

**SPORTS MANAGEMENT**

**Project plan**

**PRT 455 Software Engineering Practice**

HELEN JOSE CHITTOOPARAMBIL

(S288262)

Ali

Imran

1. **Introduction**

## 1.1 Background

Arranging the sports and events can be tiresome and difficult process sometimes. The people are organising the sports events through meetings and publishing them on different social medias. But these procedures are sometimes time and money consuming. To make the sports management and organisation’s easy, we come up with a web application named sports management. The sport management is focused to allow the people to register different sports events, deciding venue and time for the sports and challenge others.

## 1.2 Aim

The aim of this project is to make a web based application called sports management. This application allows all the people to register into it and helping them to create sport events and selecting the venue and time. They also can challenge others for their sport event. The application not only focused on individual sport item but also focuses on the group. To make all this easier, the application allows the user to chat with each other.

## 1.3 Scope

The scope of this application is to help the user to identify the nearby sport events that they like and help the people to manage and organize the sport events without any meeting or publishing it into different places. By organizing the sports through this application, helps to reach maximum of audience. This application will provide the information so that the users can identify the events which are coming soon.

1. **Product Description**
   1. **Product Perspective**

If your product is self-contained, then it should be so stated here. Otherwise, if your product is part of a larger system, then relate the requirements of that larger system to the functionality of your product and identify the interfaces between that system and your product. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

* 1. **System Interfaces**

Specify the logical characteristics of the user, hardware, software and communication interfaces of the product.

* 1. **Major Product Functions**

Provide a summary of the major functions that the software will perform. A UML use-case diagram might be helpful.

* 1. **Constraints**

List the design constraints, hardware limitations, language and development environment requirements, etc. (if any) that will limit the developer’s options. Also list and quantify the expected limitations of the product’s functionalities.

* 1. **Assumptions and Dependencies**

List the assumptions about and the dependencies of the product, if any (design constraints should be listed in the previous section, not here).

* 1. **Development Process**

Briefly describe the process tool applied during the development of the product (note that we are using Scrum in CENG 599). Mention how change and configuration management will be handled.

1. **Requirements**
   1. **Functional Requirements**

List (or tabulate) and briefly explain product functional requirements, including GUI requirements. Mock-up screenshots, UML diagrams, free-style diagrams might be useful for illustrations.

* The application should be web based
* User receive correct information about the sport events from the application.
* User will be able to understand the GUI easily.
* System functionality is improved for better performance.

|  |  |
| --- | --- |
| Requirement | Description |
| New user Registration and Login | User signup, sign in with new user credential, password forgot and reset options. |
| Post events | Post the new events with date and venue. |
| Challenge Participants | Challenge other users for the sport events. |
| Chat with others | Chat for allowing the users to communicate with each other’s to manage the sports. |
| Mange Group event | The people can also arrange group events by organising group within the users of the application. |
| Manage the application | Administrator will maintain the software quality |

* 1. **Non-Functional Requirements**

List (or tabulate) and briefly explain product non-functional requirements such as performance, memory, database, reliability, availability, security, maintainability, portability and documentation requirements.

* Better response time between interfaces
* Required temporary memory to run the application
* Safety and security about the user information
* Speed and reliability of the application

1. **Design**
   1. **Architectural Design**

Describe the overall architectural design of the product. Mention the architectural design styles and patterns (not low level software design patterns) if you used any. A high-level UML class or component diagram might be useful for illustration.

* 1. **Design Viewpoint 1: <Name of Viewpoint>**
  2. **Design Viewpoint 2: <Name of Viewpoint>**

Choose 2 design viewpoints from **Table 1** that are most appropriate for your product, and describe your product from the selected viewpoints by providing appropriate UML diagrams.

1. **Test Plan**

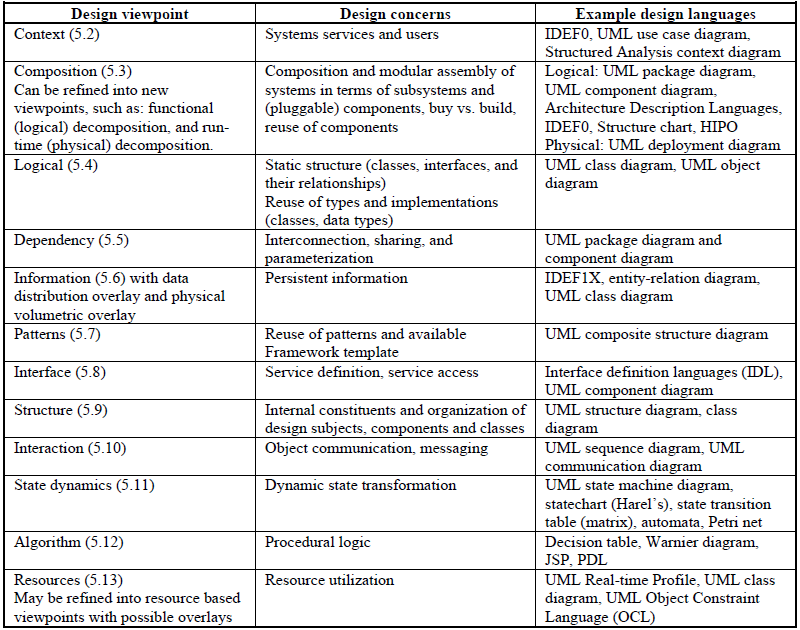
Note that testing for CENG 599 projects will be at the highest (system) level and on an as-required basis because of the time limitations.

Briefly describe how you are going to test the most important features of your product by mentioning the tools, techniques and methods you plan to use, and the metrics and the success criteria that will be defined for the verification and validation of your product.

1. **Release Plan**

Briefly describe your future plans about your product (for example, will you release it in the open source domain, will it be used as part of a larger product, will it be commercialized, etc.).

**Table 1.** Design Viewpoints (Reproduced from IEEE Std 1016™-2009, Table 1)



1. **Completed and Remaining Tasks**

List (or tabulate) all of the modules (or work packages, or tasks) of your project. For the modules you have completed, record the time you spent on each one. For the modules that still remain, record the estimated time to complete each one. Clearly mark the completed modules.

1. **References**
2. Sample reference with a proper paragraph and numbering format.
3. **Appendices**

(If any)

**Resources Used in Preparation of This Report Template**

1. IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
2. ISO/IEC/IEEE 29148, Systems and Software Engineering - Life Cycle Processes – Requirements Engineering.
3. IEEE Std 1016™-2009, IEEE Standard for Information Technology - Systems Design – Software Design Descriptions.
4. IEEE Std 829™-2008, IEEE Standard for Software and System Test Documentation.