

## Requirement Elicitation - Group 17

Taiga links:

- <https://tree.taiga.io/project/nimilshah55-ser515-group17/backlog>
- <https://tree.taiga.io/project/nimilshah55-ser515-group17/taskboard/sprint-1-19368>

Github link:

- <https://github.com/darshinshah/SER-515-Group-17>

Feasibility Study:

1. The objective of the organization is to develop a Soccer Tournament Website for the management of at least 350-750 teams. This website can schedule games between teams, manage referees, coaches, players along with payrolls, and add field maps for different teams. This becomes a sole solution for managing tournaments – and the generic nature will enable it to be used by any and all the tournament stakeholders.
2. The system will contribute to all the personnel involved in organizing a tournament and make their task seamless. They can be on top of all the transactions/scheduling with different available features (timer-based triggers).
3. The said system should be built within the time frame with limited features providing all the basic functionality. Current technology provides us many secure web frameworks so that we can get started with coding the logic and not spend time setting up the base network requirements and the HTTP/TCP protocols. We will incorporate open-source libraries which should not incur any monetary costs.
4. We have 6 sprints to work on the project with 5 people in our team. We plan to spend ~15hours per week per person = 75 hours per week = 900-man hours in total. 900 hours should be enough to match the said requirements for the Soccer Tournament Website.
5. We will be using the latest technology frameworks that are open source and should be easy to integrate with or compatible with different upcoming technologies. As for the legacy system integration, we could have APIs written to do the task.

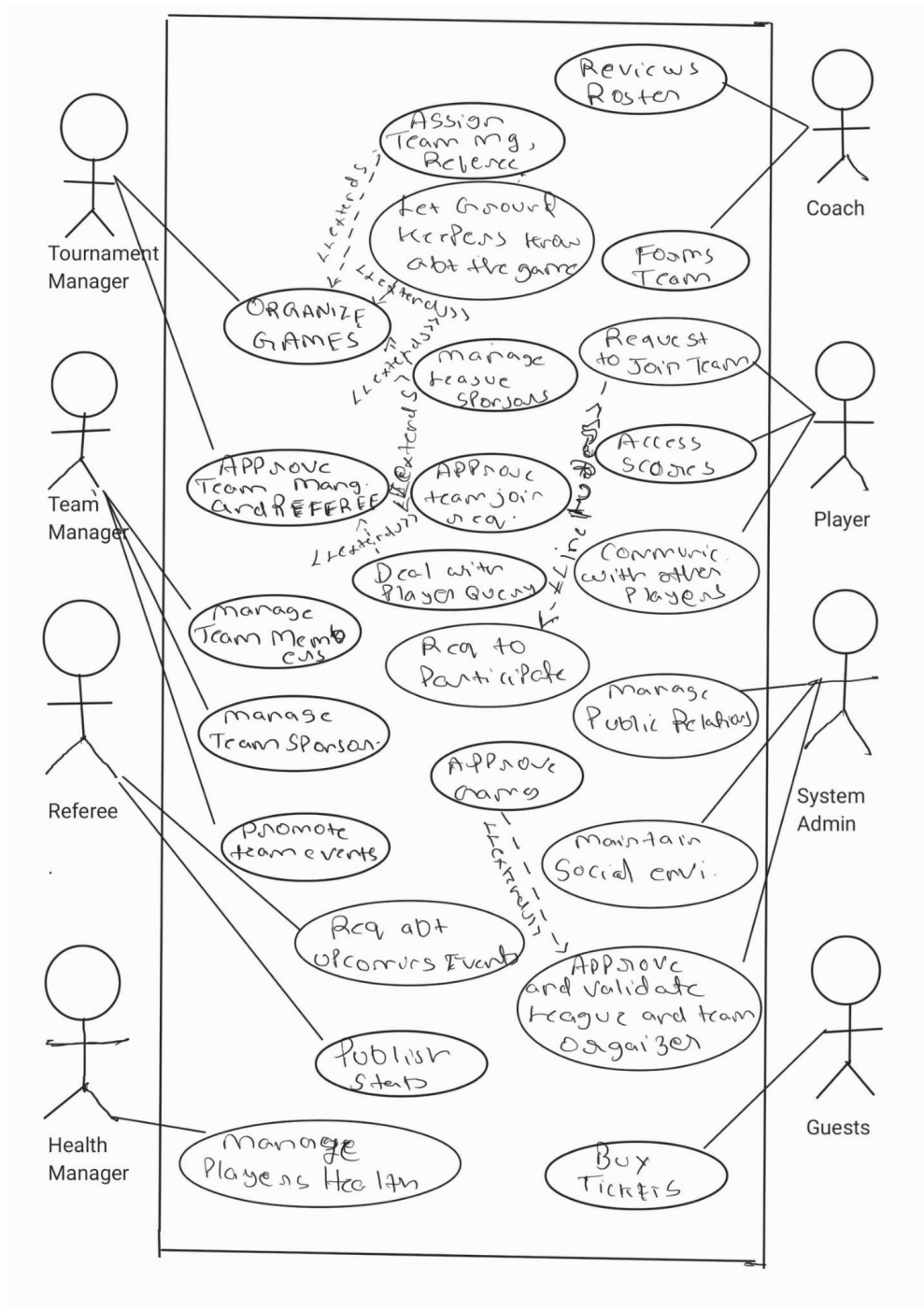
Requirements Engineering:

1. Requirements Gathering - To figure out the requirements, we have done a thorough study of the project document attached on the canvas. Few lectures that happened during the class time were dedicated to a detailed discussion on project requirements and were very helpful for us to understand and find out the requirements. We conducted standups to go through the requirements, brainstorming sessions to understand requirements, and to understand the technical skills of every individual to find out the technology we will be using to develop the website. During the brainstorming session, we produced a document that lists out roles and responsibilities the website needs to support and all the features that will be available on the website. The same is uploaded on canvas.
2. Project Vision: Develop a Soccer Tournament Website for management of at least 350-750 teams. This website can schedule games between teams, manage referees, coaches, players along with payrolls, adding field maps for different teams. This becomes a sole solution for managing tournaments – and the generic nature will enable it to be used by any and all the tournament stakeholders.
3. Use Case Diagram:

The actors and their functionalities for the system are as follows:-

- Tournament Manager: Organizes games, approves team managers and referees, and extends the system with more functionalities regarding team management.
- Team Manager: announces the team and is responsible for the hospitality of players.
- Referee: update availability and view match schedules.

- Health Manager: Manages players' health
- Player: Players requests to join a team which includes a request to participate. Check fixtures and schedules about pre match activities
- Coach: Select players for the team and register the team in the tournament.
- System Admin: Maintains the social events of the tournament and extends towards approval of games in the system and maintains the different types of access.
- Guests: Buys tickets
- Referee Manager: Manages referee schedules.
- API: Fetching data according to the profile(coach, referee, players, etc).



Agile methodologies are proven to be beneficial and can be followed when all engineering requirements are unknown or incoming. Since we don't have 40-50 pages engineering requirements document like in the Waterfall model but have a kickoff document with which we can come up with the user stories/use cases and breakdown the entire project into milestones such that each milestone will have working software by the end of each sprint and each deliverable will consist of sub features of the project - we will be prioritizing features and producing working output every sprint. Since we will be having working output every sprint we can make sure that we are heading in the right direction towards fulfilling the requirements and accomplishing the final product features. Also, with working software we can incorporate active feedback from the owner and realign every sprint - which is not possible in waterfall where small errors in the end cost very much to the development team.

Sprint 1 Goals:

**Sprint 1 goals:**

In the first sprint, we are planning to accomplish the following features:

Set up the entire tech stack in the working system which includes setting up

To come up with database design, create queries for tables, ER diagrams, MongoDB set up, physical schema creation.

To come up with API swagger documentation and sample payload, response, and header information.

To come up with a basic UI homepage that shows "About information".

To come up with User login capabilities from the UI and make sure that user information stays in persistent data storage.

Technology stack:

We are planning to use the MERN stack for the development of our website. MERN stands for MongoDB, Express, React, Node.

- MongoDB - document database
- Express(.js) - Node.js web framework
- React(.js) - a client-side JavaScript framework
- Node(.js) - the premier JavaScript web server