Use case 1

Name: Wants to find other players

Primary actor: User

Preconditions: The user has a game he wants to play with others

Success guarantee: A gaming partner is found. Both or all parties have agreed on a time to meetup and what game to play.

Main success scenario:

1. The user logs into his account on the GMU app.
2. User chooses what game, at what time and with how many players he wants to party with.
3. User puts in requirement options for his future teammates, and posts it on the board for others to find.
4. Some other users with similar requirements find the post and put in a request to play with the user.
5. The user looks at their requests and finds them to be a good match and accepts.
6. An event is automatically generated(with info such as time of meetup, what game is to be played and such) where all parties(the gamers) are notified(via desktop or phone) when the event is about to start.
7. All users manage to join the online meet-up.
8. They play the game.

Extensions / alternate scenarios:

(4) No users find the post or want to take part in the online meet-up:

1. The event is canceled an hour before the scheduled meet-up time
2. User gets notified and is asked to create a new event or try again later

(7) A user wanting to join the meet-up has to duck out in the last hour before the meet-up:

1. User sends a notification to their newly formed party that they won't be able to make it
2. Remaining users are prompted with a poll asking if they want to find a last minute replacement or if they want to keep the party as it is

Misc / open issues:

- Should users be able to restrict visibility of their post regarding who can see them or join the meetup?
- What level of customization should be allowed for user profiles, event creation and notifications?

Use case 2

Name: Wants to join the application

Primary actor: Future user

Preconditions: Access to the internet and a browser via a phone or a laptop/PC

Success guarantee: Account is set up. Account is added to the database.

Main success scenario:

1. User enters in the URL of the app and enters the site
2. User navigates to the ‘Account creation’ part of the application
3. User enters in all required information for the account creation and manages to create an account
4. User now has the ability to log onto the site and start searching for gaming partners

Extensions / alternate scenarios:

(3) User doesn’t fill in all the required fields while creating the account:

1. A pop-up appears asking the user to fill in the required fields marked with a red star

2. User manages to fill in the rest of the required fields

3. User’s account is able to be created

Misc / open issues:

- What should be required to be filled in in order to create the account?
- What other optional fields should there be?
- Should the account log-in parameters accept the users nickname or just their email?

Use case 3

Name: Wants to report a problem with the app

Primary actor: User

Preconditions: The user has found a bug or some other issue while using the app

Success guarantee: The issue is successfully reported and acknowledged by the support team

Main success scenario:

1. User is logged into their account and using the app
2. User encounters a bug or some issue while using the app
3. User navigates to the Help and Support section of the app
4. User finds the “Report a problem” button and clicks it
5. User describes the problem they encountered in detail and submits the bug report
6. The support team receives the report and starts investigating the issue.
7. Once resolved, the user receives a message thanking them for helping with improving the app.

Extensions / alternate scenarios:

(5) User submits incomplete information.

1. The system prompts the user to provide additional details before accepting the submission
2. User updates the report and successfully submits it

Misc / open issues:

- How quickly should users expect a response from the support team?
- Should there be a way for users to track the progress of their reported issue?

Additional use cases in brief format

1. After completing a gaming session, the user is prompted to rate their gaming partners. The user submits a rating, which is then added to the partner’s profile, helping others assess their compatibility in future sessions.
2. A user accesses their friends list, adds new friends from recent gaming sessions, removes inactive contacts, and organizes their list to easily invite the right people to future gaming events.

3. A user accesses their profile settings to update personal information, select preferred games, adjust privacy settings, and upload a profile picture. This customized profile helps the user connect with others who share similar interests.
4. A user is very dissatisfied with one of his partner's toxic behavior and proceeds to report him to the app moderators. A ticket is attached to the toxic user and he is put on a list of potential undesireable gaming partners.
5. A user is temporarily banned from using the gaming meet-up app due to too many reports from his partners. Some of the reports include evidence of the users toxic behavior.
6. A user wants to get extra priority and extra features that improve his experience with the app. He starts a payment process using a secure payment method. He agrees to a monthly fee of X kr. and starts to reap the benefits of the premium upgrade to his account.

1.5 Vision statement (“What will the product accomplish for whom?”)

Summary of long term purpose and intent of the product
The intended long term purpose would be to serve the gaming community with ease to find friends to play with, a long lasting software that will help gamers unite, bond and most importantly have fun.

Reflection of the expectations of all stakeholders
As a stakeholder they would like to have the software to be easily managed, reliable with few errors, also they would like to make money from it so we would need a subscription service. The stakeholders would also need a good marketing plan and like the product so they become invested and invest.

Idealistic - but grounded in realities of markets, technologies, architecture, resources etc.

Template:

- For gamers
- who want to make friends and play online
- the SkillSync
- is a gamer software
- that takes in the game you want to play and the respective rank and searches compatible users that would fit in a squad or a duo with
- Unlike GameTree
- Our product will be extremely simple and easy to use.

1.4 Success metrics (“How can we tell whether we are successful?”)

Measurable, quantifiable criteria that indicate whether the project was worth undertaking

There will be a very clear indication when we are successful because we are going to have a great vision about our income with how many subscribers we are going to have, also there will be a plausible indication of success if we have streamers and famous influencers using this program.

2.2 Scope of initial release (“What should be rolled out first?”)

Beside features, scope can also comprise quality attributes (e.g. performance)
Focus on features that will provide the most value at the most acceptable cost to the broadest community in the earliest time frame.

What will be released first would be the advertisements for the program, there would be streamers and other gamers/influencers that would promote the program and that they were intending to use it when it launches, then there would be a trailer promoting the program and making it look good and getting people hyped up.

Project Plan

- List out what use cases (and end-points) will be completed in each of the upcoming phases
    - Use the techniques you learned in HBV401G for Effort Estimation and Prioritization
    - Consider which end-points (see following slides) you need to implement your use cases
    - Schedule a set of end-points you will complete in each sprint, considering the guidelines set in assignments 2-5 (see following slides).

Project Plan for SkillSync Application

Use Cases

1. **User Registration & Profile Setup:**
    - Users create accounts, set up profiles, and input gaming preferences.
2. **Game & Rank Selection:**
    - Users select their favorite games and input their rank levels.
3. **Matchmaking:**
    - Match users with others based on game and rank level.
4. **Real-time Notifications:**
    - Notify users when a match is found or if a user of similar rank is available.
5. **Friend Requests and Chat:**
    - Users can send friend requests and chat with their matched players.

Effort Estimation and Prioritization

Effort Estimation:

- Use function point estimation and story points to determine the effort.
    - Low complexity: User Registration, Profile Setup (3-5 Story Points)
    - Medium complexity: Game & Rank Selection, Matchmaking (5-8 Story Points)
    - High complexity: Real-time Notifications, Friend Requests and Chat (8-13 Story Points)

Prioritization:

- **Phase 1 (Essential Features)**: User Registration & Profile Setup, Game & Rank Selection, and Basic Matchmaking.
- **Phase 2 (Nice-to-have Features)**: Real-time Notifications, Friend Requests, and Chat.

Phases and Endpoints

Phase 1 (Sprint 1-3): Core Functionality & Setup

1. **Sprint 1:**
    - **Use Cases**:
        - User Registration & Profile Setup
    - **Endpoints**:
        - POST /register (user registration)
        - GET /profile (retrieve user profile)
        - PUT /profile/update (update profile details)
    - **Tasks**:
        - Create user table in the database
        - Implement basic front-end for user registration
2. **Sprint 2:**
    - **Use Cases**:
        - Game & Rank Selection
    - **Endpoints**:
        - GET /games (list available games)
        - POST /rank (set rank for specific game)
        - GET /rank (retrieve user's rank)
    - **Tasks**:
        - Expand user schema with game and rank info
        - Allow users to select and update their ranks
3. **Sprint 3:**
    - **Use Cases**:
        - Basic Matchmaking
    - **Endpoints**:
        - GET /match (find users with same rank in selected game)
    - **Tasks**:
        - Implement matchmaking logic in the back-end
        - Return matching users based on the rank and game

Phase 2 (Sprint 4-6): Advanced Features

4. **Sprint 4:**
    - **Use Cases**:
        - Real-time Notifications
    - **Endpoints**:
        - POST /notifications (send notification)
        - GET /notifications (retrieve notifications)
    - **Tasks**:
        - Implement push notifications when matches are found

5. **Sprint 5:**
    - **Use Cases**:
        - Friend Requests
    - **Endpoints**:
        - POST /friends (send friend request)
        - GET /friends (list friends)
    - **Tasks**:
        - Implement friend request system, storing relationships in the database
6. **Sprint 6:**
    - **Use Cases**:
        - Chat System
    - **Endpoints**:
        - POST /chat (send message)
        - GET /chat (retrieve messages)
    - **Tasks**:
        - Implement basic chat functionality with a database to store messages

ChatGPT var notað til að fá hugmynd af uppsetningu að project plan.

Óskar Samúel Ingvarsson
Aron Daði Gautason
Daníel Theodórs Ólafsson
Kristinn Roach