# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* Official name: Football Field Reservation System
* Vietnamese name: Hệ thống đặt chỗ đá bóng
* Abbreviation: FFRS

### Problem Abstract

* The system provides a mobile application for user to support reserve field online. User have 2 options to reserve field.

+ If user has full team, user can search field, view timeline of field. User chooses field have free time as expected and reserve field.

+ If user wants match opponents, system will suggest opponent have same level with user. Rating point of user is rated by opponent, field owner after match finish. Based on result, system will calculate to identify ability of user.

* Field owner will have web application to manage field status. With system, field owner can provide promotion to attract players. When having field reservation request, system will arrange and choose suitable field automatically.
* In addition, we build web application for admin to manage profit. Besides, the admin will have right to lock account for bad users.

### Project Overview

#### Current situation

Below are the problems encountered in this project:

* New technique: Some team members are new to the techniques used in the project. The team need an amount of time to get familiar with those techniques.
* Design interface: The system serves users of varying degrees. So, we need an amount of time to analysis and evaluate how to build the easy to use user interface.
* Payment: The system processes payment online so that user must have electronic wallet.
* Server crash: All the needed data is stored in the server. So, if server crashes, field owner cannot get information and confirm field reservation request of users.
* Absence of team members: team members can get sick or unexpected problems.

#### The Proposed System

The system will have three sub-systems:

* An API application to serve API for mobile application and web application. API application is a center to process all business logic.
* A mobile application for users to perform find opponents, reserve field, feedback opponent and field, view promotion from field owner, manage and exchange discount voucher from bonus points, view notify from system.
* A web application for field owners and administrators. Field owner perform manage fields and field timeline, promote sale off price or free services, manage profits, view notify about field reservation request. Admin manage profits and lock account of bad users.

We build an algorithm to arrange field reservation request between the fields with other effectively because many field owner own multiple fields.

We use Paypal payments portals to perform transactions in the system, using Google API to get directions for user and find nearby location of field on the map.

##### Mobile Application:

* For user:

+ Manage profile

+ Find field

+ View timeline of field to reserve

+ Find opponent

+ Reserve field

+ Manage field reservation request

+ Payment online

+ View rewards, exchange voucher from bonus points

+ View notifications

* For guest

+ Become football team

##### Web Application

* For admin:

+ Manage profit

+ Receive report and lock user’s account

* For field owner:

+ Manage profile

+ CRUD fields

+ View status of field

+ View timeline of field

+ View notify about field reservation request

+ Rating users

+ Disable field/timeline of field

+ Manage profits

+ Manage promotion

* For guest:

+ Become a field owner

##### API Application

The server system takes responsibility to respond all the requests and also manages and processes data.

* Provide APIs for Mobile Application, Web Application
* Suggest opponent, field
* Arrange field reservation request
* Perform scheduled tasks

#### Boundaries of the System

The system does:

* Allow user to find fields.
* Allow user to find opponents.
* Allow user to reserve fields.
* Allow user to create a matching opponents request.
* Allow user to cancel field reservation request.
* Suggest opponents with same level.
* Suggest field have free time as expected.
* Allow field owner manage their field.
* Notify to field owner when field is reserved.
* Notify to user when request is accepted.
* Allow admin to manage profit.
* Allow admin to view report and lock user account.

#### Future plans

Current system is concentrated on core business flow. Therefore, some supporting features are restricted for the development team. These features may be expanded in the future:

* Build a system to manage each specific player in team to identify exactly ability of player.
* Organize tournament to attract more users and based on the results to assess skill of users more exactly.

#### Development Environment

##### Hardware requirement

* **For server:**

|  |  |  |
| --- | --- | --- |
|  | **Minimum Requirements** | **Recommended** |
| Internet Connection | Cable (4 Mbps) | Cable (8 Mbps) |
| Operating System | Ubuntu 12.04 LTS | Ubuntu 16.04 LTS |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| Computer Memory | 2 GB RAM | 4 GB RAM or higher |
| *Table 1: Hardware requirement for Server* | | |

* **For mobile:**

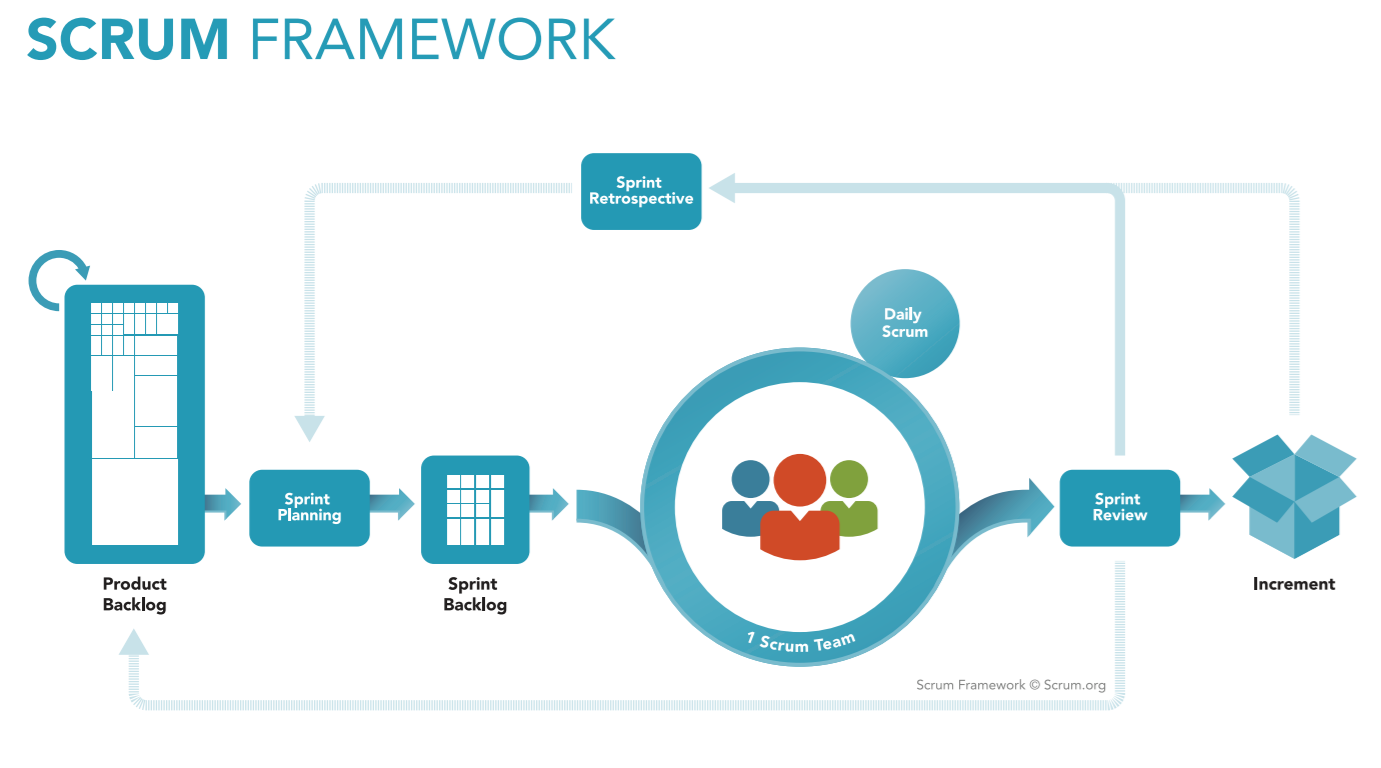
|  |  |  |
| --- | --- | --- |
|  | **Minimum Requirements** | **Recommended** |
| Internet Connection | Wi-Fi or 3G (4 Mbps) | Cable (8 Mbps) |
| Operating System | Android 6 | Android 7 |
| Device | Nexus 5 | Nexus 5 |
| *Table 2: Hardware requirement for Mobile* | | |

## Project organization

### Software Process Model

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

* In the team, there are 2 members not playing football. So, all members must work together in order to discuss about business logic and avoid misunderstanding.
* In the project there are many new technologies that need to be learned. With the Scrum model, the team can learn and develop in parallel to meet deadline.
* There is no leader, no hierarchy in team, so team members work cheerfully, stimulating the initiative and creativity of each member.
* The project implements a new idea, so maybe product owner change requirement or extend scope. The team will adapt to change better.



*Figure 1: Scrum Process*

(<https://www.scrum.org/resources/what-is-scrum>)

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Role in Scrum** | **Name** | **Responsibilities** |
| 1 | Product owner | Kiều Trọng Khánh | * Specify scope and user requirement. * Supervise the development progress. * Provide professional techniques and business analysis support. |
| 2 | Scrum master | Mai Minh Quý | * Create Sprint Backlog and Product Backlog. * Make sure the Scrum teams understand and follow the process. * Always be present to answer questions and give advice when product owner or scrum member needs. * Help the team master scrum artifacts such as: Sprint Backlog, Product Backlog, ... |
| 3 | Scrum team member | Mai Minh Quý  Trương Hữu Thành  Phan Minh Huấn  Phạm Trung Hiếu | * Designing database * Clarifying requirements * Prepare documents * GUI design * Coding * Testing |
|  | | | |

*Table 2: Roles and Responsibilities*

### Tools and Techniques

|  |  |
| --- | --- |
|  | Tool/Technique |
| Mobile Application | Android |
| Web Portal | CSS, HTML5, React JS, Java Script |
| Back-end | SpringBoot framework, Java, JPA |
| Temporary Storage | Redis |
| IDE | IntelliJ 2017.2.4, Android Studio 2.3.3 |
| Database | MySQL Server 5.7.19, MySQL Workbench 6.3.9 |
| Modeling tools | Star UML 2.8.0 |

*Table 3: Tools and Techniques*

## Project Management Plan

### Product Backlog

Refer to Appendix

### Sprint Backlog

Refer to Appendix

### Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverable** | **Deliverable date** | **Deliverable location** | **Note** |
| 1 | Introduction Document, Task list |  | FU - LMS | Report No. 1 |
| 2 | Software Project Management Plan |  | FU – LMS | Report No. 2 |
| 3 | Software Requirements Specification |  | FU – LMS | Report No. 3 |
| 4 | Software Design Description |  | FU – LMS | Report No. 4 |
| 5 | Software Test Documentation Guide Implementation (Coding) |  | FU – LMS | Report No. 5 |
| 6 | Software User’s Manual |  | FU - LMS | Report No. 6 |
|  | | | | |

*Table 4: Deliverables*

* For each Sprint, deliverables are potentially shippable products, which can be a part of document or prototype implemented based on the project’s core flow.
* Each Sprint has a fixed duration of one weeks.

### All Meeting Minutes

All meeting documents could be found where

## Coding Convention

### Naming convention

* Camel Case: variable name, function name.
* Pascal Case: class name.
* Snake Case: name of table and field of database.
* Lower Case: package name.

### Quantity convention

* Function should not exceed 30 lines (Martin Lippert-rule 30)
* Class should not exceed 500 lines (Clean Code)
* A function must not exceed 5 parameters (should keep <= 3) (Clean Code)
* A function only performs one task, in which case, two things are allowed, but the function name must specify this, ex: increatePointAndSaveToDb
* One line should not be longer than 80 characters (Oracle)
* Nested statements up to 4 levels

### Down the line convention

According to Oracle

* If there are commas, then the row after the comma,
* Down before the + operator - ...
* If there are multiple levels nested, then go down each level
* New row should start at the same column with the same level command.

### Comment convention

* Only comment should be used in case of writing documentation for library, attachment information for class or function in complex algorithm ...

**References:**

* <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>
* <https://laptrinhrails.wordpress.com/>