Sprint 1 Review -   
Recording - <https://kennesawedu.sharepoint.com/sites/Team-SWE6813-WEB-SERVICE-TEAM2/Shared%20Documents/General/Recordings/Sprint%20Review%20and%20Retrospective-20220316_205939-Meeting%20Recording.mp4?web=1>

In this sprint we projected completion of the following User Stories at the start of the sprint:

* Story 3, Market Research - 1 task
* Story 4, Become Familiar with GIT - 3 tasks
* Story 12 Revise Project and Research Proposal to get approval - 5 tasks
* Story 15 Collect and Organize User Stories - 3 tasks
* Story 23 Design UI - Partial - 2 of 3 tasks completed - Split Task
* Story 38 Display Full Profile - Unaddressed due to API Contract Issues
* Story 42 View Stack of Summary Profiles - Unaddressed due to API Contract Issues
* Story 57 Set up Azure DevOps for Sprint 1 - 2 tasks
* Story 60 Connect API - 1 task
* Story 61 Display Summary Profiles - Unaddressed due to API Contract Issues

As a team, we propose the following general solutions in Sprint 2, The actual Sprint 2 Goal and User Stories will have to be planned and presented to the team in the Friday Meeting.

* Data Connection and Table Structure with Test data, Google Firebase API
* User may select matches
* User Accounts
* PSN Sign-in API integration - We will use PSN authentication for our site login for this sprint. (A user MUST have a PSN account to log into out site.)
* We will research Chat in preparation for Sprint 3
* Basic matching algorithm
* GUI Interfaces
  + User Matching
  + Match Card
  + Match Full Profile
  + User Profile Setup
  + Splash Screen for User Login / User Account Creation
  + List of User matches selected
* At least some level of testing

Specifically, we will not address this sprint:

* Accepted Matches between gamers (friends)
* Chat or other User Interaction Implementation
* Social Media Integration
* Complex Matching Algorithm

Retrospective:

Knowledge Gaps we anticipate addressing:

* Javascript
* Firebase Integration
* PSN API Integration

Process Changes to improve operations:

* Check Sprint 2 Deliverables and Rubric before sprint planning and integrate into the sprint
* Planning can be improved by doing the following;
  + Each selected user story should be planned out with subtasks, acceptance criteria, metrics. Not leaf level activity should take more than about 4 hours so that each team member should anticipate completing at least one work item each standup.
  + All items should mark dependencies and should be placed in the sprint in order.
  + We have to plan at least some unit testing to occur after every merge. We should have a plan to accomplish that.
  + Each User Story will have clear acceptance criteria and closure will require review with at least one other team member.
  + There will be a Sprint Goal that clearly states what we expect at the end of the sprint. It will be published in the repository.
* Developers are encouraged to add tasks when a new logical division of work is identified, but before work begins on that task. Developers or Leadership may not add User Stories after the start of the sprint.
* In the event that a User Story will not be completed within the sprint, the story will be split according to the guidelines included in the readme file. All attempts should be made not to split a User Story after the start of the sprint.
* Attendance at Standup Meetings
  + The three attending agreed to almost any schedule of 2 weekday standup meetings as long as they are kept to 15 minutes only. I assured the team that they would be limited to 15 minutes. But this leaves Steven and Jerry as needed to make a firm commitment to a time that will work for both of them.
  + Attendance at the standup meetings in mandatory to everyone except the Product Owner, who is encouraged to attend.
  + As decided in Sprint 1, a production day will be considered to be 1/3 of a week since this is a student exercise. We will have 3 standup meetings each week, two on weekdays and one Sunday at 6:30 (EST) as previously established. Each person is expected to deliver 3 pieces of information, deliverables completed, expected before next standup, and impediments.
* The team will continue to operate on the basis that user stories and tasks are self-assigned by looking at the boards and picking items that you can produce. Per the Scrum Guide, we do not plan to assign tasks. This is a basic necessity of leadership as a service.
* Git - At this point, since we have not deployed the site, the main line will also be the development line. Developers will branch the main line at the beginning of a production day and once their work is confirmed will commit and submit a pull request at the end of the day. Branch Names will reflect what they plan to produce during that day. All pull requests for the day will be reviewed, approved and merged at the end of each production day (time between standup meetings.)