SaaS Application Project: Tournament Organizer

Team Hairless Cats

Andrew Cheah, Bryce Wilson, Leah Jo, Chunga Lee,
Munkhtur Myagmar, Joshua Tan

Project Plan

March 30th, 2022, Version 1.0

Introduction

Staff members' involvement into building a collaborative work environment is essential to keep a successful workflow and results. The development of this SaaS project is to allow a better administration of the tournament play between staff members. Since the software offers many functionalities to support staff members' involvement in the events, the solution will encourage a more collaborative staff community.

The solution includes the staff members to self-register and share their open hours to attend tournaments. Also, since the software is a web-based mobile-friendly application, the employees' will be able to choose from a list of tournaments based on their availability. For administrators, it will provide the system to offer an administrative module to organize each tournament. Admins will be able to describe the details of the tournament and include an expected duration of each tournament to further help the employees to manage their availability.

This project plan will further elaborate the tasks that our team needs to complete to provide the application. The critical path includes key tasks which are important parts our team needs to focus on and the path we need to follow. These are tasks we must absolutely complete in order to ensure fluent timely project delivery.

Team Roles and Responsibilities

All tasks from all phases mentioned in this project plan at this time are assigned to the group at whole. We plan to continuously update these responsibilities with more concrete decisions of individuals or entities that will take up the tasks, closer to their actual planned start date. Relying on the 'living' nature' of this document.

Critical Path Plan: (Non-critical paths Items omitted)

- **Project Conception and Initiation Phase**
  - Terms Of Reference 1/11/22 - 1/17/22
  - Project Plan 1/17/22 - 1/26/22
  - Collect User Stories From Stakeholders 1/26/22 - 1/26/22

These steps are important to understand what the project is and what tasks the project should complete. It helps to manage time and show how the project will be processed to the stakeholders. Therefore, this project plan is focused to be duration based for better time management. By collecting user stories from the stakeholders, this project can improve the quality of the project by reflecting the preferences of stakeholders.

- **Project Requirements Phase**
  - Making Additional User Stories 1/26/22 - 1/30/22
  - Identify Non-functional Requirements 1/30/22 - 2/2/22
  - Identify Functional Requirements 1/30/22 - 2/2/22

We believe that identifying the functional and non-functional requirements are critical in order to determine the scope of the project. Gathering the requirements will also allow us to better understand and break down the necessary tasks, so that we can allocate resources efficiently. Performing a thorough requirements gathering and the creation of a broad range of user stories will help to make sure that the customer's needs are fully understood and implemented.

- **Project Design**
  - Design data (build data ER diagrams) 2/3/22 - 2/4/22
  - Implementation Design of Functional Requirements 2/3/22 - 2/5/22
  - Algorithm for scheduling events 2/4/22 - 2/7/22
  - Algorithm for matching competitors 2/4/22 - 2/7/22
  - Specification and Planning of Database Schemas 2/4/22 - 2/7/22
  - Design Java Backend Class Diagrams 2/7/22 - 2/9/22
  - Design Wireframes 2/6/22 - 2/9/22

While the exact details of the design will change somewhat while we work on it, it is important to create a high-level design up front so that the different components can be worked on separately. Our goal is to create a modular design by utilizing OOP principles. This allows each module to be worked on independently which is helpful with the limited amount of time that we have to meet each week. It also provides for the business logic to be easily changed in the future as needs arise (differing tournament types, scheduling systems, information sources, etc).

We start by working on the data definitions and mapping of functional requirements. These are the most critical steps because they influence how all of the other components are interconnected. We can then work on designing the algorithms, database schema, and backend modules in parallel since the common infrastructure has been agreed upon. Once all of the components have been designed, we can create a wireframe to link them together.

- **Project Implementation**
  - AWS & public API setup 2/15/22 - 2/16/22
  - Coding for backend system by java 2/16/22 - 2/26/22
  - Setup database 2/16/22 - 2/17/22
  - Build UI 2/26/22 - 3/19/22
  - Build Staff Member Role 2/22/22 - 2/23/22
  - Build Admin Role and functionalities 2/24/22 - 2/25/22
  - Build CRUD functionality for tournaments 2/16/22 - 3/6/22
  - Build CRUD functionality for individual matches 2/16/22 - 3/6/22
  - Build tournament registration function 2/16/22 - 3/6/22
  - Build various tournament types 2/16/22 - 3/6/22
  - Build Competitor Matching function 2/16/22 - 3/6/22

In this phase, we will focus on building the technical features and functionalities of the project. We will start by setting up the necessary backend and frontend systems. Then we will implement the various features of the project in an incremental fashion. This document lists out some of the various core features that we can expect to be built for this project. Additional features will be added when we have further identified our user stories and various requirements.

We will be having both a frontend and backend team so as to be able to work on different aspects of the application concurrently. During this time, we will hold weekly meetings to keep each other updated about our progress.

- **Project Testing Phase**
  - Unit Tests 2/16/22 - 4/1/22
  - Integration Tests 2/23/22 - 4/1/22
  - Acceptance Tests 4/1/22 - 4/3/22

Testing the implementations of the previous step is critical. Using unit tests, it can check whether each function we implement is working correctly according to our expectations and specs to fix bugs and problems. Therefore, unlike other test types, the unit tests will be carried on continuously throughout the course of this project. This ensures each feature to execute smoothly before any integration. By using integration tests with unit tests, it can check if the larger modules are working and the smaller units are interacting in the right ways. Finally, the project will check whether the overall application works expectedly to check if all the requirements and parts of the contract are met. Passing the acceptance criteria and satisfying the user needs is important before demonstrating the project and its final result to the clients.

- **Project Performance/Monitoring Phase**
  - Demo preparation 4/1/22 - 4/4/22

Demo preparation is important because it is the step preceding the client meeting and showing the final result of the project. It is to show all the benefits of the projects and how the application works efficiently in the actual demo. This would be the last step before the release of our product and the final step to check upon any errors.

- **Project Maintenance Phase**
  - Actual Project Demo 4/4/22 - 4/4/22

This final step allows us to demonstrate all our team work and the project. The client will be able to review if their requirements are met and how to navigate within the application in order to check completed user stories.

Non-Critical Path Plan: (Critical path items omitted)

- **Projection Implementation**
  - Repo setup (Project structure, test file) 1/17/22 - 1/30/22

This is the first step we need to do to start the implementation, basic repo setup, not anticipated to be a substantial task.

- **Project Testing**
  - Testing Planning 2/10/22 - 2/14/22
  - Smoke Tests 2/23/22 - 4/1/22
  - Regression Tests 3/6/22 - 4/1/22
  - UI Tests 3/19/22 - 4/1/22

Non-critical path tests include the initial testing planning meeting as well as smoke tests and regression tests as by nature these tests will only occur if bugs are found

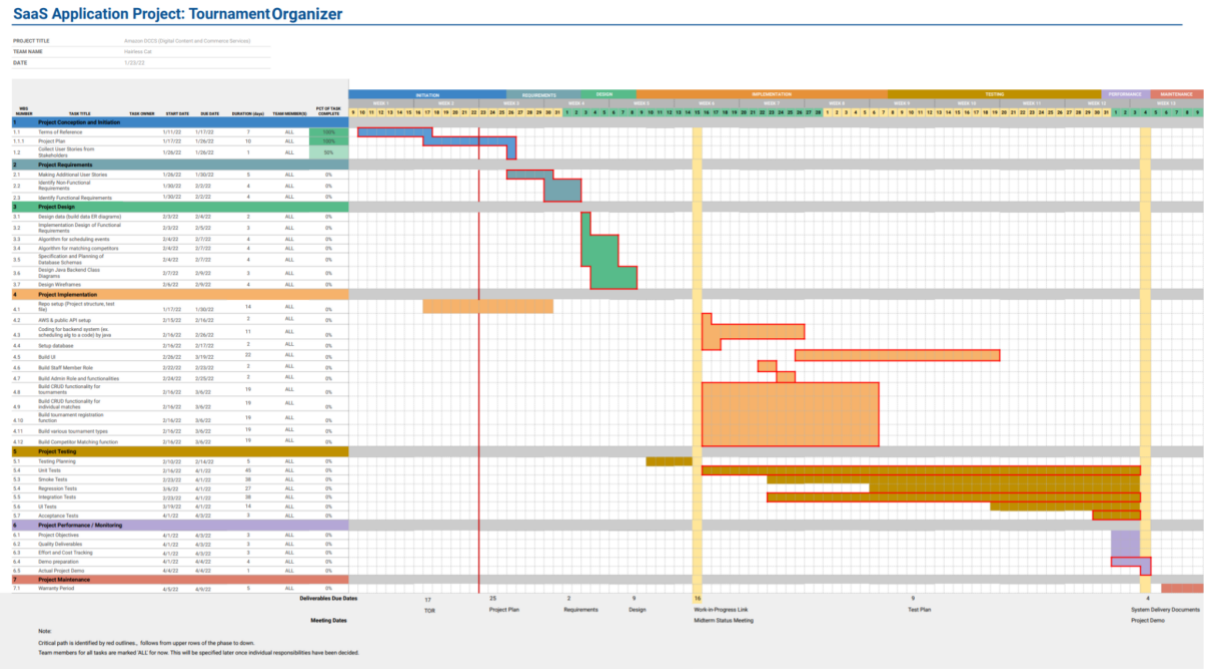
- **Project Performance / Monitoring**
  - Project Objectives 4/1/22 - 4/3/22
  - Quality Deliverables 4/1/22 - 4/3/22
  - Effort and Cost Tracking 4/1/22 - 4/3/22

Tracking efforts to ensure proper function and to prevent issues before release period, should be minimal.

- **Project Maintenance**
  - Warranty Period 4/5/22 - 4/9/22

During this phase, we will continue to monitor the status and performance of the application. If any bugs are found, we will work on fixing them. Other improvements to the application might also be added, but we expect them to be minor updates and quick to implement.

Gantt Chart



This Gantt chart provides a comprehensive visual display of our project plan. An enlarged version can be viewed at: https://docs.google.com/spreadsheets/d/1FatZKU-vUmtgv4O1bbCNOVCM-OOuv79ioHBB78w1WRU