile information into form
   4. User inputs new login information
   5. User presses submit
   6. System stores new member information to database
   7. System redirects user to profile customization page
   8. User selects custom interests, preferences.
   9. System saves new account information to member profile in database
   10. End of Alternative Flow 2, user is redirected to Primary Flow Step 3
3. User enters invalid characters into search
   1. System notifies user of invalid input (symbols, punctuation)
   2. End of Alternative Flow 3, user is directed to Primary Flow step 3
4. User enters game title that does not exist in system database
   1. System notifies user that group for game title input does not exist
   2. System performs Use Case: Creating a Group
5. User is suggested groups to join based in profile
   1. System displays suggested groups to user based on profile matches
   2. End of Alternative Flow 5, user is redirected to Primary Flow Step 3

## Use Case 3: Creating a Group

The user loads into the homepage and wants to create a new group based on a game title or genre. The user selects the button to create a new group. The user will fill out a group creation form providing information about the group. User will submit the form. The system will save the new group information to the database.

### Description:

This use case will describe how the user will create a group for a game title.

### Actors:

1. User
2. Group Admin
3. System

### Preconditions:

1. Server is online
2. User is connected to the Internet
3. User has a registered account
4. User is signed into their account

### Primary Flow of Events:

1. User selects “Create a Group”
2. System checks if user is currently signed in
   1. If user is not signed in, user will be directed to Alternative Flow 1
3. User fills in group information form
4. User presses “Create group” to submit
5. System saves new group information to database
6. User is invited to add people to group
7. Terminate Use Case 3: Creating a Group

### Alternative Flows:

1. User is not signed in
   1. System verifies user is not signed
   2. System requests login credentials from user
      1. If user does not have a registered account, alternative flow 2 is followed
   3. System validates login credentials
      1. If invalid login credentials, user is returned to step 1.2
   4. End of Alternative Flow 1, user is returned to Primary Flow step 3
2. User does not have a registered account
   1. User is displayed “Create an account” option
   2. User presses button
      1. If user does not proceed, user is directed to Primary Flow step 6
   3. System requests user to input new profile information into form
   4. User inputs new login information
   5. User presses submit
   6. System stores new member information to database
   7. System redirects user to profile customization page
   8. User selects custom interests, preferences.
   9. System saves new account information to member profile in database
   10. End of Alternative Flow 2, user is redirected to Primary Flow Step 3

## Use Case 4: Posting Content to a Page

User wants to post content to a group page as a member. The user will select to create a new post on the group page. The user will fill in the text field with text, attach content, link embedded content. The user will submit the final post and the system will send the post request to the group owner and save the new post to display on the group page.

### Description:

This use case will describe how the user will post media and content to a group page.

### Actors:

1. User
2. System

### Preconditions:

1. Server is online
2. User is connected to the Internet
3. User has a registered account
4. User is signed into their account
5. User is a member of the current group

### Primary Flow of Events:

1. User navigates to a group page
   1. If user is not a member of the group, Alternative Flow 1 is followed
2. User select “Create a new post”
3. User fills in text field with text, attaches content, links embedded content
4. User selects the “Submit Post”
5. System sends post request to Group Admin
6. Group Admin accepts post submission
7. System displays user’s post to group page
8. Terminating Use Case 4: Posting Content to a Page

### Alternative Flows:

1. User is not a member of the group
   1. User is displays group information page
   2. User presses “Request to Join Group”
   3. System sends request to group owner
   4. Alternative Flow 1 is terminated

# Data Definitions

Logging in- how user first interacts with the app and either signs in or makes a profile

Group page- where posts are shared amongst a group and posted can be added or commented on

User interaction- How the user interacts with other users in the app for example liking another user's profile or chatting with other users.

Liking- An interaction with another user profile to show interest in the other user. This can lead to chatting with the other user. Option to pass a user if they don’t have common interests.

Chatting- This is a function where two users can interact with each other by messaging. This can lead to the user going off the app like playing games and interacting in person after meeting through the app.

Notifications- These will notify the user if there are any activities on the app like messages, events, or likes.

Matching- After both users like each other's profiles, they will receive a notification that they are matched and that they can begin messaging each other.

User profile- The user can add pictures of themselves or things that represent them and add a short description of themselves in words

Group matching- if you and another user like the same person you can consider forming a group and start a group chat and interact that way.

| **Name** | **Meaning** | **Usage** | **Comment** |
| --- | --- | --- | --- |
| User interaction | Service | Site user service | User interacting with others |
| Liking | Service | Site user service | Interest in another user |
| Chatting | Service | Site user service | Users talking to others |
| Notifications | Service | Site user service | Notify users |
| Matching | Service | Site user service | Matching each user |
| User profile | Service | Site user service | User introducing themselves |
| Group Matching | Service | Site user service | A form of matching for groups |
| Group Page | Service | Site user service | A forum for the group and form of chatting for the group |
| Logging in | Service | Site user service | How user starts the app |

# List of High-Level Functional Specifications

**Non-member**

1. **Create an account using registration form at the login page (Priority - 1)**
   1. Users will be able to create an account, providing information like username, password, name, and email. If the user doesn’t create an account, none of the member features will be available for the user to use.
      1. **Features that non-members CANNOT use unless signed into an account**
         1. Creating and joining a group
         2. Viewing/posting content on a group’s page
         3. Editing user profile
         4. Being matched with other users
         5. Chatting with users
         6. Searching for groups or games using the search bar
         7. Ability to log out of account
   2. The user will have to verify their newly created account using their email that they associated their account with.
   3. After an account is created, the site will direct the user back to the site’s home page for the user to sign into their account. Once the user signs into their account, the user’s profile page will be displayed and the user will now be able to use the member features.
2. **Creating a group**
   1. Lead to login page so that the user can create an account or sign in to their account to use this feature

**Member**

1. **User will sign into their account (Priority - 1)**
   1. **User will sign into their account** using their username and password
   2. **User profile (Priority - 1)**
      1. User can view their profile page by clicking on the person icon at the top right corner of the the navigation bar
      2. The user profile will display the user’s name, age and location (optional), about me section (showing interests and biography), pictures, and groups they are a part of. Other user profiles will be displayed the same.
      3. User will be able to edit their profile, changing information like their name, biography, profile picture, gaming interests, preferences/interests
      4. An ‘Edit profile’ button or icon will be displayed
   3. **Looking for a group (Priority - 2)**
      1. Use the search bar to search for groups based on game titles or game genres
         1. A list of group suggestions will be displayed to the user
         2. Each group that is listed will each display a group name, group description, and a group picture. If the user would like to join a group, a ‘Request to Join’ button will be shown next to the group name that the user can click and the request will be sent to the group’s creator. The user can also click on the group’s name to view the group’s page or hover over the picture to see a small preview of the group’s page.
         3. Groups that are under the same game genre will also appear in the list
   4. **Looking for a partner (Priority - 1)**
      1. Match suggestions (recommender system) will appear and match a user with another user based on similar preferences, interests, games from the profile
         1. The user will have the option to accept or decline a match. If a user declines a match, they won’t be able to chat with that user and that user’s profile will disappear from the suggestions list. If a user accepts a match, the other user will be sent a match request. The user can hover over the user’s profile icon to see a small preview of a user’s profile. Clicking on a user’s profile from the suggestions can lead the user to the user’s profile page.
         2. When both users are matched together, a success message will display showing both users are matched, and the users will be given the options to chat now or later
         3. If no matches appear, a suggestion will be made to join a group to continue making connections with other users
      2. Profile suggestions (recommender system) will appear to check out a user’s profile based on similar preferences, interests, or games from the profile
   5. **Create a group (Priority - 2)**
      1. Users can create groups to make connections with other users that have similar gaming interests
         1. The user will have to use the create group form to start the process of creating a group
            1. Fill in information (e.g. group description, game, group picture, game tags, member status, etc.) about the group the user is creating
      2. Once the form is submitted and the creation of the group is successful, a success message should appear to alert the user that the group is created, otherwise, an error message will tell the user what box they didn’t complete or the information that was inputted is incorrect
      3. The newly created group will appear under group/game searches and suggestions. If a group is set to private, the membership request will be sent to the group’s admin and the user will have to wait until the admin accepts the request for the user to join the group. If a group is set to public, the user will automatically join the group and gain access to post or interact with other users in the group’s page.
      4. A page will be created for the group, displaying the group’s name, picture, and group information (e.g. description and tags), and posts made by users. There’s game recommendations that might interest the user to check out and group recommendations to join other groups that have similar tags, making it possible for the user to make more connections.
   6. **Post content to a group’s page/chatroom (Priority - 3)**
      1. In a group’s page, the user can send links and videos, or make posts.
         1. Post is saved and displayed on the page after clicking on ‘Submit Post’, post will be seen by other users
         2. Users in the group can interact with posts by liking or commenting in the post thread or they can create their own post
      2. In a chatroom, users can send links to each other
   7. **Notifications (Priority - 1)**
      1. User will be alerted when they receive acceptance from group requests and successful matches
      2. Users will be notified of chat activity
      3. Notifications will appear at the top right corner of the navigation bar
   8. **Chatrooms (Priority - 1)**
      1. Two users will be able to chat with each other through private messages or a private chatroom once they are matched with each other
      2. The user can click a chat message from the chat notification to go to the chatroom
      3. In the chatroom, messages will be displayed between the two users. A back arrow will be displayed for when the user wants to exit the chat and go back to the previous page.
2. **User can log out of their account (Priority - 1)**
   1. A ‘Sign out’ button will be clicked in order to sign the user out of their account. This can be found by clicking on the person icon at the top right corner of the navigation bar.
      1. If the process of signing out of an account is successful, a success message will be displayed, otherwise, an unsuccessful message will be displayed
3. **Reset password (Priority - 3)**
   1. If the user forgets their password, they can go through the process of resetting their password
   2. In order to start the reset password process, the user has to enter the username to make sure the user’s account exists in the system
      1. If account exists, the user can be able to reset their password
      2. If the account doesn’t exist, an error message will state ‘This username does not exist’

# List of Non-Functional Specifications

**Performance:**

1. Compatibility - LAN party will be Primarily be made with Html, php, js, css, SQL it will perform equally on mobile and computers with screens up to 1920 x 1080p. Along with near minimal hardware requirements LAN can be easily run on any mobile device, laptop, desktop, etc.
2. Load Time - Expected load time for users will be a maximum of 1 second

**Reliability and availability**- LAN will be available for 99% of the year with the exception of 1% for maintenance and updates.

**Usability:**

LAN will be easily usable with minimal time learning time. Estimated learning time is less than 5 minutes.

**Expected Load:**

LANs expected load will be 50-100 people with high fluctuations as people will only be on for shorter amounts of time at once.

**Security Requirements:**

The Security of LAN will consist of mostly just Username/Password, but since it is a more personable account;2FA will be an option for the user. All usernames and passwords will be stored within an SQL server and if needed can recover a password by answering a personal question.

**Scalability:**

LAN is highly scalable as the only parts that will need to be enlarged are the server size and data storages.

**Capacity:**

Current system storages will be limited to what is available on lamp servers as that will be where LAN will be currently held.

**Regulatory:**

LAN will hold highly personal information and therefore if wanted by the user they can request that all personal information be deleted from the server and get a confirmation that it has been done.

**Maintainability:**

If there is a bug that appears and is able to be repeated multiple times by multiple users admins will start working on a solution immediately and the bug patch will be released in the next update, which will be tested, reviewed and retested, to confirm that the bug and any others were patched correctly and permanently; only then will the patch be released.

# High-Level System Architecture and Database Organization

## Programs and servers

1. **Lamp.cse.fau.edu Server** - for hosting the final project site for the duration of the semester where we will be housing the team progress as well as new changes made over the course of the milestones
2. **Discord** - This communication platform will be used as a communication and collaboration tool where virtual meetings, calls and files can be hosted. We will be tracking resources such as links, documents that are relevant to progress. Discussions through messaging will also be tracked using this platform
3. **Atlassian: Jira Software** - Jira will be used for project management, issue and task tracking, as well as ensuring deadlines and schedules are followed and met. This platform will also allow the product owner to prioritise product backlog items that need to be featured during that current spring
4. **Github** - Github will be used for code collaboration and as a code repository, where team members can commit their changes and view changes made by other team members. Github will also allow for specific version control in the event that the project code needs to be reverted to a previous state
5. **phpMyAdmin Database** - phpMyAdmin web database will be the primary database that will store information and data for the project. As this database is hosted online, developers will host the database on the backend of the site and allow the platform to store and manage user and group information. Developers will also have access to make changes and input data as needed.
6. **Postman** - Postman will be used to test new APIs and also perform backend testing on existing code. This tool will also be used to perform performance tests quantifying response time.
7. **Core APIs (tentative)**
   1. *Search API* - to be used to execute search queries based on string inputs then return hits that match that query
   2. *Steam Web API* - to be used to extract information from Steam Web platform
   3. *RESTful API* - to be used to request access, manipulate, and use data from the server-side

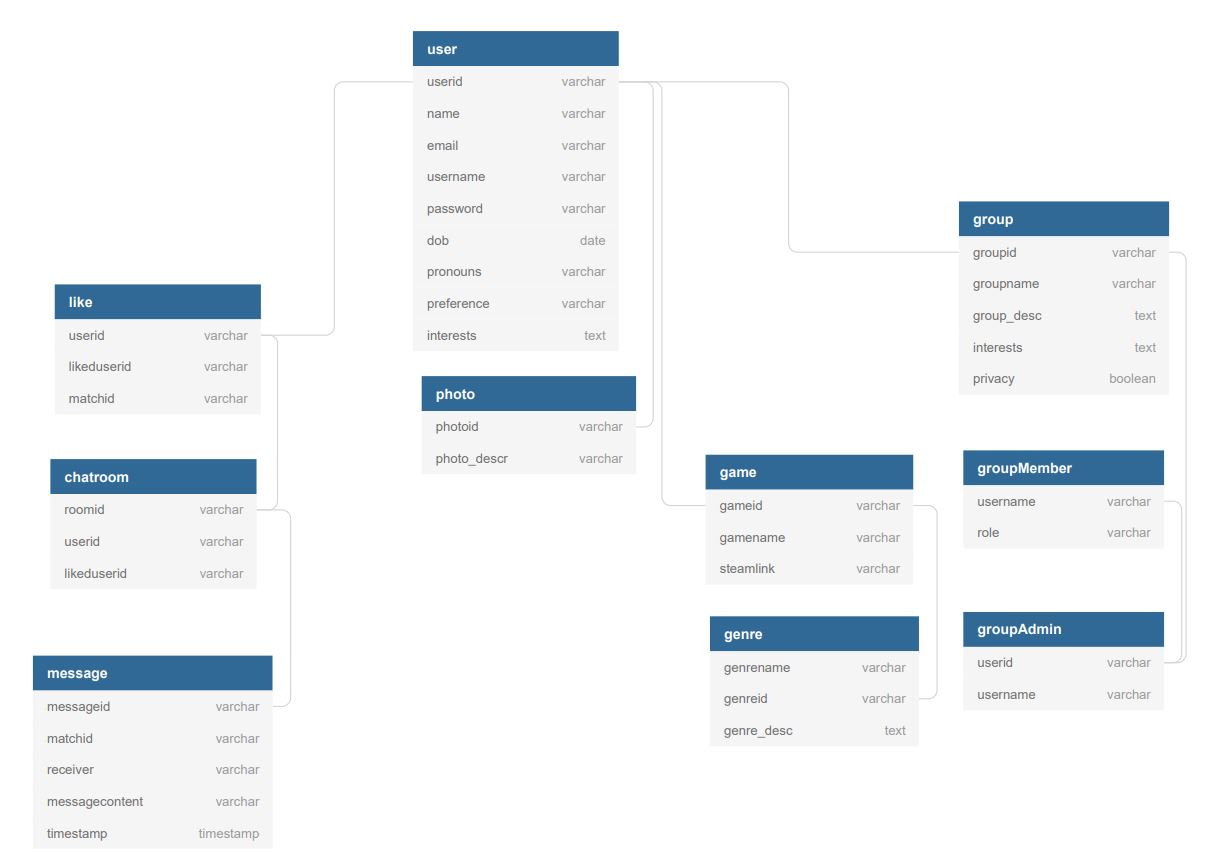
## Frameworks and Languages:

1. **Visual Studio Code -** Visual Studio Code is a code editor that developers will be using to make and commit changes to their remote branch. This tool will also be used to create new code using several languages:
   1. **Hypertext Preprocessor (PHP)** - to be used to create registration functionality that will connect to the backend where user and group information will be saved and stored in the site’s database
   2. **Javascript** - to be used to create user-facing functionality interactable to further increase user engagement and overall quality of experience
   3. **Hyper Text Markup Language (HTML)** - to be used to create structural framework for the website pages and allow for browsers to display information and the design
   4. **Cascading Style Sheets (CSS)** - to be used to create style-related customizations within each site page, further enhancing the overall look and feel of the content provided
   5. **React UI - JS Library** - to be used to create seamless, dynamic, and interactive user-interfaces, allowing for developers to reuse existing components to tie themes together.

## Supported Browsers:

1. The website platform will be developed to operate on at least two of the most popular internet browsers, including Google Chrome and Safari, but as Google Chrome is supported in both Windows and Mac systems, the priority will be aimed for full functionality for Chrome users.

Milestone 3 additions:



High-level system architecture and database organization visual

2) DB organization:

Table user:

userid - A unique identifier given to each user when they sign up

name - Birth or legal name of the user

email - The chosen email the user wanted to associate with the profile

username - The chosen name each user provides for their profile

password - The profile’s password for the sign-in process

dob - The user’s date of birth

pronouns - Any gender or role descriptors the user would like to be addressed by

preference - The user’s sexual preference or what they may be looking for on the site

interests - A text area where the user contributes any relevant information about their interests

Table photo:

photoid - A unique identifier for each photo stored on the database

photo\_descr - A text area where the user can provide a description of the photo

Table like:

userid - A unique identifier given to each user when they sign up

likeduserid - A unique identifier given to each ‘like’ interaction

matchid - A unique identifier given to each user to user match

Table chatroom:

roomid - A unique identifier given to each chatroom

userid - A unique identifier given to each user when they sign up

likeduserid - A unique identifier given to each ‘like’ interaction

Table message:

messageid - A unique identifier given to each message interaction

matchid - A unique identifier given to each user to user match

messagecontent - A text area where the user contributed any relevant messages for the other

timestamp - A date and time that will associate with the message

Table group:

groupid - A unique identifier given to each created group

groupname - An identifier/ title that distinguishes one group from another

group\_desc - A text area where the user can provide a description of the group itself

interests - A text area where members contribute any relevant information about their interests

privacy - A distinction on if a group allows other outside members to interact or not

Table groupMember:

username - The chosen name each user provides for their profile

role - Any title that each member holds example: Admin, leader, president, member, etc.

Table groupAdmin:

userid - A unique identifier given to each user when they sign up

username - The chosen name each user provides for their profile

Table game:

gameid - A unique identifier given to each game in the database

gamename - The given title of a video game

steamlink - A hyperlink that connects database with Steam game service

Table genre:

genrename - The actual name given to the genre

genreid - A unique identifier given to each genre

genre\_desc - A text area where the user or admin can provide a description of game genre

3) Media storage:

As of right now the only media that will be stored on the database itself are photos of the users and possibly photos of games that will appear in the search functions. The user photos will be displayed on the profiles itself and linked to the “user” database. The game photos will most likely be pulled from the gaming service called “Steam” and would be implemented from a link in the ‘Game’ database. The administrators of the “group” database may also be allowed to show photos there, but the area that the file will be stored is to be determined.

The accepted file types for the images will be: JPEG (or JPG) - Joint Photographic Experts Group , PNG - Portable Network Graphics, and possibly GIF - Graphics Interchange Format for the startup of the application. Any additional image formats may be added as we progress.

4)Search/filter architecture and implementation:

The main search algorithm will be linking users to other users, game pages, or groups of interest. The way the user could do this is through a search on the UI that will utilize the “user”, “game”, or “group” databases. Once inside the search the user may narrow the search down based on factors such as: game name, the Steam link, genre, group name, or general interests. Each one of these search terms has a unique id that the database can then associate the user profile with. When it concerns the Steam link: there will be an API that will share the link to the website and become available as a query. This hyperlink will be stored inside the “game” database.

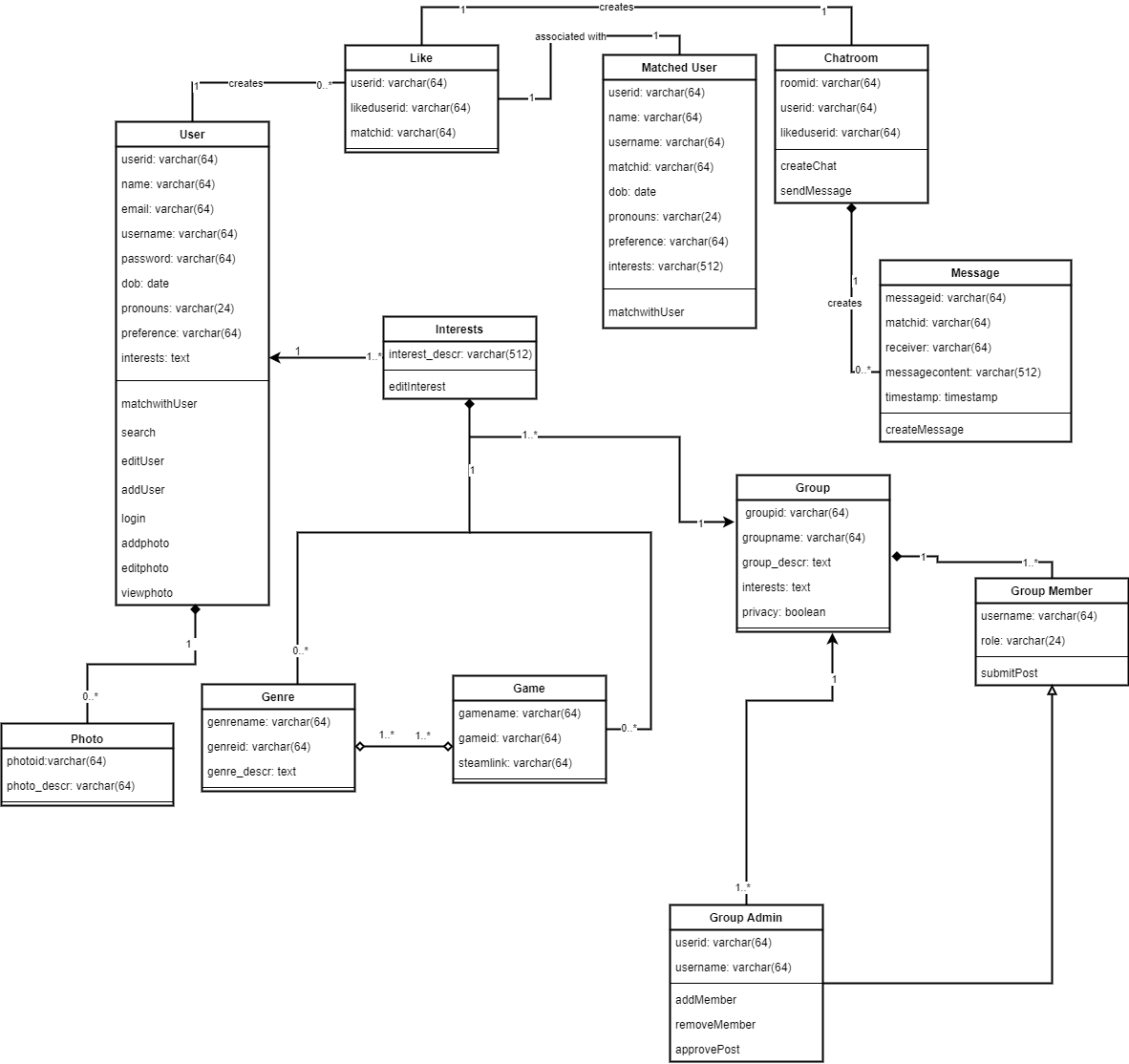
5) Our own APIs:

There are two main application programming interfaces (API) that are being considered at this stage in development. The first API being an authentication measure for when a user signs up for a profile. This API will essentially validate a user’s chosen email address for registration by using a token of some kind. This function will be reinforced by having an email sent out to the user that the user will click a link to and verify the email. This API will be created by Group 9 to add a common security measure. The second prospective API will be one that links LAN PARTY with the Steam gaming service and we may have to use a pre-existing API for this functionality. Steam provides a freely available API called ‘Steamworks’ that can be used for linking game descriptions and photos.

6) Any significant non-trivial algorithm or process:

One of the key features of LAN PARTY is that the application will have a matching system for those who want to use it as a dating or meet up website. A method that Group 9 will do this is through a recommender system that will appear in the UI. An algorithm needs to be created in order to match these users to each other based on the pairing of the user, like, group, game, and genre databases. From the match the users may choose to utilize the chatroom or messaging databases to communicate with each other. Another more optimistic function will be to have a user gain a notification when the group pages, new matches, or chat activity happens.

# High Level UML Diagrams

High-level Class Diagram

Identify actual key risks for your project at this time

Identify actual key risks for your project at this time

# Use-case Diagram

# 

# 

# 

# 

# 

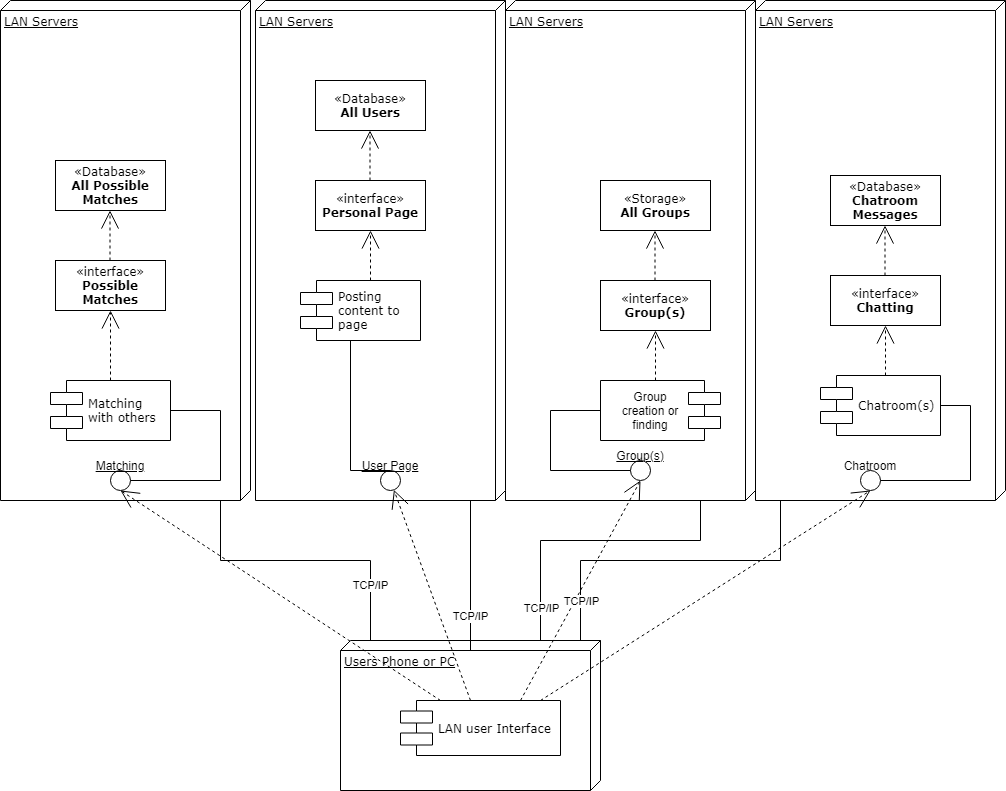
# 

# 

# 

# 

# Component and Deployment Diagram



# 

# 

# Identify key risks for your project at this time

1. For specific risks, this would also be learning new frameworks and working a lot with jQuery can be a challenge for the team members who may not have worked with it in the past. Many teammates are knowledgeable in HTML, C++, and Python to help assist any frontend and backend issues. Team integration and catching everyone up to speed with newly added code they had not directly worked on can also be a challenge.

2. Scheduling risks for our team would be gathering information about team progress, maintaining momentum during tight deadlines can create lag time while creating our website.

3. For technical risks, the most common issue to come upon when creating the website would be trying to solve each other’s code and fix up any issues someone else created. The reason why would be because since the communication is minimal, another user will try to edit the code that someone else created and think there is an issue when really there isn’t. This can cause complications for the code written already and mess up a lot of the data created. Another issue would be how each person on the team applies their coding knowledge. If someone isn’t too familiar with a certain language, this can cause complications with completion time for the website. For example, someone who is extremely comfortable with writing PHP code may need to catch another, less experienced, teammate on what the code means and how changes can be applied without disrupting the established functionality

4. Teamwork risks will imply lack of communication and assisting each other throughout the coding process. This can cause major slow down time for completion.

5. The only issue with copyright would be obtaining pictures and videos for our project. We would need to be able to get content without illegally applying it to our website. Another legal risk is utilizing open-source resources like libraries and frameworks, and style designs. We will need to ensure all of our code follows does not coincide with any copyrighted content.

# Competitive Analysis

Below is an analysis of ratings for a select few of the competitors from the CNET website article “[Best dating sites for 2021](https://www.cnet.com/tech/services-and-software/best-dating-sites/)” by Rebecca Fleenor. The table will display ratings of the websites based on some general aspects such as visual aesthetics and ease of use: homepage, design, and navigation.

An analysis will also be made based on sorting abilities in the website itself or ability to choose a mate: search, matching, communication, or Rate-a-Mate. The “Rate-a Mate” system would be to rate the experience with the other gamer. An example of this is a rating of the driver after a food delivery versus just ‘liking’ or making a profile a ‘favorite’.

Lastly the table visualizes how well a user with specific interests can find another user and if the user would have to pay to get a better match. The rating criteria will be a commendation system where (1 = poor; 2 = below average; 3 = average; 4 = above average; 5 = excellent). The ratings were concluded during a mock creation of an account for each site and attempting to search for other gamers to date.

**First impressions and ease of use:**

|  | OkCupid | Hinge | Match | Bumble |
| --- | --- | --- | --- | --- |
| Homepage | 2 | 3 | 4 | 4 |
| Design | 4 | 3 | 3 | N/A |
| Navigation | 5 | 3 | 5 | N/A |

**Sorting for specific type of mate:**

|  | OkCupid | Hinge | Match | Bumble |
| --- | --- | --- | --- | --- |
| Search | 4 | 1 | 4 | N/A |
| Matching | 4 | 2 | 4 | N/A |
| Communication | 3 | 1 | 4 | N/A |
| Game related | 3 | 1 | 5 | N/A |
| Rate-a-Mate | Not found | Not found | Not found | Not found |

**Money and metrics:**

|  | OkCupid | Hinge | Match | Bumble |
| --- | --- | --- | --- | --- |
| Pay to win | 3 | 3 | 4 | 5 |
| Average of table ratings | 3.5 | 2.1 | 4.1 | Inconclusive |
| Google Play Store app rating | 3.8 | 3.4 | 3.3 | 3.2 |

**OkCupid -** [**https://www.okcupid.com/**](https://www.okcupid.com/)

The homepage for OkCupid left more to desire. The page was a few rotating photos of couples in love and being playful with each other. The main page also has an area to sign in or make a profile. However, the homepage did not give any inkling as to what the layout, search, or matching process is. All of that type of information was blocked until the user made an account.

During the profile creation process the user would encounter questions to enter their geolocation, demographics and personal preferences. A few examples of the data collected were: gender; zip code; type of relationship desired; short or long term commitments; and age range for partners. In addition to those questions they had tried to personalize the experience. The way they did this was by making the user upload a photo and fill a textbox description about themselves. To further gain personalization they gave 15 prompted questions that span topics such as: personality traits, religion, life goals, astrology, employment status, and so on. After all those questions were complete: they required the user to validate through a phone number.

Once the account was created a main matched profile had been highlighted on the screen and multiple matches with percent profile match in bubbles at the stop of the screen. From here the user can like or dismiss them via a button at the top of the profile. Within the profile is displayed information about themselves, their likes and dislikes, and their demographics. There are filter options for any of these as well as other things such as smoking or non smoking, drug use, religious views, etc. For the purpose of this analysis a filter was placed in a “search” bar listed under a “Questions” tab. From here a general term of “gaming” can be filtered. Only 3 people had shown up using this filter which could possibly be due to the fact that the other user has to list it specifically in their profile in order for it to show up in the search.

The pay to win (gain access to more functionality) of the site was the middle of the road in terms of pricing for what you get. OkCupid has a Boost system: become a “top profile” for only 30 minutes. The user can get 1 boost for $8.99, 5 boosts $7.79 each, and 10 boosts for $6.49 each. The site also has more long term options below:

|  | Basic | Premium |
| --- | --- | --- |
| What you get: | No outside ads  Send unlimited Likes  Dealbreakers  Unlock all of your Intros | All of the features of OkCupid Basic, plus:  See everyone who ‘Likes’ you before you ‘Like’ them  See everyone's public answers to their questions before you answer |
| 6 months | $9.99/ mo | $24.99/ mo |
| 3 months | $14.99/ mo | $29.99/ mo |
| 1 month | $19.99/ mo | $34.99/ mo |

**Hinge -** [**https://hinge.co/**](https://hinge.co/)

One of the first things about Hinge to make note of is the fact that it is exclusively delivered in an app format. There is no website support for functional aspects of this service. The homepage is set up to be an information hub about Hinge, how to apply for jobs, and articles about the dating scene. Their main theme is to “delete this app” which implies the user would have found a match.

Hinge has an option to connect with a phone number or to Facebook which most likely would populate the geolocation and demographics and possibly other information. However the analysis was done manually. After all those questions were complete: they required the user to validate through a phone number. Much like OkCupid there were questions about gender, sexuality, religion, and so on. Where Hinge differs is what seems to be an Artificial Intelligence ‘like” system at play. The app will display a main match profile and the user will choose themes inside the photos that they like. The algorithm will run and then a new search would go through profiles on it’s own and give new matches based on that like. An example of this would be a “rose” in an uploaded photo from the user then a new “rose” search term was populated in filters. This was a unique way to sort for preferences, but it made searching for “gaming” specifically almost impossible at the start of an account creation.

The pay to win aspect for Hinge was fairly low. It would appear that the user could get by just fine without ever having to pay for premium services. They did have a Boost system as follows: 1 hour for each boost at a rate of 1 boost $9.99, 3 boosts $8.99, 5 boosts $7.99, super boost 24 hours for $19.99. The premium options are listed below.

|  | Premium |
| --- | --- |
| What you get: | Send unlimited likes  See everyone who likes you in one place  Set advanced preferences |
| 6 months | $4.99/ mo |
| 3 months | $6.99/mo |
| 1 month | $9.99 /mo |

**Match -** [**https://www.match.com/**](https://www.match.com/)

Match had one of the most straightforward homepages and it instantly got the user sorting and filtering data for searching for matches. The homepage asks for age ranges, zip codes, and who the user is interested in. The main drawback of the homepage would have to be that it does look a bit outdated in terms of styling versus the other websites listed in the analysis.

Like Hinge the user can connect their Facebook to populate data or use a phone number. When making an account it asks the user to provide their emails, birthday, name, and create a password. Additional questions were asked about ideal mate demographics and the type of relationship they are looking for. One thing that sets this website apart is that they make a point to not have an “it’s complicated” relationship status since they want serious members only. After all those questions were complete: they required the user to validate through a phone number.

Match was the best in terms of being able to narrow down the search for “gaming”. They gave a prompt to click all areas of interest and “gaming” was one of them. Narrowing down even further they ask open ended questions where a user could input “gaming” in as a hobby or interest.

Once the account was fully created the profile of other users populated the screen. The profiles looked great and the communication chat box was conveniently located directly on the match’s page. The search options inside the match dashboard were amazing. When searching for “gaming” as in interest it was able to show multiple pages of users that are also interested in games. By far this is the most competition for Group 9’s app.

The drawback for Match is that if a user chooses a pay to win option: it is one the more expensive sites out of every analysis. The price options are listed below:

|  | Standard | Premium |
| --- | --- | --- |
| What you get: | Sending messages  Seeing who viewed your profile | Unlimited messaging  Up to 12 Top Picks daily  Send unlimited likes  See everyone who likes you    See who's an upgraded  member  Read receipts  Monthly Boost  Weekly Super Like  Unlimited Rewinds  Discounts to Match Events |
| 12 months | $18.99/ mo | $19.99/ mo |
| 6 months | $22.99 /mo | $24.99/ mo |
| 3 months | $31.99 /mo | $34.99/ mo |

**Bumble -** [**https://bumble.com/**](https://bumble.com/)

The bumble homepage had a large photo with a ton of key words on it and what appeared to be samples of what a user can expect during the experience. One of the featured keyword searches on that image was of “gaming”. The main theme of this app is that the women take the initiative if and when a conversation is started.

The account set up had asked for demographic information such as name, birthday, gender in addition to adding at least one photo. The user is prompted to select which type of relationship they want to search for. This was a very unique feature since they have three different sections: dating, find a friend, and a “bizz” section for career meet-ups.

Bumble, unlike other dating sites, had given a huge roadblock in terms of access to the site. The photo that was uploaded in the profile creation was then to be verified with a “selfie”. This is where the website would show you a photo of a person doing some gesture that needed to be replicated in a “selfie” or a snapshot of the user. The website then asked for permission to access camera services on the user devices. A photo then would be taken and analyzed against the gesture in the sample as well as double referencing with the photo the user added to the profile earlier. If the photos do not match to their standards then the photo is marked as “moderated” and a flag comes up for not being within the guidelines of the site. One of the “features” of this site is “no catfishing” which means the person does not look like the person in the profile picture. This could be for many reasons including but not limited to: showing a skinny photo when the user actually has more weight, showing a photo of another person, or any other deviance from the appearance. Due to this particular roadblock: the analysis ended there.

The pay to win options were a bit unique here as they span for shorter and longer periods of time. They are by far the most expensive options on this analysis, but one of them spans for an entire lifetime. The price options are listed below:

|  | Boost | Premium |
| --- | --- | --- |
| What you get: | Backtracking  Unlimited 24-hour match extensions  Unlimited swipes  One Spotlight a week  Five Super Swipes per week | Access to your Beeline: This feature was originally part of the legacy Bumble Boost — it's the list of everyone who's swiped right on you that you haven't seen yet  Rematches with expired connections  Unlimited access to advanced filters  Travel Mode  Incognito Mode |
| Lifetime | N/A | $199.99 |
| 6 months | $47.99 | N/A |
| 3 months | $29.99 | $66.99 |
| 1 month | $14.99 | $32.99 |
| 1 week | $7.99 | $17.99 |

Advantages of LAN Party:

One of the main advantages that LAN Party has over all the other websites listed is the specificity of the site. Out of the websites analyzed (OkCupid, Hinge, Match, and Bumble) none had a game specific search option. If the site did have a search function the available term was generic. An example of a generic term that was integrated is “gaming” and “gaming PC building”. This leaves the user unclear on what type of game another user plays. According to [this Wikipedia page](https://en.wikipedia.org/wiki/List_of_video_game_genres) there are at least 11 confirmed main genres of video games with 69 sub-genres. The competition leaves it up to the user to engage with the other gamer with the potential for disappointment if they do not enjoy the same type of games. The website/app we propose to build would address this problem. The narrow search will let each user know exactly what game genre they are getting involved with.

For users searching for this experience it will be delivered with a playful page where they can let themselves or their game characters shine. This website/app will be more character and personality based versus the competition that focuses on physical appearances. Some users may feel comforted by this and feel as if they can finally be themselves. Gamers work hard on their characters appearances, armor, and stats so we want to make showcasing that a new normal.

Based on the progress of the project during the semester and if we have time to add it: another advantage would be a “Rate-a-Mate” system. This will be a way to attempt to keep every interaction a mature one and on track within the general conduct rules of the website/app. As seen in the analysis for competitors: rating a mate is not a feature they spend time on or care about. This will be just one of many ways that the LAN party can stand above the rest.

\*https://www.cnet.com/tech/services-and-software/best-dating-sites/

\*https://en.wikipedia.org/wiki/List\_of\_video\_game\_genres

# Team Roles

**Trang Leminh -** Team Lead, GitHub Master, Product Owner, Front & Backend Developer

**Bethlyn Joseph -** Scrum Master, Frontend Developer

**Tierney Mathis -** Backend Developer

**Raven Ruggs -** mySQL Database Admin, Frontend Developer

**Olivia Razzo -** Frontend Developer

**Kevin Young -** Backend Developer