**Requirement Specification of Sports Facilities Booking System**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide a detailed description of the functionalities of Sports Facilities Booking System. This document will cover each of the system’s intended features, as well as the technical dependencies.

**1.2 Project Scope**

The system should be built as a web application, focusing on sports facilities booking functionalities and related services. Initially the system should support racquet sports (tennis, squash, table tennis, paddle, etc.). In the future the sports range may extend to ball games (soccer, basketball, etc.).

The web application should be able to support all major browsers (Chrome, Firefox, Safari, IE10+), and should be mobile friendly.

**2. Overall Description**

**2.1 Project Perspective**

The Sports Facilities Booking system is an easy to use, mobile friendly product, consisted with two major components:

* Web service:

The web service is running on server side, exposing a list of secured interfaces supporting general CRUD operation

* User Interface:

The UI is running on client side, either on desktop or mobile browsers. It provides functions to users to book/view/search sports facilities, as well as some additional services (e.g. map view). UI communicates with web service to perform CRUD operations.

**2.2 Product Users**

The users of this product are divided into 4 types:

2.2.1 Site administrators

Site administrators are users who manage the system. Their responsibilities are:

* Manage registered clubs.
* Manage signed up users.
* Publish upcoming events.
* Advertise new products/clubs/facilities/plans.
* Send notifications to other users.

2.2.2 Club administrators

Club administrators are generally the owners/crews of clubs. They manage club information, other club members and club events. Below is a list of the responsibilities:

* Maintain valid club information, address and contact details.
* Manage other club members.
* Administer user bookings.
* Add/update facilities information and restrictions.
* Publish events.
* Handle payments.

2.2.3 General members

These users are unprivileged users who join clubs or book facilities.

2.2.4 Unregistered users

Although the application should provide easy to use interface for user registration, there will still be a big amount of unregistered users. They could be new users, or users who just don't want to register. They may explore, search clubs, and even book the facilities they interest in.

**2.3 Product Features**

Functions and features are divided into 4 major categories:

* Club administration

This type of functions are oriented to club administrators, for managing club details and memberships.

* General functions

General functions and pages such as user registration, landing.

* Membership functions

Membership functions allow registered users to manage their person information and bookings.

* Site management

This type of features focuses on additional facilities, such as broadcasting marketing emails, publishing news.

The list below outlines main features and functionalities of each category.

2.3.1 Club Administration functions should be able to:

* Club registration.
* Club profile management.
* Advertise club facilities.
* Define booking restrictions.
* Manage club members.
* Manage roles.
* Publish events.
* Handle payment.

2.3.2 General functions should provide:

* User landing page.
* Club searching functions.
* Pricing plans page.
* Terms and conditions page.
* Site description page.
* User sign up functions.
* Multilingual support.
* Book/cancel club facilities.

2.3.3 Membership functions should be able to:

* Manage personal information.
* Join clubs.
* Receive notifications and upcoming events.
* Manage bookings.

2.3.4 Site Management functions should be able to:

* Broadcast marketing emails.
* Publish news.
* Send notifications to members.

**3. System Features**

**3.1 Club Administrations**

3.1.1. Club registration

The system should support club registrations. It provides a club registration page with form controls, requiring the club crews to input their club name, description, address, contact details and additional mandatory information.

3.1.2 Club profile management

Club crews should be able to set their club profile via a profile settings page. The settings contains:

* General club information such as name, address, description and contact details.
* Attach/detach photos for glance purpose.
* Club social network presence and link to home page.

3.1.3 Advertise club facilities

Each club should have one or more types of sports facilities. The club staff should have the ability of advertising their facilities through a page. The page contains a set of form controls, which collects facility name, description, pricing and related information from the user.

3.1.4. Define booking restrictions

Many facilities have particular limitations. For instance, they may limit the maximum duration of a booking, or require club members to book ahead of a certain period, or restrict the amount of players, etc. The application should produce an interface allows the users to set these restrictions.

3.1.5 Manage club members

Club crews should have the capability of managing their club members via the application. The activities include:

* Approve new member requests.
* Edit some information of club members.
* Receive notifications of pending member requests.
* Explore all club members.

3.1.6 Manage roles

Club staff should be able to manage the roles of their members. The types of roles include:

* General members: Unprivileged club members.
* Professor: Professional players.
* Administrator: These are the club crews who can manage club information and other members via the application.

3.1.7 Publish events

Clubs often organize some activities. The system should provide an interface that allows club administrators to publish the upcoming events to their members. Once an event is published, the club members receive in-system notifications, as well as emails.

3.1.8 Handle payment

After club members placed a reservation, they may pay the fee in advance (pre payments are optional). The payments go through third party platforms. Once the application received new payments, club crews should handle them, whether accept or reject. If they reject the payments, they must produce sufficient reasons.

**3.2 General Functions**

3.2.1 User landing

The application should provide a good landing page to welcome new users. The page should contain most important features, link to other functions, buttons to sign up/log in, search bar to search clubs, etc.

3.2.2 Club searching

Club searching is an important function. Good search features will attract more users. There are three major search facilities:

* Search toolkit:

Search toolkit is a self-contained, reusable component. It provides a set of controls allow users to search clubs based on certain conditions. Conditions include:

* Club name.
* Club location (street, region, city, country).
* Date and time.
* Club/facility type.
* Map view:

Map view provides another searching experience. Users should be able to find the clubs on a map view. They could also select a club to view some information. This is often helpful to users who wish to find clubs in a specific region.

* Restrictions setter:

Users often wish to have the ability to search 'on the fly', which means once they placed a searching and got some results, they may want to narrow the results. The restrictions setter provides this kind of function, which filters the results.

3.2.3 Pricing plans

The system should display clear pricing plans to users.

3.2.4 Terms and conditions

The system should show terms and conditions to users (although almost nobody will read it).

3.2.5 Site description

This page describes the important features and services of the system.

3.2.6 User sign up

User sign up should be easy to use, the only required information are user email, full name and preferred password, whereas other information may be set later. The application should also support signing up from third party accounts, such as Google account or Facebook.

3.2.7 Multilingual support

The system should have the capability to support multiple languages. Initially it must support English.

3.2.8 Manage bookings

Unregistered users should also be able to place reservations on club facilities. The system will send them an order ID. They could retrieve/modify/cancel the booking by producing the order ID and answer some security questions.

**3.3 Registered member**

3.3.1 Manage personal information

Registered member should manage their personal information through an account settings interface. They could modify most of their details.

3.3.2 Join clubs

A registered member could join multiple clubs they interested in. A pending request will be generated with a notification to the club administrators.

3.3.3 Receive notifications and upcoming events

The system should display notifications from site administrators or club crews to the registered members, as well as showing the upcoming events of clubs they have joined.

3.3.4 Manage bookings

The members should be able to place their bookings by general booking functions. The application should provide a clear interface allows the members to view/modify/cancel their bookings.

**3.4 Site management**

3.4.1 Broadcast marketing emails

There should be an interface gives the emails broadcasting ability to site administrators. The emails should only be broadcasted to users who subscribed this kind of service. Users should be able to subscribe/unsubscribe the advertisements easily through a page.

3.4.2 Publish news/advertisements

The application should give site administrators the ability of publishing news/advertisements to some pages, for example, publish some attractive clubs in the landing page, or place some discount news in some prominent locations.

3.4.3 Send notifications to members

The system provides the notification functions to site administrators. They could notify some members.

**4. Technical Dependencies**

**4.1 Python 2.7**

Python is the major language for server side code.

**4.2 Django 1.6.x**

Server side uses Django framework to manage HTTP/HTTPS requests and sessions, as well as data models and database access.

**4.3 HTML5/CSS3/Bootstrap 3.x**

These are the major technologies for building the very front views.

**4.4 JavaScript/AngularJS**

JavaScript/AngularJS manages core functions of UI.

**4.5 PayPal API**

PayPal API manages the payments.

**4.6 Google Maps API**

This API supports map views for club searching.

**4.7 SMTP**

SMTP powers the email broadcasting ability.

**4.8 GIT**

The version control system is GIT.