NORTHERN ARIZONA UNIVERSITY
College of Engineering, Forestry & Natural Sciences

Collegiate Esports Platform

https://github.com/deltarod/CS386-Group-20

Members:
Blake Lawton
Tristan Miller
Ruben Rincon
Jennie Rychman

Course:
CS 386
Marco Gerosa
Spring 2018

D.2 Requirements

1. Positioning

1.1 Problem Statement

The problem of low exposure and weak communication for the collegiate esports scene affects college and incoming college students interested in colleges esports scene; the impact of which is less connection for collegiate esports fans and exposure of players.

1.2 Product Position Statement

For college and incoming college students interested in the collegiate esports scene, the Collegiate Esports Platform is a website/desktop app that connects collegiate esports fans and esports in a organized and easy to use fashion; unlike any other product ever created, our product will be the first to link this connection in an extremely proficient and organized way.

1.3 Value Proposition

Collegiate Esport Finder is a website that allows college students to find other video game players and create teams for their college, allowing them to find players of similar games, and build an online community around a college. Consumer segment: College students that like to play video games have a hard time finding users.

2. Stakeholders

Users: College students interested in Esports, or video games in general
Clients: Colleges or companies looking to sponsor,
Competitors: Other friend finding games, facebook groups and discord chats
Detractors: Someone who doesn't like the layout of the website, or someone who dislikes that it is centered around college players
Developers: Maybe

3. Functional Requirements

1. A chat lobby - To communicate with other players or managers.
2. Player profiles - To show stats per game, whether the player is casual or competitive, and give managers a baseline of a player's skill.
3. A search engine to browse players, managers, local events, etc. - To provide basic search capabilities for the site/app.

4. Non-functional Requirements

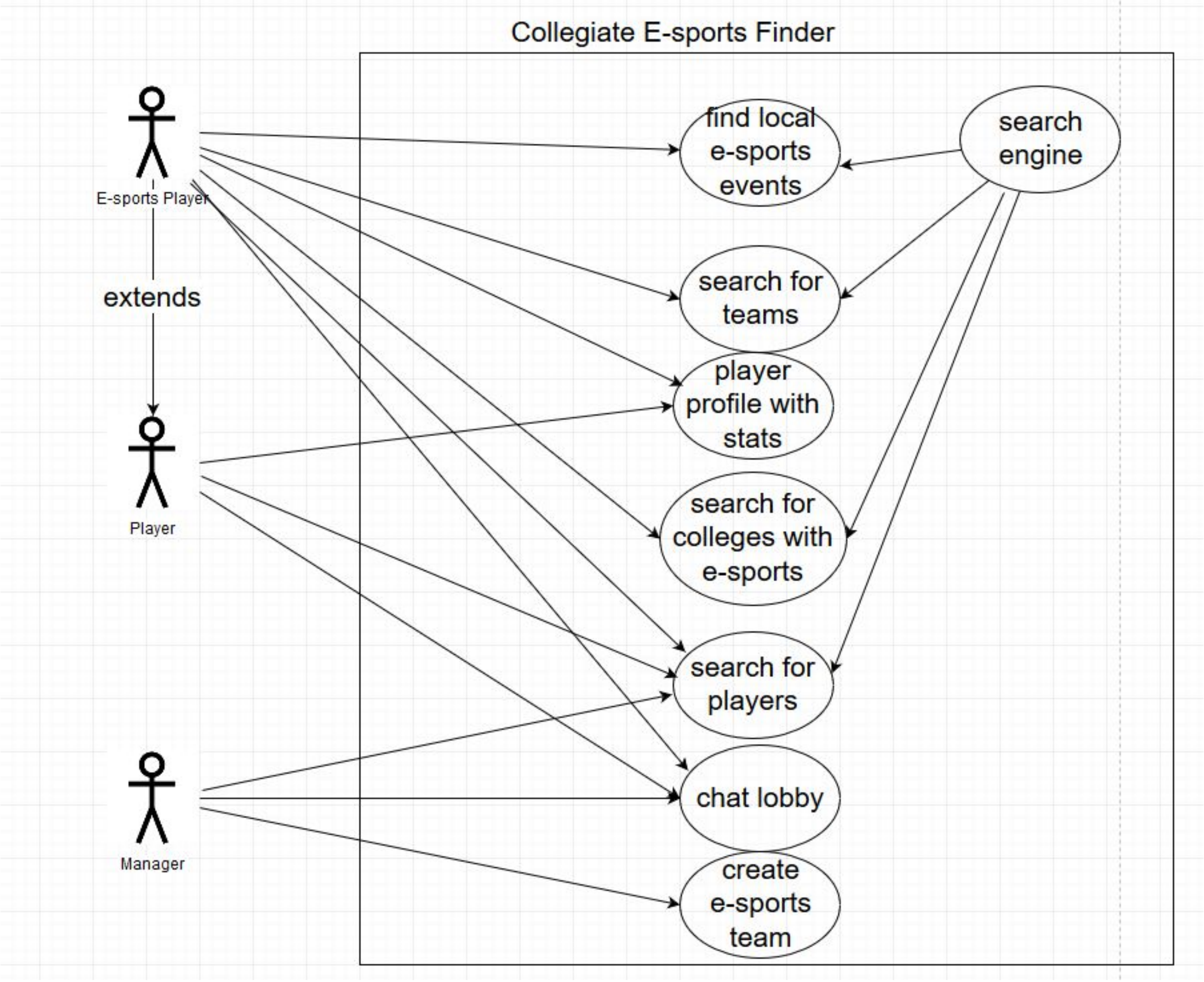
1. Portability - Our product could be accessible through a website, a mobile app, or desktop app.
2. Co-existence - Player and manager data needs to seamlessly integrate between website and apps.
3. Maintainability - Site and apps should be lightweight enough to allow easy maintenance between different browsers, apps, etc.
4. Learnability - Players, managers, teams, etc should find it simple to find local events, players, and stats after a short period of time.

5. MVP

For our minimum viable product, we will need to at least implement a way for players and managers to communicate, a way to join local e-sports teams, and a way to find local e-sports events. With these bare minimum features, our product will serve as a basic product for those interested in their local college/area's e-sports scene. It would be best to first start with a simple website so that we can implement these features simply, quickly, and without having to worry about porting to apps before we're done testing.

6. Use Cases

6.1 Use Case Diagram



6.2 Use case descriptions

Use-Case 1: Find a local e-sports event

1. Brief Description: A player is attempting to find a local e-sports event near their area

2. Actor Brief Descriptions:

2.1 Player

3. Preconditions:

The player has an active internet connection to the database

4. Basic Flow of Events:

1. The user case begins when the player accesses the search bar
2. The player specifies filters and keywords (game, max distance, date, etc)
3. The player begins the search
4. The search engine filters results based on keywords
5. The search engine displays a list of local events that satisfied the criteria

5. Alternative Flows:

5.1 No results:

In step 4 of the basic flow, if there are no results that fit the player's search criteria then the use case ends with a failure condition

6. Subflows: None

7. Key Scenarios:

7.1 No response from database

8. Post-conditions:

8.1 Successful Completion:

A list of all nearby events is displayed to the player.

8.2 Failure Condition:

Page announces then displays lack of results.

9. Special Requirements: None

Use-Case 2: Find players for similar games

1. Brief Description: A player is attempting to find players for games that they play

2. Actor Brief Descriptions:

2.1 Player

3. Preconditions:

The user has an active internet connection to the database

4. Basic Flow of Events:

1. The use case begins when the user accesses the search bar
2. The player specifies filters and keywords (game, ranks, console)
3. The player begins the search
4. The search engine filters results based on keywords
5. The search engine displays a list of players that satisfy the criteria

5. Alternative Flows:

5.1 No results:

In step 4 of the basic flow, if there are no results that fit the player's search criteria then the use case ends with a failure condition

6. Subflows: None

7. Key Scenarios:

7.1 No response from database

8. Post-conditions:

8.1 Successful Completion:

A list of all players that attend their college and play their game

8.2 Failure Condition:

Page then displays lack of results.

9. Special Requirements: None

Use-Case 3: Search for an e-sports player to recruit

1. Brief Description: A manager searches for potential players to recruit

2. Actor Brief Descriptions:

2.1 Manager

3. Preconditions:

The manager is connected to the database

4. Basic Flow of Events:

1. The use case begins when the manager accesses the search bar
2. The manager specifies all e-sports players with optional filter keywords
3. The manager begins the search
4. The search engine filters out all non-e-sports players
5. The search engine filters out the remaining players based on keywords
6. The search engine displays a list of all e-sports players that satisfied the criteria

5. Alternative Flows:

5.1 No results:

In step 4 or 5 of the basic flow, if there are no results that fit the manager's search criteria then the use case ends with a failure condition

6. Subflows: None

7. Key Scenarios:

7.1 No response from database

8. Post-conditions:

8.1 Successful Completion:

A list of all e-sports players is displayed to the manager.

8.2 Failure Condition:

The page displays the lack of results.

9. Special Requirements: None

7. User Stories

1. As a manager, I want to be able to see a list of all players looking for an e-sports team so that I can find new potential players to recruit.
Priority: High
Estimated time: 5 hours

2. As a player, I want a way to display my stats and highest ranks for several games so that managers only need to go to one place to gauge my potential.
Priority: Medium
Estimated time: 4 hours

3. As a player, I want a way to communicate with other players or managers so that I can find players to play with or e-sports teams to try out for.
Priority: High
Estimated time: 3 hours

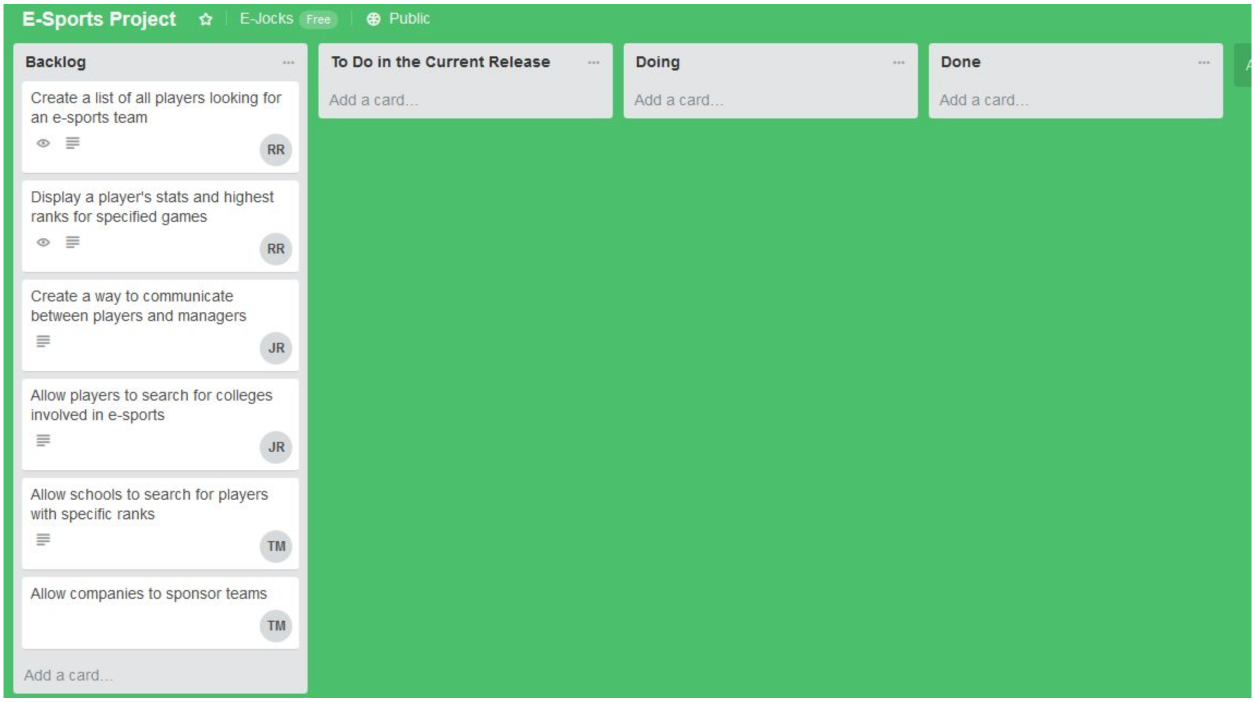
4. As a player, I want a way to search for colleges that are involved in e-sports so that I can have an idea of what colleges to apply for that might support an e-sports career.
Priority: Medium
Estimated time: 3 hours

5. As a school, I want a way to search for players with a specific rank in a game such that we can recruit them for that school.
Priority: Medium
Estimated time: 2 hours

6. As a company, I want to sponsor colliegiate teams with our logos, gear, or money, for that team
Priority: Low
Estimated time: 2 hours

8. Trello

Trello link: https://trello.com/b/NPTDEIUD/e-sports-project



Backlog

- Create a list of all players looking for an e-sports team — RR
- Display a player's stats and highest ranks for specified games — RR
- Create a way to communicate between players and managers — JR
- Allow players to search for colleges involved in e-sports — JR
- Allow schools to search for players with specific ranks — TM
- Allow companies to sponsor teams — TM

To Do in the Current Release

Doing

Done

Group Participation

Blake Lawton: Completed 1.1 and 1.2
Tristan Miller: Completed 1.3 and 2, use case description 2, user stories 5 and 6
Ruben Rincon: Completed 5, 6.1, use case description 1, user stories 1 and 2
Jennie Ryckman: Completed 3, 4, use case description 3, user stories 3 and 4