**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

Football Pitches Booking

|  |  |
| --- | --- |
| **Group 10** | |
| **Group Members** | Nguyễn Duy Phương – SE60761 – [phuongndse60761@fpt.edu.vn](mailto:phuongndse60761@fpt.edu.vn)  Nguyễn Hữu Hoàng Giang – SE60606 – [giangnhhse60606@fpt.edu.vn](mailto:giangnhhse60606@fpt.edu.vn)  Nguyễn Đức Thịnh – 60398 – [thinhnd60398@fpt.edu.vn](mailto:thinhnd60398@fpt.edu.vn)  Nguyễn Quốc Khánh – 60556 – [khanhnq60556@fpt.edu.vn](mailto:khanhnq60556@fpt.edu.vn) |
| **Supervisor** | Lại Đức Hùng |
| **Ext Supervisor** |  |
| **Capstone Project code** | FPB |

- Ho Chi Minh, 01/2014 –

Table of Contents

[Definitions, Acronyms, and Abbreviations 3](#_Toc380231193)

[Report No.1: Introduction 4](#_Toc380231194)

[1.1. Initial idea of group: 4](#_Toc380231195)

[1.2. Overview of existing system: 4](#_Toc380231196)

[1.3. Benefit of expected system: 4](#_Toc380231197)

[Report No.2: Project Management Plan (PMP) 5](#_Toc380231198)

[2.1. Problem Definition 5](#_Toc380231199)

[2.1.1. Name of this CapStone Project. 5](#_Toc380231200)

[2.1.2. Problem Abstract 5](#_Toc380231201)

[2.1.3. Project Overview 5](#_Toc380231202)

[2.2. Project organization 6](#_Toc380231203)

[2.2.1. System Process Model 6](#_Toc380231204)

[2.2.2. Roles and Responsibilities 7](#_Toc380231205)

[2.2.3. Tools and Techniques 8](#_Toc380231206)

[2.3. Project management plan 8](#_Toc380231207)

[2.3.1. Tasks: 8](#_Toc380231208)

[2.3.2. Task Sheet: Assignments and Timetable 10](#_Toc380231209)

[2.4. Convention Rules 10](#_Toc380231210)

[2.5. Other material 10](#_Toc380231211)

[Report No.3: System Requirements Specifications (SRS) 11](#_Toc380231212)

[3.1. User requirement specification 11](#_Toc380231213)

[3.1.1. Guest requirement 11](#_Toc380231214)

[3.1.2. Unauthorized member 11](#_Toc380231215)

[3.1.3. Authorized member 11](#_Toc380231216)

[3.1.4. Stadium Owner 11](#_Toc380231217)

[3.1.5. Stadium Staff 12](#_Toc380231218)

[3.1.6. Website Master 12](#_Toc380231219)

[3.1.7. Website Staff 12](#_Toc380231220)

[3.2. System requirement specification (Specific Requirements) 12](#_Toc380231221)

[3.2.1. External Interface Requirements 12](#_Toc380231222)

[3.2.2. Functional Requirements 13](#_Toc380231223)

[3.2.3. Non-Functional Requirements 88](#_Toc380231224)

[3.3. Entity Relationship Diagram 89](#_Toc380231225)

## Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| No. | Keyword | Definition |
| 1 | B2B2C | Stand for Business to business to customer.  In this system, first business is the organization that provides stadium owner to register their stadiums. The second business is stadium owner, who provides hire stadium service, and customer is the user, who want to hire field by this system. |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

## Introduction

### Initial idea of group:

Football Pitches Booking is a B2B2C (Business to Business to Customer) website that allows customers to book field online. Beside that it helps the stadium owners with advertising football stadiums and managing reservation. In addition, the website enables its admin to manage information of those customers and stadium owners.

In this situation, the organization is first business provide services

### Overview of existing system:

For customers: Existing system just provides customers with address and contact information of stadium owners, then customers have to contact with the stadium owner to book space. They have to call each stadium owner individually to check whether the field is available or not. Moreover, the current system does not have either functions to make statistics of customers' habits or some smart functions like offering options/suggestion to customers, which is time-consuming and inconvenient to customers.

For the stadium owners, sometimes they make mistakes in management such as duplicate reservation, which leads to bad result: they could lose their customers as well as profit. Beside the management is executed manually, they do not have tools to optimize the timing to utilize the fields, this has bad impact on profit. Sometimes customers cancel their reservation, the stadium owners will suffer the loss. Apart from cancellation, the website does not have functions to create events to appeal customers to take part in.

For the Web masters, the shortage of functions of the website is the obstacle to persuade stadium owners to pay for the services.

### Benefit of expected system:

Football Pitches Booking makes the reservation easier and save time for customers. Once they have booked, they don’t have to worry there will be a duplicate reservation caused by stadium owners' mistake. The website's smart functions help users to find the appropriate field by searching stadium around customers' position, comparing price among stadiums. The benefits to customers are saving time, getting more information and enjoying discount or voucher from the stadium owners.

This system helps stadium owner to manage their field easily, reduce the implement process of business operations, save time and costs. The automatic system helps to prevent the risk of duplicate booking and to minimize the loss caused by reservation cancellation. The self-arrangement and self-timing offered to customers makes the utilization of field possible, reduce the dead time and increase profit for the stadium owners. Furthermore, stadium owners can use the website as a channel to advertise their services as well as communicate with their customers; hence, they can win more customers and gain more prestige through rating-feedback functions. Analyzing and evaluating the feedback to make the business plan more reasonable.

For the webmaster, with many smart and outstanding functions of the new system, they have more chances to increase the profit by attracting stadium owners to use this service.

## Project Management Plan (PMP)

### Problem Definition

#### Name of this CapStone Project.

This Capstone Project Name is “Football Pitches Booking” or “FPB” in short.

#### Problem Abstract

Ho Chi Minh City (HCMC for short) has over seven million of population, is Vietnam’s centre of commerce and the country’s biggest city by far, though not its administrative capital. HCMC is a youthful city, almost residents are student and clerical worker. Even live in a busy environment, they still take free time to play sport for their health and their relaxation. One of the most popular sport is football. Nowadays more and more spontaneous football games are opened. So the number of artificial grass field increase faster. Based on the latest count, there are more than 100 hundred stadium around district 1 and 3 in 2012, it include five, seven and eleven-man field. And the amount of stadium around HCMC increases constantly.

With the high demand of hiring field, almost stadium owners still manage their ground manually. Player used to contact with stadium owner to book field by phone, and some problem was occurred. They have to call stadium owner to ask for an available booking time, sometimes the manager’s mistake makes the duplicated reservation. This process is wasting time and making more inconvenience.   
There are some websites allow player to search stadium and book it, but it still has limitation. Player have to wait for confirmation of stadium owner after they check this booking time. Manager still arrange reservation manually and mistake still happens.

To meet that demand, we develop a website that allow user to book field online. They can find an appropriate field base on booking time, location or field type. Some automatic function help them with saving time by customer habits statistic. System arranges the booking time and allows manager to change it manually if necessary, it minimize doubling reservation. Some promotion functions help stadium owner to attract their customer. Obviously player get more priority when they register this system.  
In summary, Football Pitches Booking is develop for attending to customer, stadium owner and so website administrator. It makes the reservation becomes easily, helps service provider manage precisely and encourages business efficiently.

#### Project Overview

##### The Current System

According to our research, there has never been a Football pitches booking online system like the one we are developing. There are some websites that provides stadium’s information. We divide it into two types: single stadium service (Website for one stadium owner) and multiple stadium service (Website for more than one stadium owner).

Single stadium service:

There are some services applied for one stadium. For example: <http://tuyensonsport.vn>; <http://sanbongdamini.seonhatnghe.com>

* User just books field belonging to this stadium. The way to book is call stadium owner.
* It is easy to makes mistake in management when executing manually such as duplicate reservation.

Multiple stadium service:

* Some websites collect stadium and provide information to user and they must contact stadium owner then. They have to call each stadium owner individually to check whether the field is available or not.
* Some websites have the booking online system, for example: http://datsan.com.vn  
  Datsan.com.vn is a website that allows users to pick a field and book it online. This website provides searching function to make it more convenient for users. Firstly, users have to choose field, then they fill in the booking form; one point of this form that must be noticed is the booking time. After that, they wait for the stadium owner's call to confirm this reservation.    
  Maybe the booking time that user booked has been occupied. So when the stadium owner calls back, user must change the booking time. It is a waste of time and it causes inconvenience to both customers and service providers.

##### The Proposed System

The system that needs to be established should have these abilities:

* The website must have more categories that are built on the basis of districts,  types of field etc. to make it more convenient for users.
* The website has "search stadium" function. Information can be searched based on location, booking time, type of field (5, 7 or 11 men field). In addition, it should have some auto functions such as suggesting appropriate stadium based on customer habits.
* This website must implement account level to make good offer to users.
* Management function for stadium owners to manage their fields, arrange booking time manually if necessary and manage their users.
* Promotion function to attract customers.
* The website enables players to create matches to challenge other teams.

##### Boundaries of the System :

* Website focus about booking field for player, managing stadium for stadium owner
* Website don’t intervene about finance problem of stadium owner.

##### Development Environment:

**Hardware requirements:**

* + A computerwith hardware that capable of running Windows 7 operating system, 2GB of RAM for addition software

**Sotfware requirements:**

* + Operating System : Windows 7 Professional (32 or 64 bit).

### Project organization

#### System Process Model

The software will be developed by using waterfall model. The waterfall model developing process includes five main phases:



Figure 1: Waterfall model

* + - Requirement analysis and software definition: The first phase is also the most important phase of the process which may cause great effect to other phases. This phase includes gathering requirements from customer, or from other source like the Internet, and producing the most detail and accurate software definition.
    - System design: This phase is fundamental for implementation phase. Based on customer’s requirements to create logical modules, and definite their inter relations. Using algorithm and diagram to describe implementation of those modules.
    - System implementation and Unit testing: Developing software modules follow detail designs, and doing unit testing for each module.
    - Integration and System testing: Testing output, performance in modules integrating process, and retests all functions of whole system.
    - System deployment and maintenance: After testing completely, the software is handle over client, developing team will respond for maintenance of the system.

#### Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Full Name | Role | Responsibilities |
| Lại Đức Hùng | Supervisor, Project Manager | * Tracking and managing process * Advising ideas and solutions * Specify requirement |
| Nguyễn Duy Phương | Team Leader, Business Analyst, Developer, Tester | * Tracking and managing process * Analyzing requirement * Writing documents and reports * Designing database * Creating and managing project’s structure * Creating and managing project configuration * Creating project’s plan * Designing GUI * Coding and testing |
| Nguyễn Hữu Hoàng Giang | Team Member, Business Analyst, Developer, Tester | * Analyzing requirement * Writing documents and reports * Designing database * Designing GUI * Coding and testing |
| Nguyễn Đức Thịnh | Team Member, Business Analyst, Developer, Tester | * Analyzing requirement * Writing documents and reports * Designing database * Designing GUI * Coding and testing |
| Nguyễn Quốc Khánh | Team Member, Business Analyst, Developer, Tester | * Analyzing requirement * Writing documents and reports * Designing database * Designing GUI * Coding and testing |

#### Tools and Techniques

##### Tools

* Visual Studio 2012
* Microsoft SQL Server 2008 R2
* Star UML 5.0
* Microsoft Office 2013
* Tortoise SVN
* Google Code
* Log4net

##### Techniques

* .NET Framework 4.0
* ASP.NET MVC 4
* UML Modeling 2.0
* HTML5, CSS3, jQuery
* LINQ

### Project management plan

#### Tasks:

##### Study requirement and review technology

|  |  |
| --- | --- |
| Description | Discussing with project manager to understand requirement, business process.  Self-review the technology will be used in project. |
| Deliverables | N/A |
| Resources Needed | 12 man-days |
| Dependencies and Constraints | N/A |
| Risks | Requirement too strange to develop.  The project is not feasible.  The chosen technology is not feasible. |

##### Project Introduction

|  |  |
| --- | --- |
| Description | From requirements and result of researching old system create project introduction document. |
| Deliverables | Project Introduction document. |
| Resources Needed | 4 man-days |
| Dependencies and Constraints | Depend on result of studying requirement. |
| Risks | Lack of experience in researching old system  Do not understand requirements clearly. |

##### Project Management Plan

|  |  |
| --- | --- |
| Description | Prepare project plan and task list. |
| Deliverables | Project Management Plan (PMP) document. |
| Resources Needed | 4 man-days |
| Dependencies and Constraints | Depend on result of studying requirement. |
| Risks | Lack of experience in scheduling may lead to fail in keeping deadline. |

##### System Requirement Specification

|  |  |
| --- | --- |
| Description | Specify system requirements |
| Deliverables | System Requirement Specification (SRS) document |
| Resources Needed | 28 man-days |
| Dependencies and Constraints | Depend on result of studying requirement.  Follow the SRS Template.  Keeping the deadline in plan. |
| Risks | Do not understand requirements clearly. |

##### System Design Description

|  |  |
| --- | --- |
| Description | Design system architecture, describe system component, user interface design, database in details level for developer. |
| Deliverables | System Design Description (SDD) document |
| Resources Needed | 38 man-days |
| Dependencies and Constraints | Clear requirements of system.  Follow design documents template. |
| Risks | Lack of experience in system designing.  Do not understand requirements clearly.  Requirement change quickly. |

##### System Implementation and Testing

|  |  |
| --- | --- |
| Description | Implement System follow System Design Description document.  Execute unit testing.  Integrate all parts into complete system.  Integration test and system test. |
| Deliverables | Source code.  Unit test document. |
| Resources Needed | 111 man-days |
| Dependencies and Constraints | Follow SDD.  Keeping schedule. |
| Risks | Developers may not misunderstand the design.  Testers are not experience in testing.  Lack of experience in integration.  Resources cannot keeping schedule. |

##### System User’s Manual

|  |  |
| --- | --- |
| Description | Prepare guide document for users. |
| Deliverables | System User’s Manual document. |
| Resources Needed | 6 man-days |
| Dependencies and Constraints | Depend on SDD document.  The guide must easy to understand. |
| Risks | User may not understand the guide |

#### Task Sheet: Assignments and Timetable

Reference to “Football Pitches Booking.mpp” in same folder.

### Convention Rules

Reference to Microsoft C# Programming Guide – C# Coding Conventions.

<http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

### Other material

N/A.

## System Requirements Specifications (SRS)

### User requirement specification

Football Pitches Booking is a website that allows customers to book field online. Beside that it helps the stadium owners with advertising football stadiums and managing reservation. In addition, the website enables its admin to manage information of those customers and stadium owners.

More detail, the system should have list of features below:

#### Guest requirement

* View list of stadium
* Find and view the stadium’s information
* Find available stadium at specific date, time and duration
* Register to become member of website

#### Unauthorized member

* View list of stadium
* Find and view the stadium’s information
* Find available stadium at specific date, time and duration
* Login
* Recovery password

#### Authorized member

* Update account profiles
* Booking stadium
* Cancel stadium booking
* View booking history
* Rating stadium
* Review stadium
* Create a game
* Find game and join
* Feedback website
* Receive appropriate stadium suggestion
* Request for stadium join into system

#### Stadium Owner

* Update stadium profiles
* Manage fields of stadium:
  + Add field to stadium
  + Update number of field
  + Update field type of field
  + Update price table of field
  + Remove field from stadium
* Manage reservation of stadium:
  + Create new reservation
  + Update status of reservation
  + View reservation detail
  + View list reservations of stadium
* Manage promotion of stadium
  + Create new promotion
  + Update promotion details
  + Delete promotion from stadium
  + View promotion detail
  + View list promotion of stadium

#### Stadium Staff

* Manage reservation of stadium
* Manage promotion of stadium

#### Website Master

* Manage stadium:
  + View list of joining request
  + Update joining request status
  + Add new stadium
  + Update stadium status
* Manage member ranking
  + Create new rank
  + Update rank
  + Remove rank
  + View rank detail
  + View list rank
* Manage advertisement
  + Create new advertisement
  + Update advertisement detail
  + Remove advertisement
  + View advertisement detail
  + View list advertisement
* Manage user
  + Update account status
  + Update role
* Manage Website Block
  + Update block content

#### Website Staff

* Manage advertisement
  + Update advertisement detail
  + View advertisement detail
  + View list advertisement
* Manage user
  + Update account status
* Manage Website Block
  + Update block content

### System requirement specification (Specific Requirements)

#### External Interface Requirements

##### User Interfaces

* The interface of website is simple and friendly
* Main function with multistep should be careful note, easy to understand

##### Hardware Interfaces

N/A

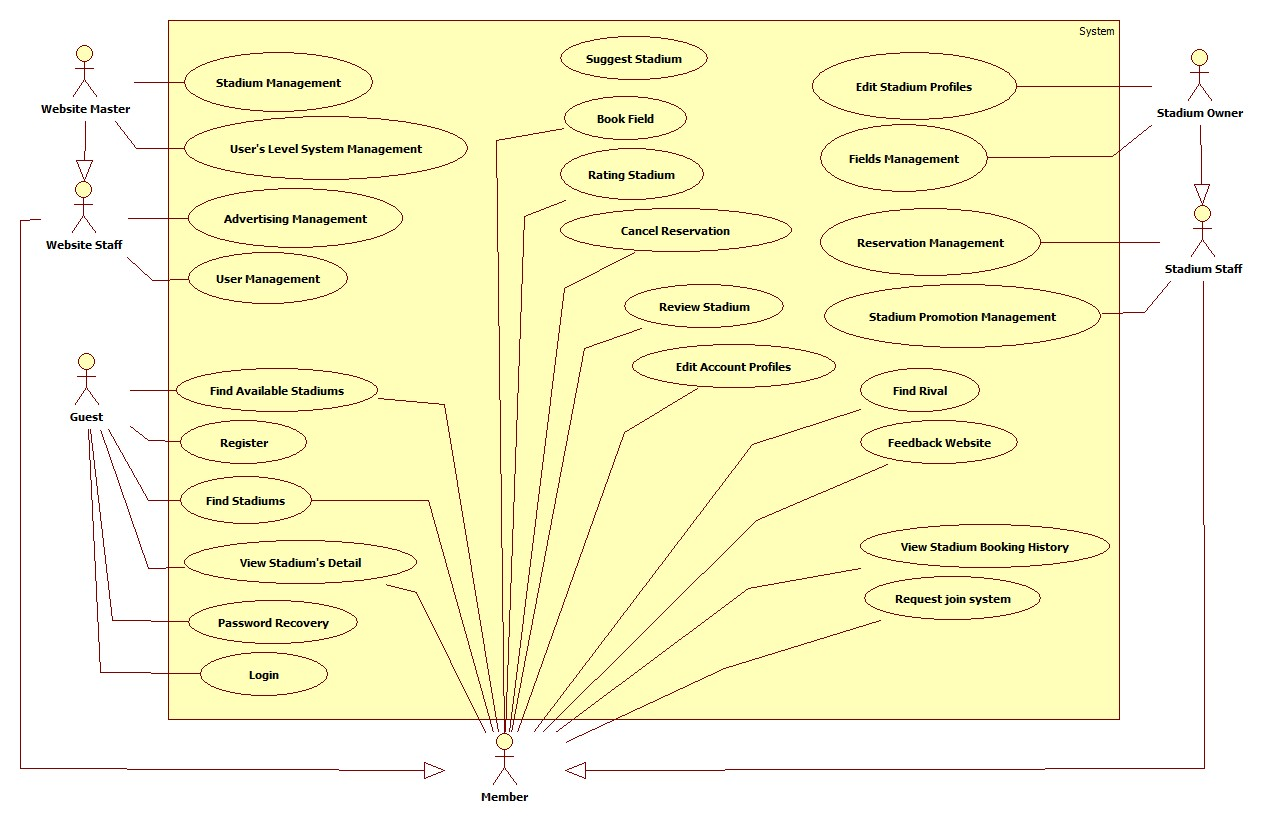
##### System Interfaces

N/A

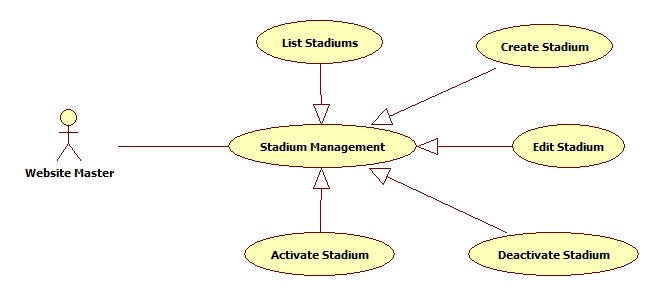
##### Communications Protocol

Website using HTTP protocol for communication with the web browser and the web server

#### Functional Requirements



##### Stadium Management



###### List Stadiums

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-001 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_001 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Stadium | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to see list of stadium  **Goal:**  List all Stadium in Website  **Triggers:**  Actor go to Stadium Management page  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | Display list of stadium |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Create Stadium

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-002 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_002 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Stadium | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to Create stadium in website  **Goal:**  Create Stadium in Website  **Triggers:**  Actor click Create button Create stadium in website  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Create to create a stadium | The system will respond to Create stadium page | | 2 | Enter information of stadium. Click on Create button to finish | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to Stadium Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Edit Stadium

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-003 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_003 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Stadium | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to edit a stadium  **Goal:**  View and change stadium information  **Triggers:**  Actor click on stadium in list of stadium  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Stadium that actor want to edit | Display stadium information | | 2 | Enter stadium information  Click on Save button to save information | Display successful message | |  |  |  |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to Stadium Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

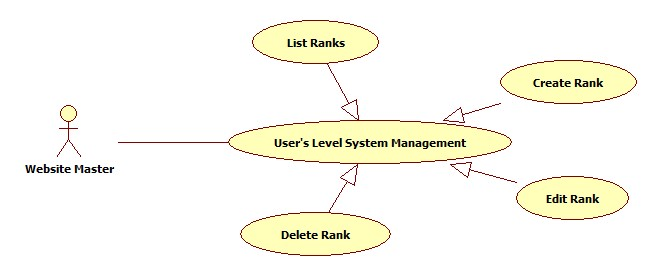
###### Activate Stadium

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-004 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_004 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Activate Stadium | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to activate stadium in website  **Goal:**  Activate a stadium in system  **Triggers:**  Actor click on Activate button to active the stadium in website  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Activate button of stadium | Display confirmation message | | 2 | Click on OK button | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | Hide confirmation message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Deactivate Stadium

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-005 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_005 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Deactivate Stadium | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to deactivate stadium in website  **Goal:**  Deactivate a stadium in system  **Triggers:**  Actor click on Deactivate button to de-active the stadium in website  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Deactivate button of stadium | Display confirmation message | | 2 | Click on OK button | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | Hide confirmation message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### User’s Level System Management



###### List Ranks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-006 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_006 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Ranks | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to see list of user’s rank of system  **Goal:**  See list of user’s rank  **Triggers:**  Actor go to User’s Level System Management page  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | Display list of user’s rank |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Create Rank

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-007 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_007 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Rank | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to create a rank in website  **Goal:**  Create Rank in Website  **Triggers:**  Actor click Create button Create rank in website  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Create button to create a rank | The system will respond to Create Rank page | | 2 | Enter information of stadium. Click on Create button to finish | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to User’s Level System Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

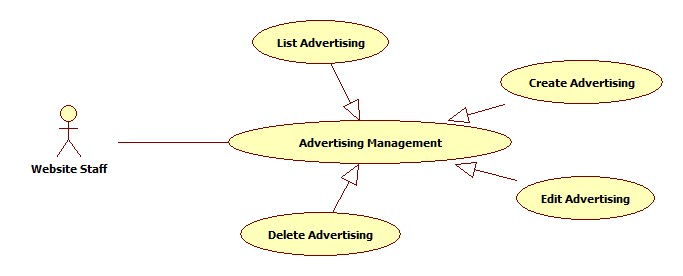
###### Edit Rank

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-008 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_008 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Rank | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to edit a rank  **Goal:**  View and change rank information  **Triggers:**  Actor click on rank in list of rank  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Rank that actor want to edit | Display stadium information | | 2 | Enter rank information  Click on Save button to save information | Display successful message | |  |  |  |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to User’s Level System Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Delete Rank

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-009 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_009 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete Rank | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Master  **Summary:**  This use case describe how to delete a stadium in website  **Goal:**  Delete a stadium in system  **Triggers:**  Actor click on Delete button to delete the stadium in website  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Delete button of stadium | Display confirmation message | | 2 | Click on OK button | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | Hide confirmation message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Advertising Management



###### List Advertisings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-010 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_010 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Advertisings | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to see all advertising in website  **Goal:**  See all advertisings  **Triggers:**  Actor go to Advertising Management page  **Preconditions:**  N/A  **Post-conditions:**    **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | Display list of advertising in system |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Create Advertising

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-011 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_011 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Advertising | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to create a Advertising in website  **Goal:**  Create Advertising in Website  **Triggers:**  Actor click on Create button to create Advertising in website  **Preconditions:**  N/A  **Post-conditions:**    **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Create button | The system will respond to Create Advertising page | | 2 | Enter information of advertising  Click Create button | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to Advertising Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Edit Advertising

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-012 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_012 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Advertising | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how edit an advertising  **Goal:**  View and change Advertising information  **Triggers:**  Actor click on Advertising on advertising list  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Advertising on advertising list | The system will show the Advertising Information Page | | 2 | Update advertising information  Click Save button | Display successful message | |  |  |  |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to Advertising Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Delete Advertising

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-013 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_013 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Delete Advertising | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to delete an advertising of website  **Goal:**  Delete an advertising  **Triggers:**  Actor click on Delete button correspond with Advertising  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Delete button to delete the advertising | Display confirmation message | | 2 | Click OK button | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | Hide confirmation message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### User Management

###### List Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-014 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_014 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Users | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to see all users in website  **Goal:**  See all Users in website  **Triggers:**  Actor go to User Management page  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | Display all users in system |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Edit User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-015 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_015 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit User | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to edit a user information  **Goal:**  View and change user information  **Triggers:**  Actor click on user in list of users  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on user in list of users | The system will show the User Information Page | | 2 | Update information  Click on Save button to save information | Display successful message | |  |  |  |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to User Management page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Activate User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-016 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_016 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Activate User | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to activate a user  **Goal:**  Activate a user  **Triggers:**  Actor click on Activate button to active user  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Activate button to active the user | Display confirmation message | | 2 | Click OK button | Show successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | Hide confirmation message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

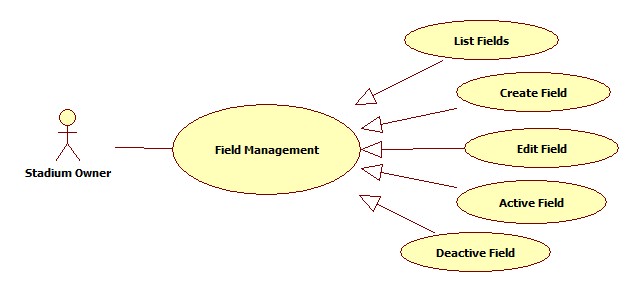
###### Deactivate User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-017 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_017 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Deactivate User | | | |
| **Author** | KhanhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Website Staff  **Summary:**  This use case describe how to deactivate a user  **Goal:**  Deactivate a user  **Triggers:**  Actor click on Deactivate button to de-active user  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Deactivate button to de-active the user | Display confirmation message | | 2 | Click OK button | Show successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | Hide confirmation message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Edit Stadium Profiles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-018 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_018 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Stadium Profiles | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Owner  **Summary:**  This use case describes how to edit stadium profiles  **Goal:**  Edit profiles information of stadium  **Triggers:**  Actor click on Edit button  **Preconditions:**  Login as Stadium Owner role  **Post-conditions:**  Successful message is displayed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Edit button | Display Edit form with filled information | | 2 | Enter new profiles information of stadium  Click Save button | Update new profiles information |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Fields Management



###### List Fields

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-019 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_019 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Fields | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Owner  **Summary:**  This use case describes listing fields of stadium.  **Goal:**  List all fields of stadium.  **Triggers:**  Actor go to Fields Management page  **Preconditions:**  Login as Stadium Owner role  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | Show list of fields. |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Create Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-020 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_020 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Field | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Owner  **Summary:**  This use case describes how to create a field of stadium.  **Goal:**  Create a field  **Triggers:**  Actor click on Create button  **Preconditions:**  Login as Stadium Owner role  **Post-conditions:**  Successful message is displayed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Create button | Display Create form | | 2 | Enter information of field  Click on Save button | Create new field |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Edit Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-021 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_021 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Field | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Owner  **Summary:**  This use case describes how to edit a field of stadium.  **Goal:**  Edit information of field  **Triggers:**  Actor click on Edit button  **Preconditions:**  Login as Stadium Owner role  **Post-conditions:**  Successful message is displayed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Edit button | Display Edit form with filled information | | 2 | Enter new information of field  Click on Save button to finish | Update new information of field |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Discard recent filled information and back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

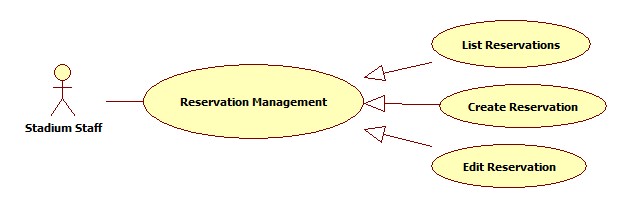
###### Active Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-022 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_022 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Active Field | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Owner  **Summary:**  This use case describes how to active a field of stadium  **Goal:**  Active a field  **Triggers:**  Actor check the checkbox  **Preconditions:**  Login as Stadium Owner role  **Post-conditions:**  Activated message is displayed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Check the checkbox | Show message to request user’s confirmation | | 2 | Click OK button | Active this field |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Discard recent action and close message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### De-active Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-023 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_023 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Deactive Field | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Owner  **Summary:**  This use case describes how to deactive a field of stadium  **Goal:**  Deactive a field  **Triggers:**  Actor uncheck the checkbox  **Preconditions:**  Login as Stadium Owner role  **Post-conditions:**  Deactivated message is displayed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Uncheck the checkbox | Show message to request user’s confirmation | | 2 | Click OK button | Deactive this field |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Discard recent action and close message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Reservations Management



###### List Reservations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-024 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_024 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Reservations | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Staff  **Summary:**  This use case describes listing reservations.  **Goal:**  List all reservations.  **Triggers:**  Actor go to Reservation Management page  **Preconditions:**  Login as Stadium Staff role  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | The system shows list of reservation. |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Create Reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-025 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_025 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Reservation | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Staff  **Summary:**  This use case describes creating a reservation.  **Goal:**  Create a reservation.  **Triggers:**  Click Create button  **Preconditions:**  Login as Stadium Staff role  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Create button | Display Book form | | 2 | Enter information of booking  Click on Check available button | Show Field is available message | | 3 | Click OK button | Show successful message and save this reservation |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Check available button | Show Field is not available message | | Click on Cancel button | Back to previous page |   **Exceptions:**  This field is not available such as existing reservation, duplicate booking time, etc.  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Edit Reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-026 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_026 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Reservation | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Staff  **Summary:**  This use case describes editing a reservation.  **Goal:**  Edit reservation’s information.  **Triggers:**  Click the Reservation which will be edited  **Preconditions:**  Login as Stadium Staff role  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Reservation which will be edited | Display this Reservation’s information | | 2 | Enter new reservation information Click Check Available button | Show Field is available message | | 3 | Click Save button | Show successful message and save this reservation |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Check available button | Show Field is not available message | | Click on Cancel button | Show confirm cancelation message  Discard all recent filled information and back to previous page |   **Exceptions:**  This field is not available such as existing reservation, duplicate booking time, etc.  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Stadium promotion management



###### List Promotions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-027 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_027 | **Use-case Version** | | 1.0 |
| **Use-case Name** | List Promotions | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Staff  **Summary:**  This use case describe listing Promotions  **Goal:**  List all Promotions of stadium  **Triggers:**  Actor go to Promotion Management page  **Preconditions:**  Login as Stadium Staff role  **Post-conditions:**    **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | The system shows list of promotion. |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

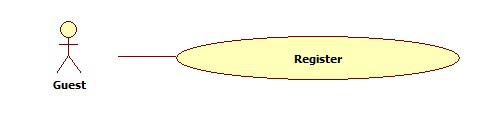
###### Create Promotion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-028 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_028 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Promotion | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Staff  **Summary:**  This use case describes creating a Promotion.  **Goal:**  Create a Promotion for stadium.  **Triggers:**  Click Create button  **Preconditions:**  Login as Stadium Staff role  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Create button | Display Promotion form | | 2 | Enter information of Promotion Click OK button | Show successful message and save this Promotion |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Show cancelation confirm message and back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

###### Edit Promotion

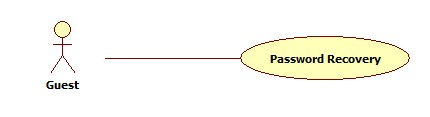
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-029 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_029 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Promotion | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Stadium Staff  **Summary:**  This use case describes editing a promotion.  **Goal:**  Edit reservation’s information.  **Triggers:**  Click the promotion which will be edited  **Preconditions:**  Login as Stadium Staff role  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Promotion which will be edited | Display this Reservation’s information | | 2 | Enter new Promotion information Click Save button | Show successful message and save this reservation |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Show confirm cancelation message  Discard all recent filled information and back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Register

****

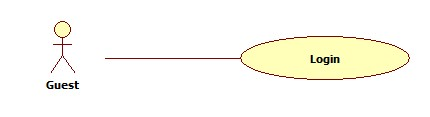
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-030 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_030 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Register | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Guest  **Summary:**  Guest uses this use case to register account  **Goal:**  Allow guest to register account  **Triggers:**  On header page, click on Register link and Register page will be showed, then input information and then click Register button to finish.  **Preconditions:**  N/A  **Post-conditions:**  Guest can register an account successfully. System will transfer to Homepage  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on register link | System will response Register page | | 2 | Enter information into text fields. Click on Register button to finish | System will create new account in database, user will be login automatically and system will transfer to Homepage of User |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to “Homepage |   **Exceptions:**     |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | | 1. | Email has existed in database. | Display error message: “Email has existed. Try another email.” | | 2. | Input not a valid information in Email’s textbox. | Display error message: “Email is invalid”. | | 3. | Password’s length is not in range: [6,20]. | Display error message: “Passwords must be at least 6 characters ”. | | 4. | Input confirm password information different with password. | Display error message: “Confirm password wrong . Please re-enter”. |   **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Password Recovery



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-031 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_031 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Password Recovery | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Guest  **Summary:**  Guest uses this use case to recovery account’s password  **Goal:**  Recovery account’s password  **Triggers:**  Actor go to recovery password page  **Preconditions:**  N/A  **Post-conditions:**  A message about password recovery result will be displayed to user  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on recovery password link | System will response Recovery Password page | | 2 | Enter required information  Click Recovery button | Display successful message |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to “Homepage |   **Exceptions:**     |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | |  | Invalid Inputted Information | Display error message |   **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Login



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-032 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_032 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Login | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Guest  **Summary:**  Guest uses this use case to login  **Goal:**  Login to system  **Triggers:**  Actor use login form or go to login page  **Preconditions:**  N/A  **Post-conditions:**  User logged into system and redirect to Homepage or a specific page  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter account information  Click Login button | Redirect user to homepage or specific page | |  |  |  |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button to cancel | System will transfer to “Homepage |   **Exceptions:**     |  |  |  | | --- | --- | --- | |  | Actor Action | System Response | |  | Invalid Inputted Information | Display error message |   **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### View Stadium’s Detail:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-033 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_033 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Stadium’s Detail | | | |
| **Author** | KhánhNQ | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Guest  **Summary:**  This use case describe how to view detail of stadium  **Goal:**  See detail of stadium  **Triggers:**  Actor use View Stadium’s Detail when click on Detail link.  **Preconditions:**  N/A  **Post-conditions:**  Display detail of stadium  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click Detail button | Display detail of stadium |   **Alternative Scenario:**  N/A  **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Find Available Stadiums



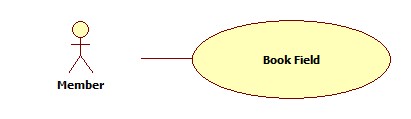
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-034 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_034 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Find Available Stadiums | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to find available stadium with conditions  **Goal:**  Find available stadium  **Triggers:**  Actor click on Find button  **Preconditions:**  N/A  **Post-conditions:**    **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter conditions information  Click on Find button | Show list of stadiums |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Find Stadium



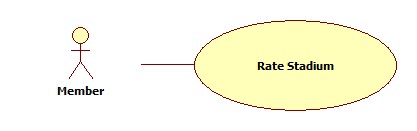
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-035 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_035 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Find Stadium | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to find stadium  **Goal:**  Find stadium  **Triggers:**  Actor click on Find button  **Preconditions:**  N/A  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter conditions information  Click on Find button | Show list of stadiums |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Book Field



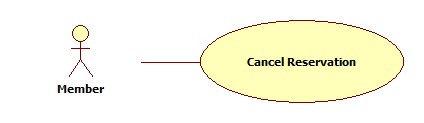
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-036 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_036 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Book Field | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to book field  **Goal:**  Book a field of stadium  **Triggers:**  Actor click on Book button  **Preconditions:**  Login as user  **Post-conditions:**  Successful message is displayed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Book button | Display Book form | | 2 | Enter information of booking  Click on Check available button | Show Field is available message | | 3 | Click OK button | Show successful message and save this reservation |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Check available button | Show Field is not available message | | Click on Cancel button | Back to previous page |   **Exceptions:**  This field is not available such as existing reservation, duplicate booking time, etc.  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Rate Stadium



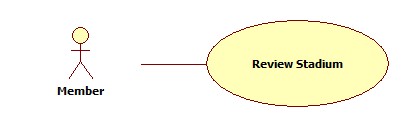
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-037 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_037 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Rate Stadium | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to rate stadium  **Goal:**  Rate a stadium  **Triggers:**  Actor click on Rate stars of stadium  **Preconditions:**  Login as user  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Rate stars | Save this rate |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Cancel Reservation



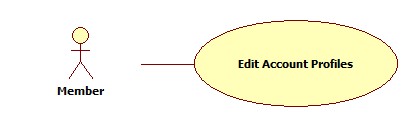
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-038 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_038 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Cancel Reservation | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to cancel reservation  **Goal:**  Cancel a reservation  **Triggers:**  Actor click on Cancel button  **Preconditions:**  Login as user. Reservation is not expired.  **Post-conditions:**  Successful message is displayed  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Cancel button | Show message to request user’s confirmation | | 2 | Click OK button to confirm this action | Show successful message and cancel reservation |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Close the message |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |
|  | | | | |

##### Review Stadium



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-039 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_039 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Review Stadium | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to review stadium  **Goal:**  Review a stadium  **Triggers:**  Actor write a review in review message box in stadium details page  **Preconditions:**  Login as user  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Write a review stadium | Save this review |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  Blank content  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Edit Account Profiles



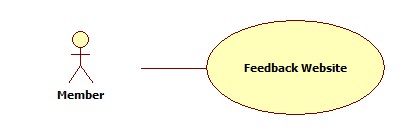
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-040 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_040 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Account Profiles | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to edit user’s account profiles  **Goal:**  Update user’s account profiles  **Triggers:**  Actor click on Edit button  **Preconditions:**  Login as user  **Post-conditions:**  Successful message is displayed  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Edit button | Display Edit form with filled information | | 2 | Enter new profiles information of user’s account  Click Save button | Update new profiles information |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Find Rival



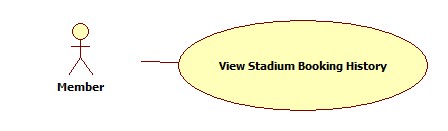
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-041 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_041 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Find Rival | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to find rival  **Goal:**  Find rival  **Triggers:**  Actor click on Find Rival button  **Preconditions:**  Login as user  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter conditions information  Click on Find Rival button | Show list of rivals |   **Alternative Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Feedback Website



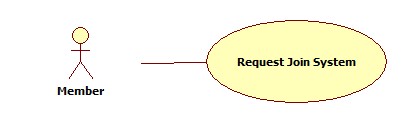
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-042 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_042 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Feedback Website | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to feedback website  **Goal:**  Feedback website  **Triggers:**  Actor click on Feedback button  **Preconditions:**  Login as user  **Post-conditions:**  Successful message is displayed  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Feedback button | Display Feedback form | | 2 | Enter feedback information  Click on Send button | Send an email to webmaster. |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Respond to home page |   **Exceptions:**  Required Fields are blank  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### View Stadium Booking History



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-043 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_043 | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Stadium Booking History | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to view stadium booking history  **Goal:**  View stadium booking history  **Triggers:**  Actor click on History button  **Preconditions:**  Login as user  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on History button | Show list of stadium booking histories |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | |  |  |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

##### Request Join System



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-044 SPECIFICATION** | | | | |
| **Use-case No.** | UC\_044 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Request Join System | | | |
| **Author** | ThinhND | | | |
| **Date** | 12/02/2014 | **Priority** | High | |
| **Actor:**  Member  **Summary:**  This use case describes how to request join system  **Goal:**  Request join system  **Triggers:**  Actor click on Join System button  **Preconditions:**  Login as user  **Post-conditions:**  N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Join System button | Display Stadium Create form | | 2 | Enter information of stadium  Click on Join button | Show message request success and save stadium information |   **Alternative Scenario:**     |  |  | | --- | --- | | Actor Action | System Response | | Click on Cancel button | Back to previous page |   **Exceptions:**  N/A  **Relationships:**  N/A  **Business Rules:**  N/A | | | | |

#### Non-Functional Requirements

##### Reliability

* Mean Time Between Failures (MTBF): The acceptable failure is once a year
* Mean Time To Repair (MTTR): The acceptable mean time for a particular failure must be least than 12 hours
* Maximum Bugs or Defect Rate: 05 bugs / KLOC

##### Availability

* The system always responds to every user who got permission. The server shall be working 24/7.

##### Security

* Customer information must be secured
* The system always check authenticate and permission for every function

##### Maintainability

N/A

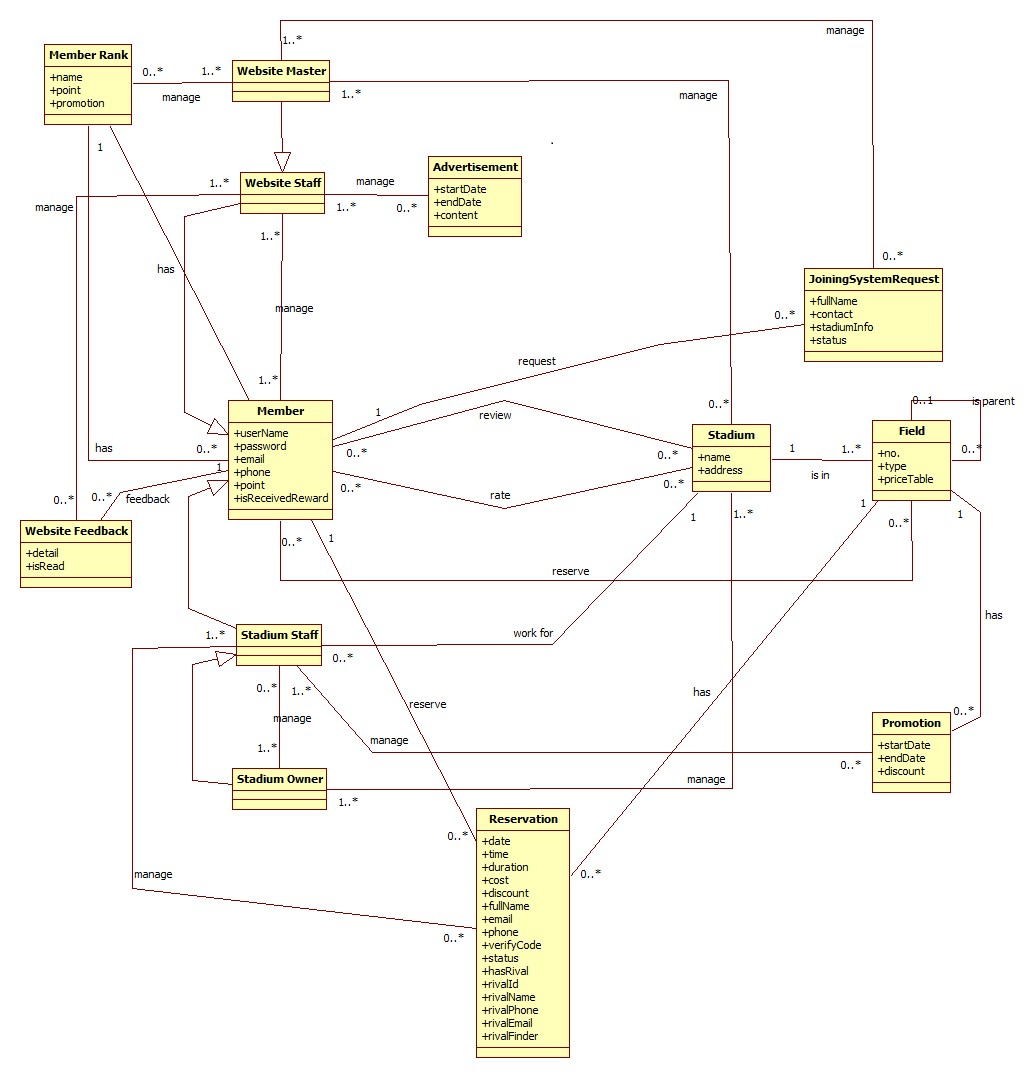
##### Portability

N/A

##### Performance

N/A

### Entity Relationship Diagram



## System Design Description (SDD)

### Design Overview

* This document describes the technical and UI design of the FPB system. It includes the architectural design and the detailed design of common functions and business functions. It also includes the design of database model.
* The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem. It will describe the patterns being used, the role of each component and the role of the system in the working environment.
* The detailed design describes static and dynamic structure for each component and function. It includes class diagrams, class explanations, and sequence diagrams of the main use cases. The detailed design uses notations of .NET framework 4.0 and C#, as they are the framework and language for developing this system.
* The database design describes the relationship between entities, and details of each entity. It uses notations of SQL Server 2008 R2 as it is the database server for developing the system.
* The user interface design describes the layout of the system, and some design for the screens

### System Architectural Design

#### Choice of System Architecture

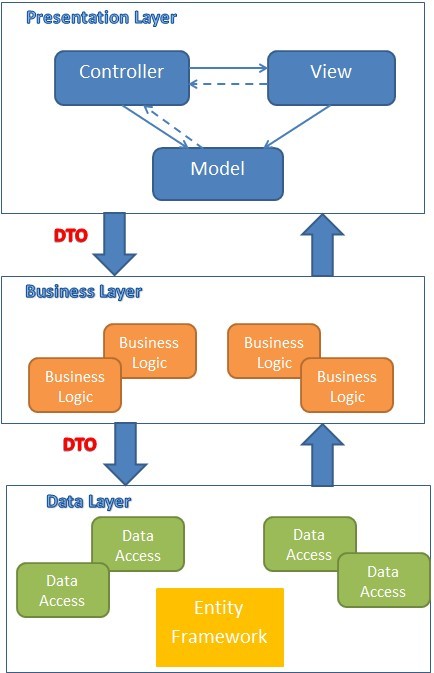
****

Figure 2: MVC implemented in a three-layer architecture

The above diagram is explained as the following:

**Model**: Used to prepare data for view

**View**: The view displays the data and also takes input form user. It renders the model data into a form, such as graphics, buttons, and tables, and so on to display to the user.

**Controller:** The controller handles all requests coming from the view or user interface. The data flows to whole application are controlled by controller.

**Business Logic:** This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

**Data Access:** Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

#### Discussion of Alternative Designs

N/A

#### Description of System Interface

N/A

### Component Diagram

### 

### Detailed Description of Components



Figure 3: General Class Diagram

#### Class Diagram Explanation

##### Role

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of Role |
| Role | String | Role |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Account | Initialize a role with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Role | Integer | Create a role and save it to database |
| Update | * Role | Integer | Update information of an existing role in database |

##### Account

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of an user in system |
| Username | String | Name of account, it is unique in system |
| Password | String | Password of account |
| FullName | String | Full name of user |
| Address | String | Address of user |
| PhoneNumber | String | Phone number of user |
| Email | String | Email of user |
| Point | Float | Point of user, it is used to calculate rank of account |
| RankId | Integer | Rank of account |
| IsReceivedReward | Boolean | Every time the rank grow up, account may receive a reward, this field is used to tracking whether account received reward or not |
| JoinDate | Date | The date when user join this system |
| IsActive | Boolean | Is this account active |
| Role | Integer | Id of role |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Account | Initialize an account with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Account | Integer | Create an account and save it to database |
| Update | * Password * FullName * Address * PhoneNumber * Email * Point * RankId * IsReceivedReward * Role | Integer | Update information of an existing account in database |
| Deactivate | N/A | Integer | De-active an account |
| Activate | N/A | Integer | Activate an account |
| GetById | * Id | Account | Get an account instance by Id of account from database |
| GetByUserName | * UserName | Account | Get an account instance by UserName of account from database |
| GetByEmail | * Email | Account | Get an account instance by Email of account from database |
| GetAll | N/A | List<Account> | Get all account from database |

##### Stadium

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of stadium |
| Name | String | Name of stadium |
| Street | String | Name of street that stadium located |
| Ward | String | Name of ward that stadium located |
| District | String | Name of district that stadium located |
| City | String | Name of city or province that stadium located |
| Description | String | Description of stadium |
| PhoneNumber | String | Phone number of stadium |
| Email | String | Email of stadium |
| IsActive | Boolean | Is the stadium active |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Stadium | Initialize an Stadium with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Stadium | Integer | Create a stadium and save it to database |
| Update | * Name * Street * Ward * District * City * Description * PhoneNumber * Email | Integer | Update information of an existing stadium in database |
| Deactivate | N/A | Integer | De-active a stadium |
| Activate | N/A | Integer | Activate a stadium |
| GetById | * Id | Stadium | Get an stadium instance by Id of stadium from database |
| GetAll | N/A | List<Stadium> | Get all account from database |
| FindStadium | * Name * Address * FieldType | List<Stadium> | Get all stadiums correspond with filter |
| FindAvailableStadium | * Address * FieldType * Date * StartTime * Duration | List<Stadium> | Get all stadiums those are available with request date, time, location |

##### Field

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of field |
| StadiumId | Integer | Id of stadium that contain this field |
| Number | Integer | Number of field in stadium |
| ParentField | Integer | Id of another field that contain this field |
| FieldType | Integer | Type of field |
| IsActive | Boolean | Is this field active |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Field | Initialize an Field with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Field | Integer | Create a field and save it to database |
| Update | * Name * Street * Ward * District * City * Description * PhoneNumber * Email | Integer | Update information of an existing field in database |
| Deactivate | N/A | Integer | De-active a field |
| Activate | N/A | Integer | Activate a field |
| GetById | * Id | Field | Get a field instance by Id of field from database |
| GetByStadiumId | * StadiumId | List<Field> | Get all field of the stadium that has corresponding id |
| GetAll | N/A | List<Field> | Get all field from database |

##### FieldPrice

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of field price |
| FieldId | Integer | Id of field that has this price |
| TimeFrom | Float | Start time in day that use this price |
| TimeTo | Float | End time in day that use this price |
| Price | Float | Price |
| Day | Integer | Day in week that use this price  Start with Sunday is 0, Monday is 1, … |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | FieldPrice | Initialize an FieldPrice with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class FieldPrice | Integer | Create a price of field and save it to database |
| Update | * FieldId * TimeFrom * TimeTo * Price * Day | Integer | Update information of an existing field price in database |
| Delete | * Id | Integer | Delete a field price |
| GetById | * Id | FieldPrice | Get a field price instance by Id of field pricefrom database |
| GetByDayTime | * Day * StartTime | FieldPrice | Get Field Price for Day, StartTime |
| GetAll | N/A | List<FieldPrice> | Get all field price from database |
| GetByFieldId | * FieldId | List<FieldPrice> | Get All FieldPrice of a field |

##### Promotion

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of promotion |
| FieldId | Integer | Id of field that use this promotion |
| PromotionFrom | Date | Start date that use this promotion |
| PromotionTo | Date | Expired date of this promotion |
| Discount | Float | Discount amount of this promotion |
| Creator | Integer | Id of staff that create this promotion |
| IsActive | Boolean | Is this promotion active |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Promotion | Initialize an Promotion with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Promotion | Integer | Create a promotion and save it to database |
| Update | * FieldId * PromotionFrom * PromotionTo * Discount * Creator | Integer | Update information of an existing promotion in database |
| Delete | * Id | Integer | Delete a field price |
| GetById | * Id | Promotion | Get a Promotion instance by Id from database |
| GetByDate | * Date | List< Promotion > | Get all Promotion for date |
| GetAll | N/A | List< Promotion > | Get all Promotion from database |
| GetByFieldId | * FieldId | List< Promotion > | Get all Promotion of a field |
| GetByStadiumId | * StadiumId | List<Promotion> | Get all Promotion of a stadium |

##### StadiumReview

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of review |
| UserId | Integer | Id of user create this review |
| StadiumId | Integer | Id of stadium being reviewed |
| ReviewContent | Float | Content of this review |
| IsApproved | Float | Is this review approved |
| Approver | Integer | Id of stadium staff approve this review |
| CreateDate | DateTime | The date that this review is created |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | StadiumReview | Initialize an StadiumReview with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class StadiumReview | Integer | Create a review of stadium and save it to database |
| Update | * ReviewContent * IsApproved * TimeTo | Integer | Update information of an existing review in database |
| Delete | * Id | Integer | Delete a Review |
| GetById | * Id | StadiumReview | Get a review by Id of StadiumReview from database |
| GetByUserId | * UserId | List<StadiumReview> | Get all review of an user |
| GetAll | N/A | List<StadiumReview> | Get all review from database |
| GetByStadiumId | * StadiumId | List<StadiumReview> | Get all review of a stadium |

##### StadiumRating

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of review |
| UserId | Integer | Id of user create this rate |
| StadiumId | Integer | Id of stadium being rated |
| Rate | Float | Rate |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | StadiumRating | Initialize an StadiumRating with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class StadiumRating | Integer | Create a rate of stadium and save it to database |
| Update | * Rate | Integer | Update rate of an existing rate in database |
| Delete | * Id | Integer | Delete a Rate |
| GetById | * Id | StadiumRating | Get a review by Id of StadiumRating from database |
| GetByStadiumId | * StadiumId | List<StadiumRating> | Get all rate of a stadium |

##### Reservation

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of Reservation |
| FieldId | Integer | Id of field that user book |
| UserId | Integer | Id of user create reservation |
| FullName | String | Name of person who book the field (booker) |
| PhoneNumber | String | Phone number of booker |
| Email | String | Email of booker |
| Date | Date | Date which booker want to reverse |
| StartTime | Float | Start time which booker want to start using |
| Duration | Float | How long booker want to using |
| Price | Float | Price of this reservation |
| VerifyCode | String | Verify code of this reservation, booker use this to trade with stadium manager for their approval |
| CreateDate | DateTime | The time this reservation is created |
| Approver | Integer | Id of stadium manager who approve this reservation |
| Status | String | Status of reservation |
| HasRival | Boolean | False if booker want to get a help for finding a rival  True if booker has a rival |
| RivalId | Integer | Id of an user who join as rival of booker (rival) |
| RivalName | String | Name of rival |
| RivalPhone | String | Phone number of rival |
| RivalEmail | String | Email of rival |
| RivalFinder | Integer | Id of user who found rival for this reservation  It can be Id of booker or stadium manager |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Reservation | Initialize an Reservation with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Reservation | Integer | Create a reservation and save it to database |
| Update | * FieldId * FullName * PhoneNumber * Email * Date * StartTime * Duration * Price * CreateDate * Approver * Status * HasRival * RivalId * RivalName * RivalPhone * RivalEmail * RivalFinder | Integer | Update reservation information of an existing reservation in database |
| Cancel | * Id | Integer | Cancel a Reservation |
| GetById | * Id | Reservation | Get a reservation by Id from database |
| GetByUserId | * UserId | List<Reservation> | Get all reservation of an user |
| GetByFieldId | * FieldId | List<Reservation> | Get all reservation of a field |
| GetByStadiumId | * StadiumId | List<Reservation> | Get all reservation of a stadium |

##### JoinSystemRequest

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of review |
| UserId | Integer | Id of user submit request (requester) |
| FullName | String | Full name of requester |
| Address | String | Address of requester |
| Phone | String | Phone of requester |
| Email | String | Email of requester |
| StadiumName | String | Name of stadium owned by requester |
| StadiumAddress | String | Address of stadium |
| Note | String | Note of requester |
| Status | String | Status of request |
| CreateDate | DateTime | Create date of request |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | RequestJoinSystem | Initialize an RequestJoinSystem with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class RequestJoinSystem | Integer | Create a request and save it to database |
| Update | * FullName * Address * Phone * Email * StadiumName * StadiumAddress * Note * Status | Integer | Update request information of an existing request in database |
| Delete | * Id | Integer | Delete a RequestJoinSystem |
| GetById | * Id | RequestJoinSystem | Get a request by Id from database |
| GetByUserId | * UserId | List<RequestJoinSystem> | Get all request of an user |
| GetAll | N/A | List<RequestJoinSystem> | Get all request |

##### Member Rank

* **Atributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of rank |
| RankName | String | Name of rank |
| RankPoint | Integer | Minimum point to get rank |
| Promotion | String | Promotion of rank |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | MemberRank | Initialize an MemberRank with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class MemberRank | Integer | Create a MemberRank and save it to database |
| Update | * RankName * RankPoint * Promotion | Integer | Update MemberRank information of an existing MemberRank in database |
| Delete | * Id | Integer | Delete a MemberRank |
| GetByPoint | * Id | MemberRank | Get a MemberRank by point |
| GetById | * UserId | MemberRank | Get a MemberRank by id |
| GetAll | N/A | List<MemberRank> | Get all MemberRank |

##### Advertisement

* **Atributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | Integer | Id of advertisement |
| Position | String | Position of advertisement |
| AdvertiseDetail | String | Content of advertisement |
| CreateDate | DateTime | Create date of advertisement |
| ExpiredDate | DateTime | Expired date of advertisement |
| Status | String | Status of advertisement |
| Creator | Integer | Id of creator |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | Advertisement | Initialize an Advertisement with value (optional) |
| Getters | N/A | Type of attribute | Get value of an attribute corresponding with getter of attribute |
| Setters | Value of attribute | Void | Set value for an attribute corresponding with setter of attribute |
| Create | All of attribute of class Advertisement | Integer | Create an Advertisement and save it to database |
| Update | * Position * AdvertiseDetail * ExpiredDate * Status | Integer | Update Advertisement information of an existing Advertisement in database |
| Delete | * Id | Integer | Delete an Advertisement |
| GetByPosition | * Position | Advertisement | Get an Advertisement by position |
| GetById | * Id | Advertisement | Get an Advertisement by id |
| GetAll | N/A | List<Advertisement> | Get all Advertisement |

##### Member

Inherit all attributes and methods of class Account

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Authenticate | * UserName * Password | Boolean | Authenticate a user with system |
| FindStadium | * StadiumName * Location * FieldType | List<Stadium> | Find stadium with some option, let a option null to ignore that option when search |
| ViewStadiumDetail | * StadiumId | Stadium | View detail of a stadium with member view |
| FindAvailableStadium | * Location * FieldType * Date * StartTime * Duration | List<Stadium> | Find available stadium with options |
| FindRival | * Location * FieldType * Date * StartTime * Duration | List<Reservation> | Find list reservation that do not have rival |
| JoinMatch | * ReservationId | Void | Join a match as rival of booker |
| BookField | * FieldId * FullName * Address * Phone * Email * Date * StartTime * Duration * HasRival | Boolean | Book a field |
| CancelReservation | * ReservationId | Void | Cancel a reservation |
| ViewBookingHistory | * UserId | List<Reservation> | View history of booking with Member view |
| ReviewStadium | * StadiumId * UserId * ReviewContent | Void | Review a stadium |
| RateStadium | * StadiumId * UserId * Rate | Void | Rate a stadium |
| UpdateAccountProfile | * UserId * Password * FullName * Address * PhoneNumber * Email | Void | Update account information |
| RequestJoinSystem | * UserId * FullName * Address * PhoneNumber * Email * StadiumName * StadiumAddress * Note | Void | Request for joining system of owned stadium |
| FeedbackWebsite | * Content * FullName * Email | Void | Send feedback content to email |

##### WebsiteStaff

Inherit all attributes and methods of class Member

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| CreateAdvertising | * Position * AdvertisementDetail * ExpiredDate * Status | Void | Create an advertisement |
| UpdateAdvertising | * AdvertisementId * Position * AdvertisementDetail * ExpiredDate * Status | Void | Update an advertisement |
| DeleteAdvertising | * AdvertisementId | Void | Delete an advertisement |
| ViewAdvertisings | N/A | List<Advertisement> | View all Advertisements with website staff view |
| UpdateUser | * UserId * Password * Email * Point * IsReceivedReward * IsActive * Role | Void | Update information of an user with website staff permission |
| ActivateUser | * UserId | Void | Activate an user |
| DeactivateUser | * UserId | Void | Deactivate an user |
| ViewUsers | N/A | List<User> | View all users with website staff view |

##### WebsiteMaster

Inherit all attributes and methods of class WebsiteStaff

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| CreateStadium | * Name * Street * Ward * District * City * Description * PhoneNumber * Email * IsActive | Void | Create a stadium |
| UpdateStadium | * StadiumId * Name * Street * Ward * District * City * Description * PhoneNumber * Email * IsActive | Void | Update a stadium |
| ActivateStadium | * StadiumId | Void | Delete an advertisement |
| DeactivateStadium | * StadiumId |  |  |
| ViewStadiums | N/A | List<Stadium> | View all Stadium with website master view |
| CreateMemberRank | * RankName * RankPoint * Promotion | Void | Create a member rank |
| UpdateMemberRank | * MemberRankId * RankName * RankPoint * Promotion | Void | Update a member rank |
| DeleteMemberRank | * MemberRankId | Void | Delete a member rank |
| ViewRanks | N/A | List<MemberRank> | View all member rank with website master view |

##### StadiumStaff

Inherit all attributes and methods of class Member

* **Attributes**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| StadiumIds | List<Integer> | List ids of stadium those staff work for |

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| Constructors | Optional | StadiumStaff | Initialize a StadiumStaff with value (optional) |
| GetStadiumIds | N/A | List<Integer> | Get list ids of stadium those staff work for |
| SetStadiumIds | * StadiumIds | Void | Set list ids of stadium those staff work for |
| ReverseReservation | * FieldId * FullName * PhoneNumber * Email * Date * StartTime * Duration * HasRival | Void | Create a reservation for a field that staff manage |
| UpdateReservation | * FieldId * FullName * PhoneNumber * Email * Date * StartTime * Duration * HasRival * RivalName * RivalPhone * RivalEmail | Void | Update a reservation that staff manage |
| ApproveReservation | * ReservationId | Void | Approve a reservation that staff manage |
| DenyReservation | * ReservationId | Void | Deny a reservation that staff manage |
| ViewReservations | N/A | List<Reservation> | View list reservation that staff manage |
| CreatePromotion | * FieldId * PromotionFrom * PromotionTo * Discount * IsActive | Void | Create a promotion for a field that staff manage |
| UpdatePromotion | * ReservationId * FieldId * PromotionFrom * PromotionTo * Discount * IsActive | Void | Update a promotion for a field that staff manage |
| ActivatePromotion | * ReservationId | Void | Activate a promotion that staff manage |
| DeactivatePromotion | * ReservationId | Void | Deactivate a promotion that staff manage |
| ViewPromotion | N/A | List<Promotion> | View list promotion that staff manage |

##### StadiumOwner

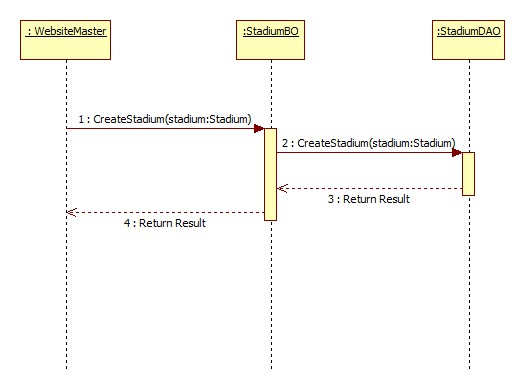
Inherit all attributes and methods of class StadiumStaff

* **Methods**

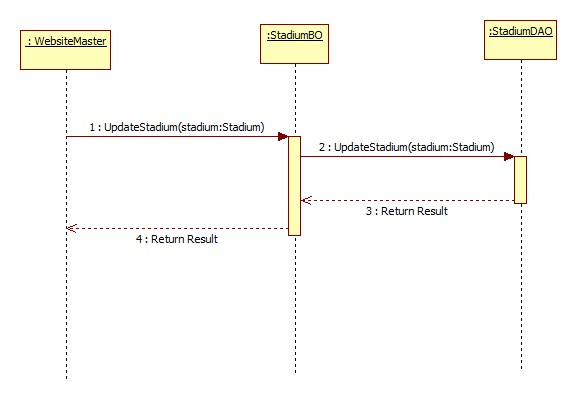
|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameter** | **Return Type** | **Description** |
| UpdateStadiumProfile | * StadiumId * Name * Street * Ward * District * City * Description * PhoneNumber * Email | Void | Update a stadium of stadium owner |
| AddFieldToStadium | * StadiumId * Number * ParentField * FieldType * IsActive | Void | Add a field to an owned stadium |
| UpdateFieldInformation | * FieldId * Number * ParentField * FieldType * IsActive | Void | Update an owned field |
| ActivateField | * FieldId | Void | Activate an owned field |
| DeactivateField | * FieldId | Void | Deactivate an owned field |
| ViewFields | * StadiumId | List<Field> | View list owned fields |

### Sequence Diagram

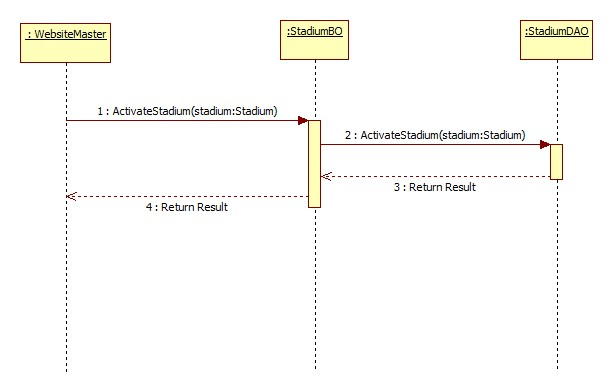
#### Create Stadium



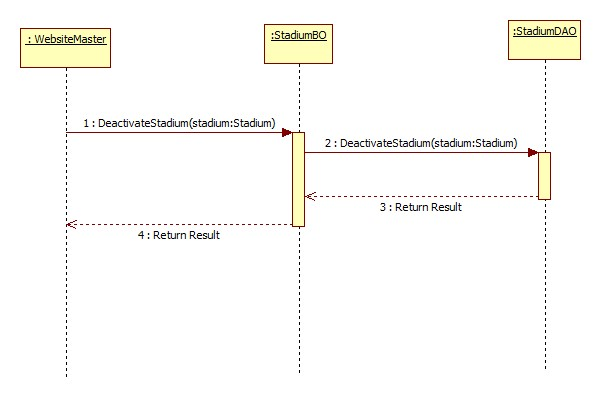
#### Update Stadium



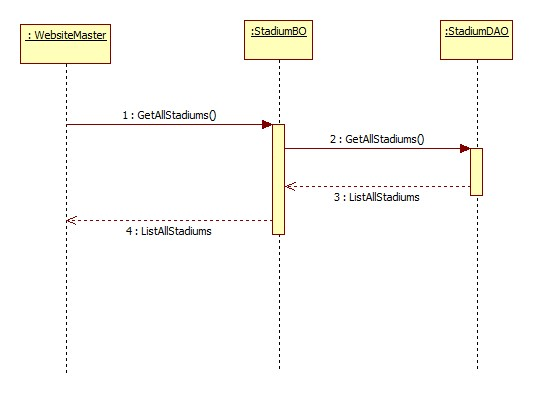
#### Activate Stadium



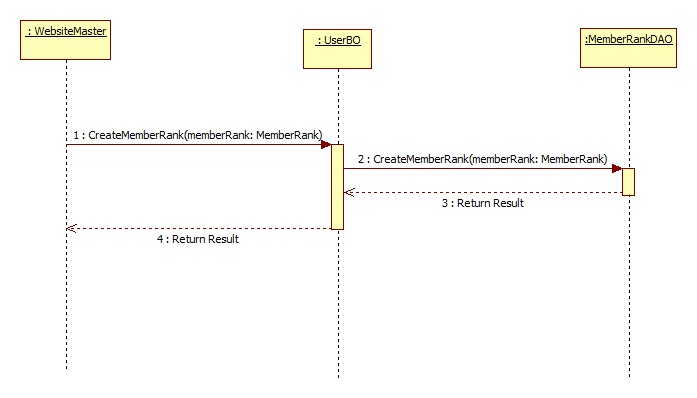
#### Deactivate Stadium



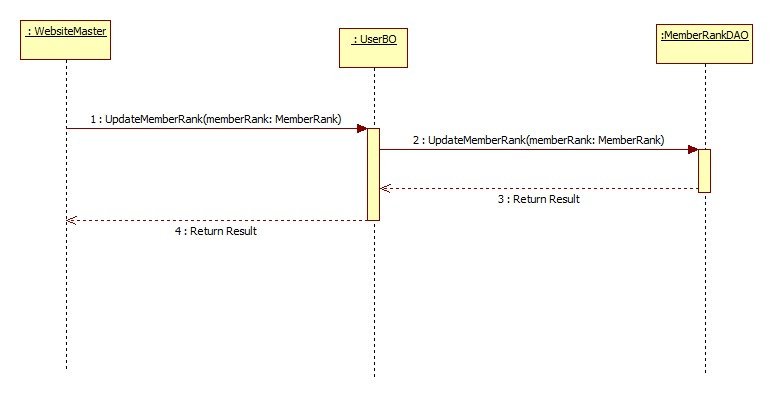
#### View Stadiums



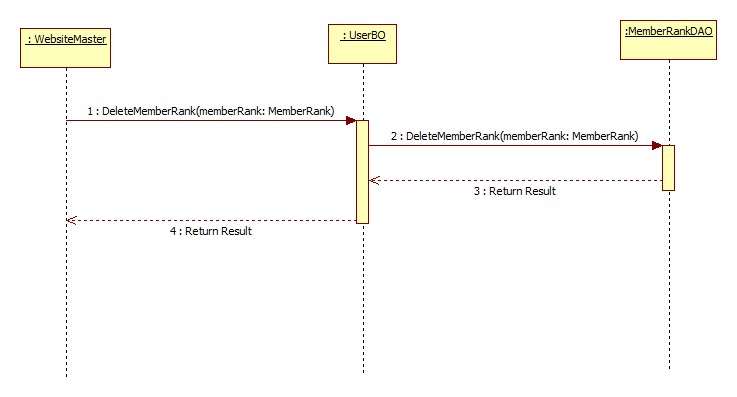
#### Create Member Rank



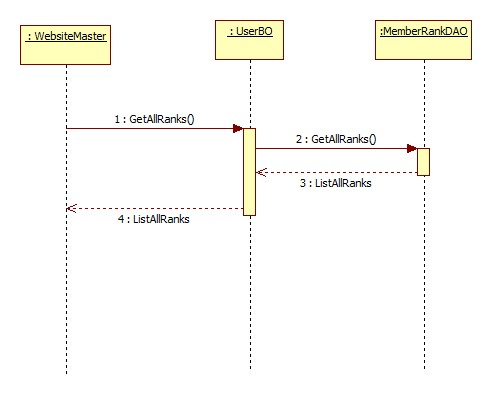
#### Update Member Rank



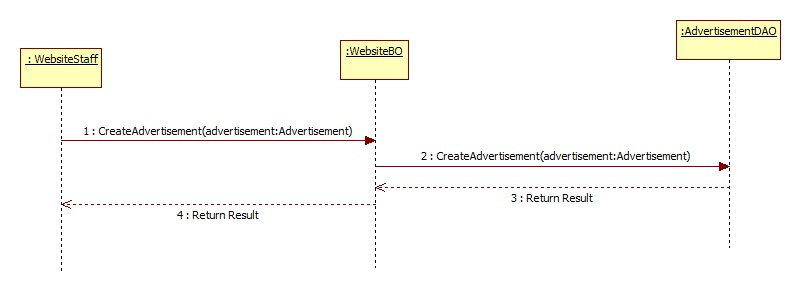
#### Delete Member Rank



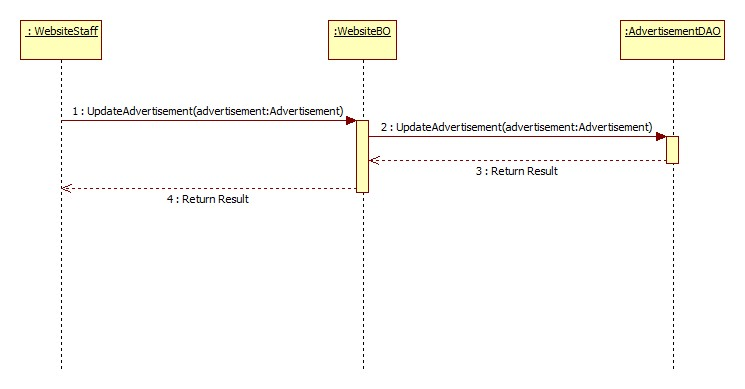
#### View Member Ranks



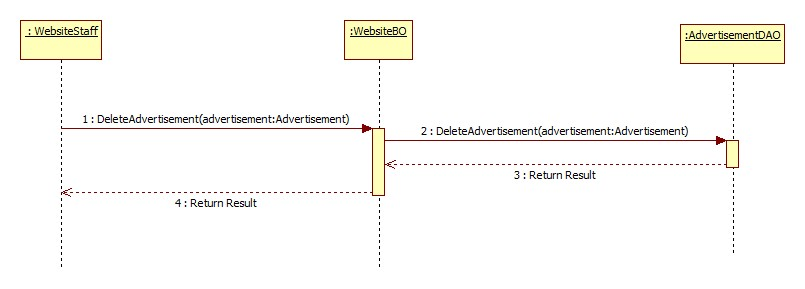
#### Create Advertising



#### Update Advertising



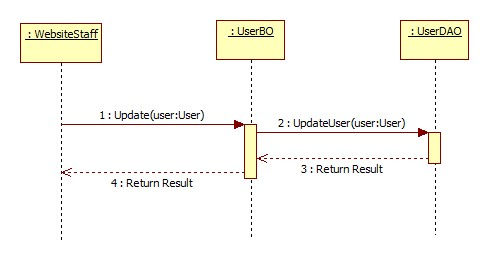
#### Delete Advertising



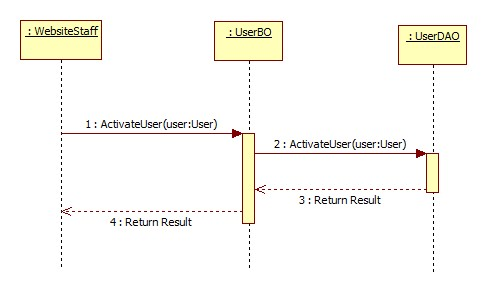
#### View Advertisings



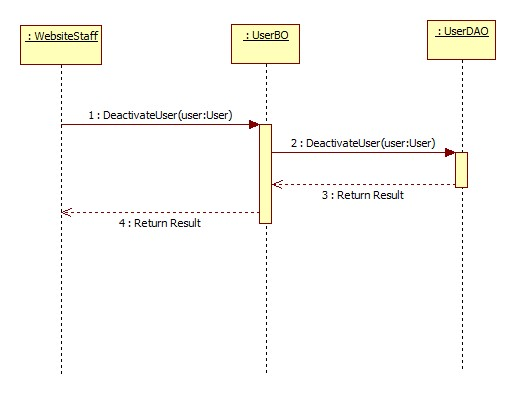
#### Update User



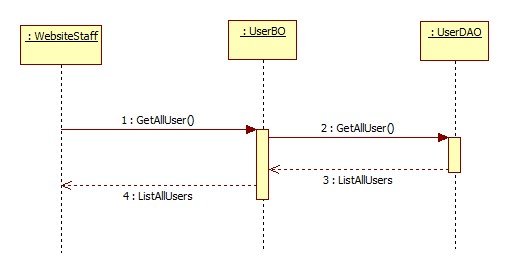
#### Activate User



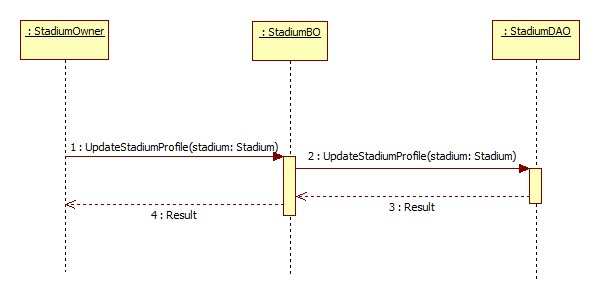
#### Deactivate User



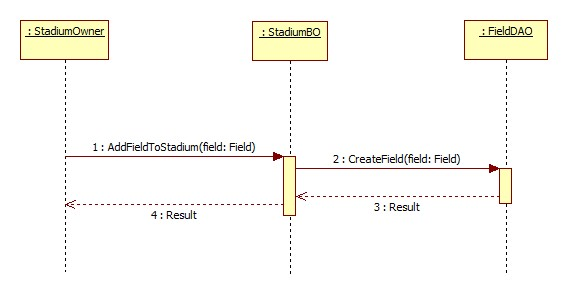
#### View Users



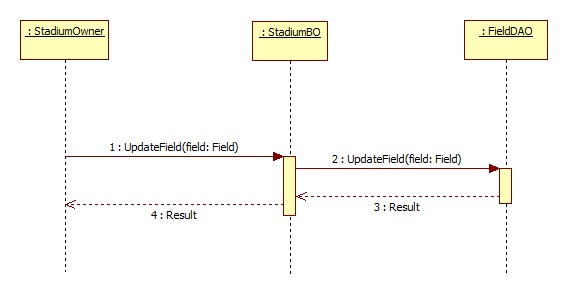
#### Update Stadium Profile



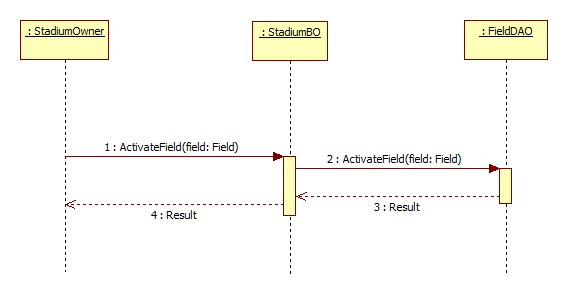
#### Add Field To Stadium



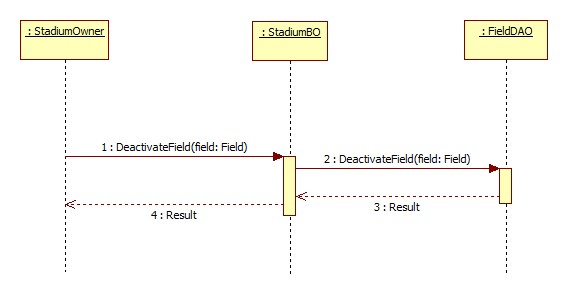
#### Update Field



#### Activate Field



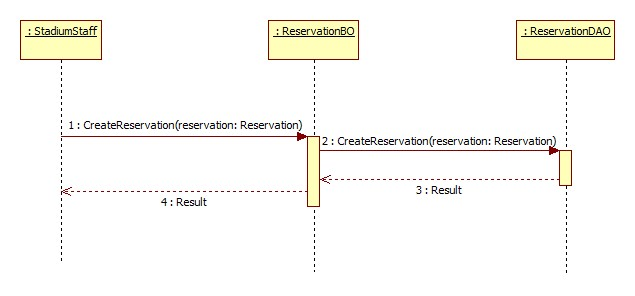
#### Deactivate Field



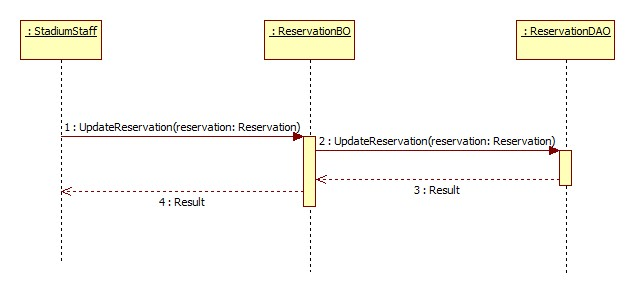
#### View Fields



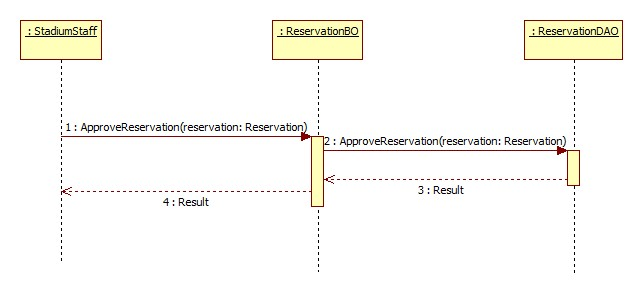
#### Create Reservation



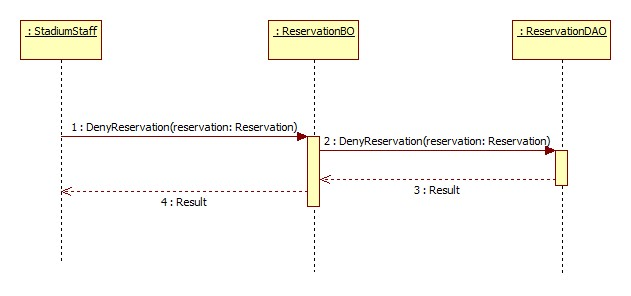
#### Update Reservation



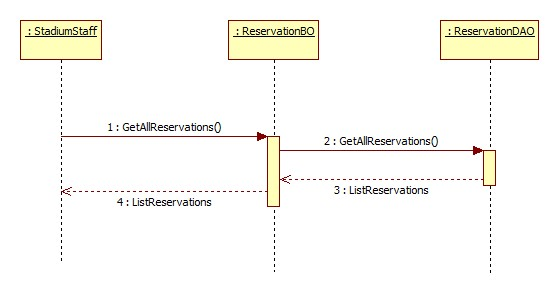
#### Approve Reservation



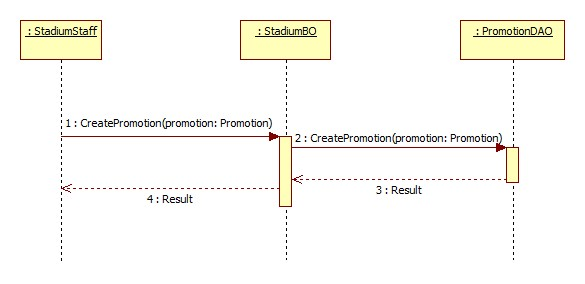
#### Deny Reservation



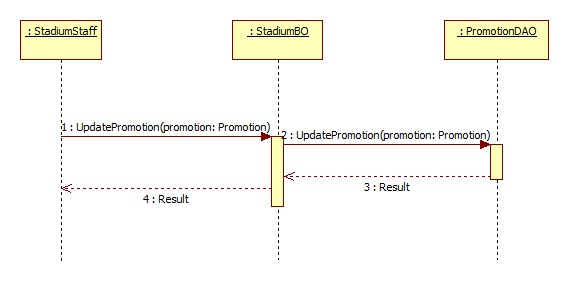
#### View Reservations



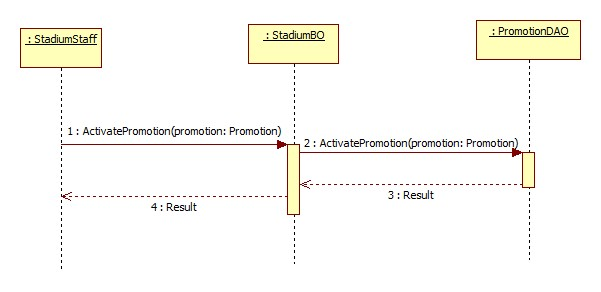
#### Create Promotion



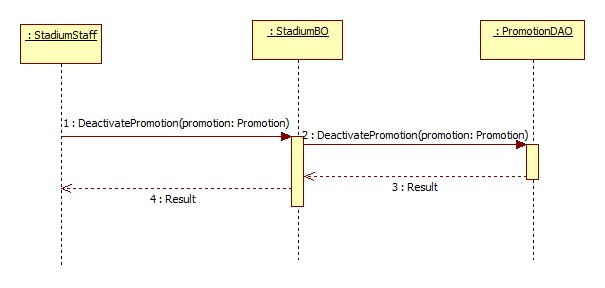
#### Update Promotion



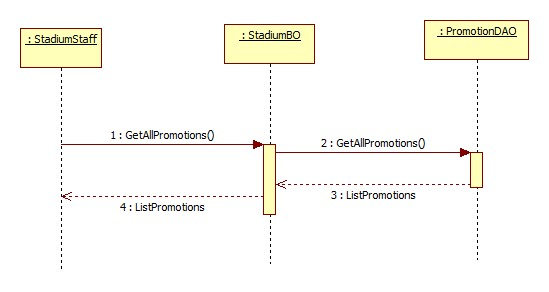
#### Activate Promotion



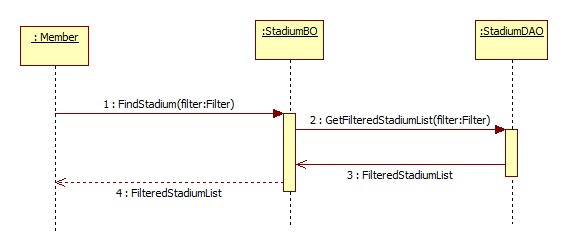
#### Deactivate Promotion



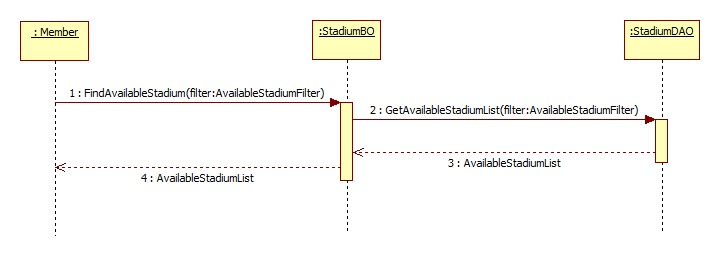
#### View Promotions



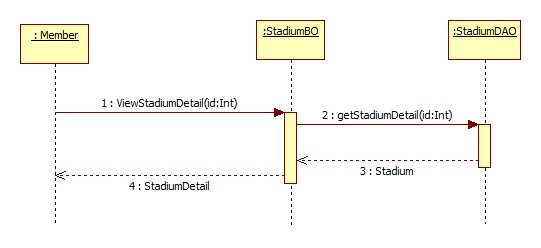
#### Find Stadium



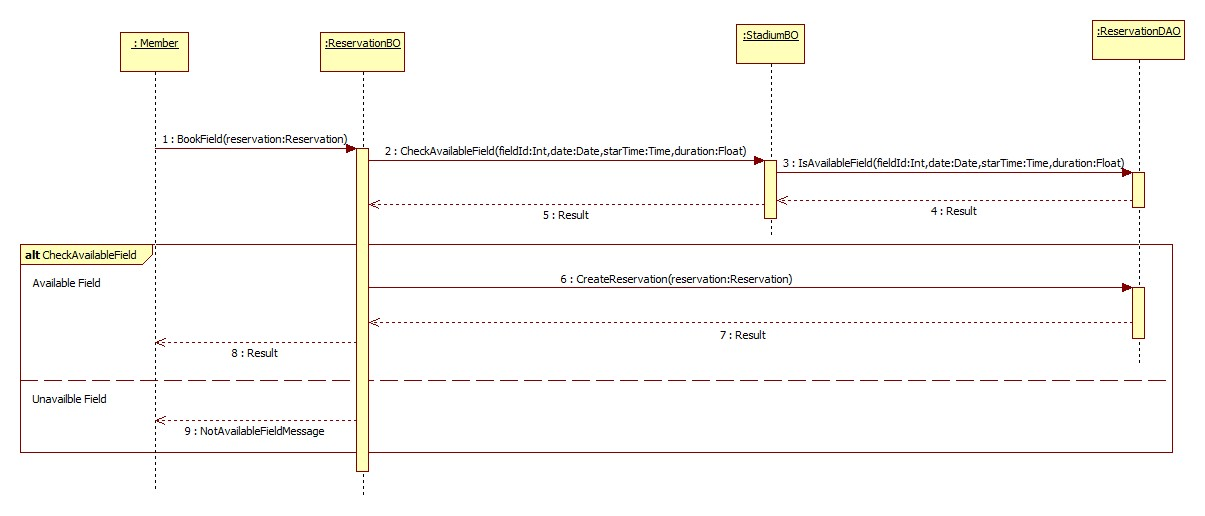
#### Find Available Stadium



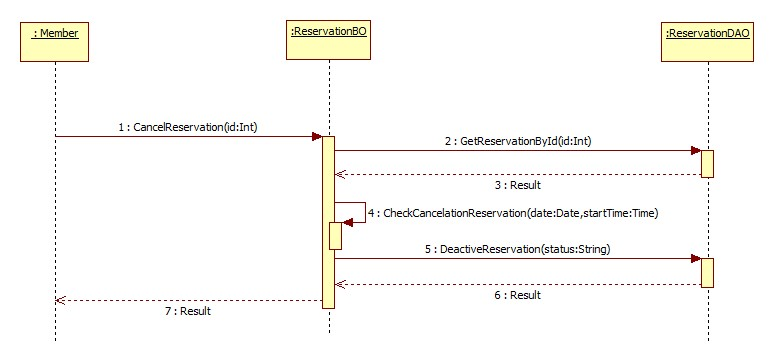
#### View Stadium Detail



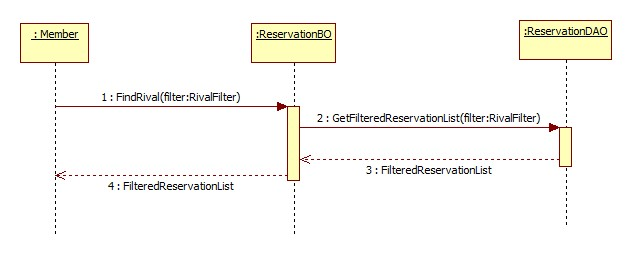
#### Book Field



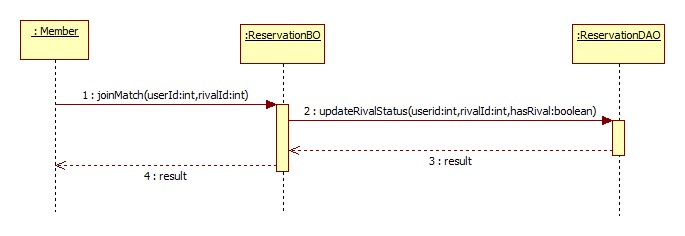
#### Cancel Reservation



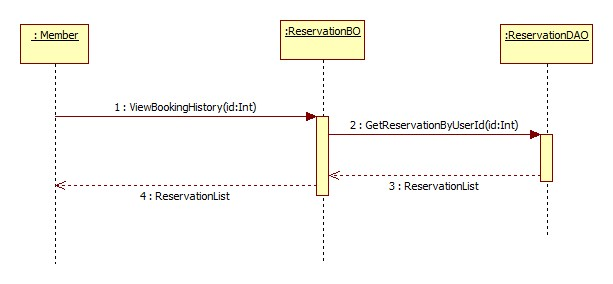
#### Find Rival



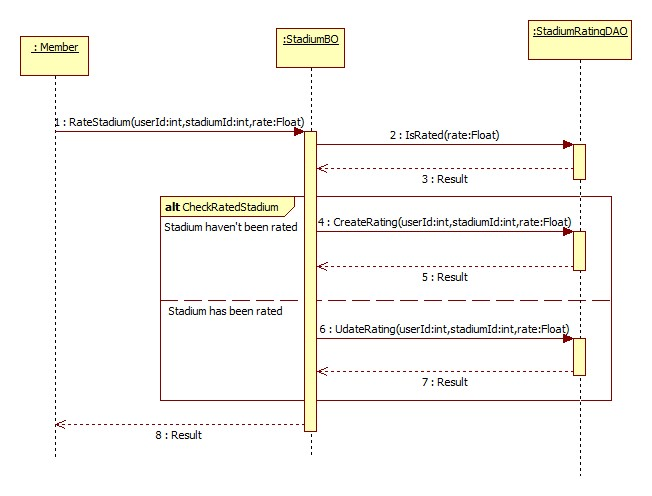
#### Join Match



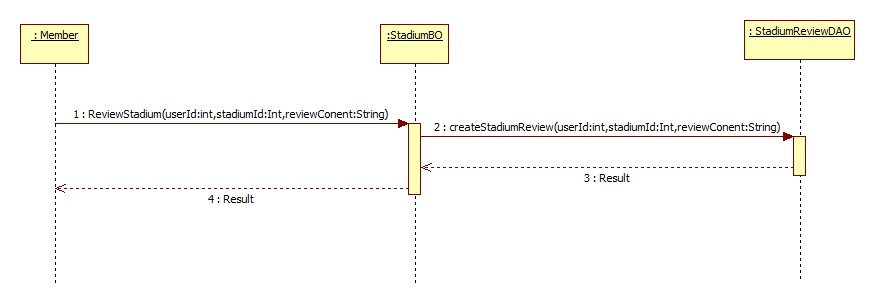
#### View Booking History



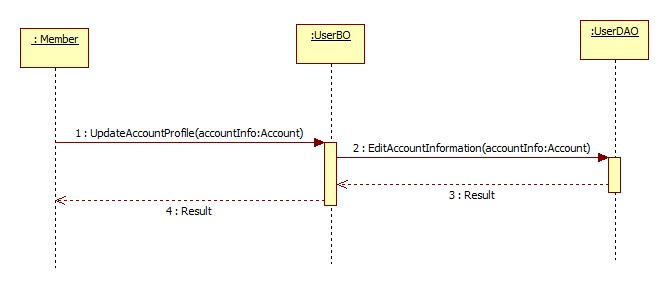
#### Rate Stadium



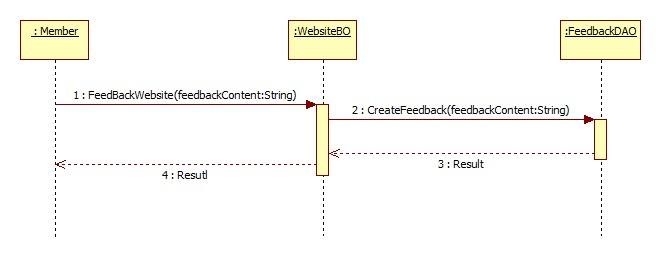
#### Review Stadium



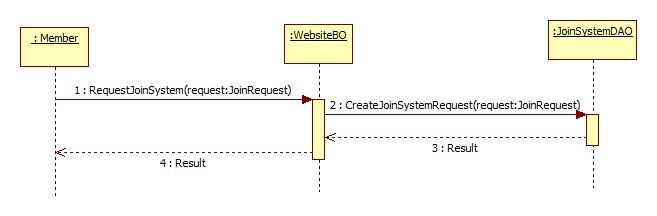
#### Update Account Profile



#### Feedback Website



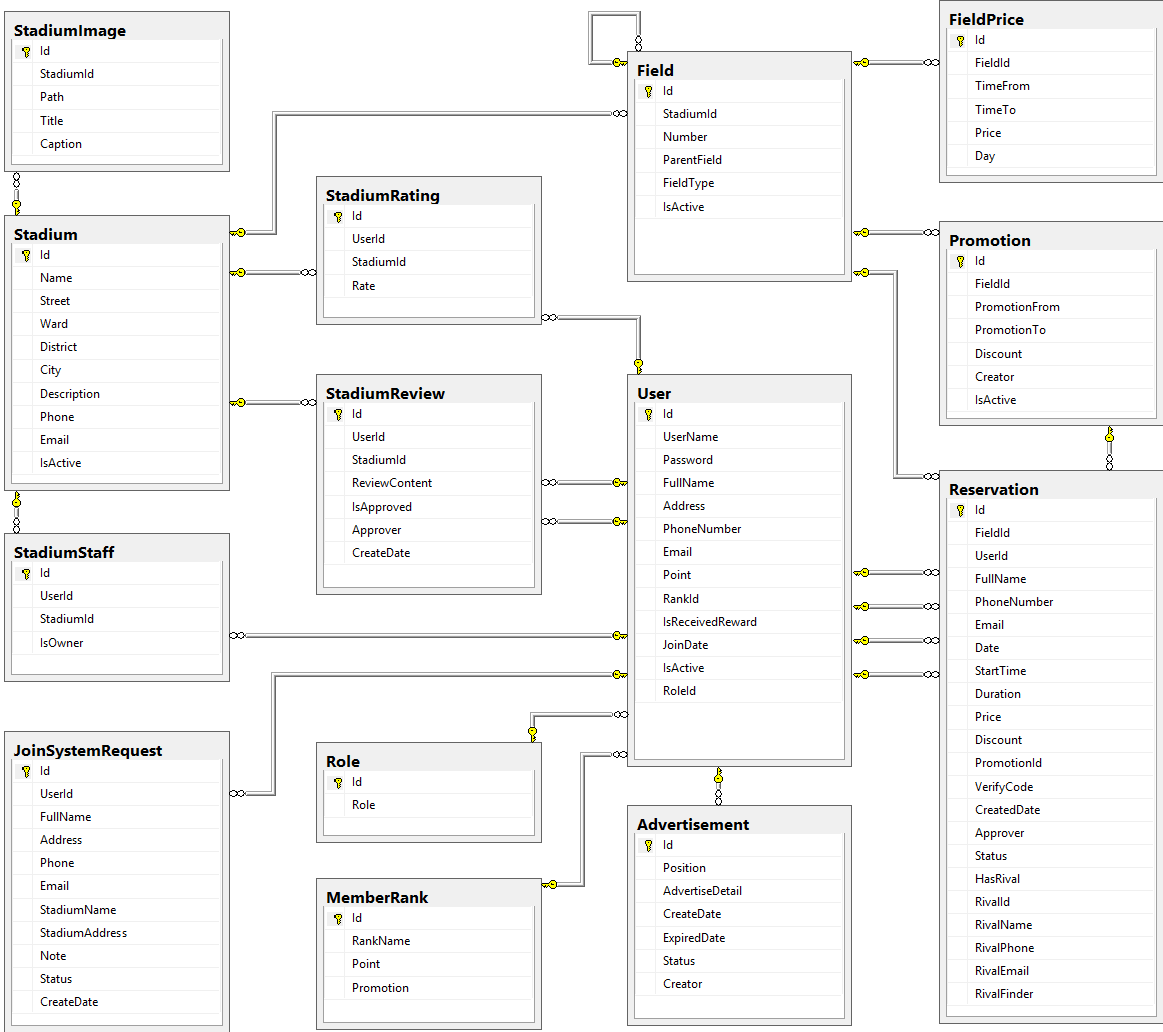
#### Request Join System



### User Interface Design/ Hardware Interface Design (if any)

Update Later

### Database Design



#### Tables description

|  |  |  |
| --- | --- | --- |
| No. | Table name | Description |
| 1 | Advertisement | Store information about advertising of website |
| 2 | Field | Store information about fields in system |
| 3 | FieldPrice | Store information about price of fields in system |
| 4 | JoinSystemRequest | Store information about join system requests in system |
| 5 | MemberRank | Store information about rank of member in system |
| 6 | Promotion | Store information about promotions of stadiums in system |
| 7 | Reservation | Store information about reservations of stadiums in system |
| 8 | Role | Store information about roles of user in system |
| 9 | Stadium | Store information about stadiums in system |
| 10 | StadiumImage | Store information about images of stadiums in system |
| 11 | StadiumRating | Store information about rating of stadiums in system |
| 12 | StadiumReview | Store information about reviews of stadiums in system |
| 13 | StadiumStaff | Store information about staffs of stadiums in system |
| 14 | User | Store information about users in system |

#### Advertisement

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this advertising |
| Position | nvarchar(10) |  |  |  |  | The position to display of the advertising |
| AdvertiseDetail | ntext |  |  |  |  | The content of this advertising |
| CreateDate | date |  |  |  |  | The date when this advertising is created |
| ExpiredDate | date |  |  |  |  | The date when this advertising is expired |
| Status | nvarchar(50) |  |  |  |  | The status of this advertising |
| Creator | int | true |  | FK |  | The creator who is created this advertising |

#### Field

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this field |
| StadiumId | int |  |  | FK |  | The index of stadium which contain this field |
| Number | nvarchar(5) |  |  |  |  | The number of this field |
| ParentField | int | true |  | FK |  | The index of field which is parent of this field |
| FieldType | int |  |  |  |  | The type of this field |
| IsActive | bit | true |  |  |  | The status of this field |

#### FieldPrice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of price of field |
| FieldId | int |  |  | FK |  | The index of field |
| TimeFrom | float |  |  |  |  | The price of field begin from this time |
| TimeTo | float |  |  |  |  | The price of field end from this time |
| Price | float |  |  |  |  | The price of field |
| Day | int |  |  |  |  | The price of fiend in this day |

#### JoinSystemRequest

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this join system request |
| UserId | int |  |  | FK |  | The user who send this join system request |
| FullName | nvarchar(50) |  |  |  |  | The full name of user who send this join system request |
| Address | nvarchar(100) |  |  |  |  | The address of user who send this join system request |
| Phone | nvarchar(20) |  |  |  |  | The phone number of user who send this join system request |
| Email | nvarchar(50) |  |  |  |  | The email of user who send this join system request |
| StadiumName | nvarchar(100) |  |  |  |  | The name of stadium which is user want to join system |
| StadiumAddress | nvarchar(100) |  |  |  |  | The address of stadium which is user want to join system |
| Note | nvarchar(1000) | true |  |  |  | The note of this request |
| Status | nvarchar(50) |  |  |  |  | The status of this request |
| CreateDate | datetime |  |  |  |  | The date when this request is created |

#### MemberRank

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this rank |
| RankName | nvarchar(50) | true |  |  |  | The name of this rank |
| Point | int | true |  |  |  | This point is condition to have this user rank |
| Promotion | nvarchar(500) | true |  |  |  | The promotion for user who have this rank |

#### Promotion

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this promotion |
| FieldId | int |  |  | FK |  | The index of field which this promotion apply |
| PromotionFrom | date |  |  |  |  | The time when this promotion is started |
| PromotionTo | date |  |  |  |  | The time when this promotion is finished |
| Discount | float |  |  |  |  | The discount per cent of this promotion |
| Creator | int | true |  |  |  | The creator who is created this promotion |
| IsActive | bit | true |  |  |  | The status of this promotion |

#### Reservation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this reservation |
| FieldId | int |  |  | FK |  | The index of field is booked |
| UserId | int | true |  | FK |  | The index of user who book the field |
| FullName | nvarchar(50) |  |  |  |  | The full name of user who book the field |
| PhoneNumber | nvarchar(20) |  |  |  |  | The phone number of user who book the field |
| Email | nvarchar(50) | true |  |  |  | The email of user who book the field |
| Date | date |  |  |  |  | The date that is booked the field |
| StartTime | float |  |  |  |  | The start time when user need the field |
| Duration | float |  |  |  |  | The duration time that user need the field |
| Price | float |  |  |  |  | The price of field |
| Discount | float | true |  |  |  | The discount per cent of the promotion is applied to this reservation |
| PromotionId | int | true |  |  |  | The index of the promotion is applied to this reservation |
| VerifyCode | nvarchar(50) |  |  |  |  | The code which use for verifying this reservation |
| CreatedDate | datetime |  |  |  |  | The date when this reservation is created |
| Approver | int | true |  | FK |  | The index of stadium staff who approve this reservation |
| Status | nvarchar(20) |  |  |  |  | The status about approving of this reservation |
| HasRival | bit |  |  |  |  | The status about having rival of this reservation |
| RivalId | int | true |  | FK |  | The index of rival who join this request rival of this reservation |
| RivalName | nvarchar(50) | true |  |  |  | The name of rival who join this request rival of this reservation |
| RivalPhone | nvarchar(20) | true |  |  |  | The phone number of rival who join this request rival of this reservation |
| RivalEmail | nvarchar(50) | true |  |  |  | The email of rival who join this request rival of this reservation |
| RivalFinder | int | true |  |  |  | The index of user who find rival for this request rival of this reservation |

#### Role

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | | Null | | Unique | | P/F key | | Default | | Description | |
| Id | | int | |  | |  | | PK, FK | |  | | The index of this role |
| Role | | nvarchar(50) | |  | |  | |  | |  | | The name of this role |

#### Stadium

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this stadium |
| Name | nvarchar(100) |  |  |  |  | The name of this stadium |
| Street | nvarchar(50) |  |  |  |  | The street address of this stadium |
| Ward | nvarchar(50) |  |  |  |  | The ward address of this stadium |
| District | nvarchar(50) |  |  |  |  | The district address of this stadium |
| City | nvarchar(50) |  |  |  |  | The city address of this stadium |
| Description | nvarchar(500) |  |  |  |  | The description about this stadium |
| Phone | nvarchar(20) |  |  |  |  | The phone number to contact with this stadium staff |
| Email | nvarchar(50) |  |  |  |  | The email to contact with this stadium staff |
| IsActive | bit | true |  |  |  | The status of this stadium |

#### StadiumImage

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this image |
| StadiumId | int |  |  | FK |  | The index of stadium which the image about |
| Path | nvarchar(100) |  |  |  |  | The path of this image store in server |
| Title | nvarchar(100) | true |  |  |  | The title of this image |
| Caption | nvarchar(200) | true |  |  |  | The caption of this image |

#### StadiumRating

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this rating |
| UserId | int |  |  | FK |  | The index of user who rate the stadium |
| StadiumId | int |  |  | FK |  | The index of stadium which is rated |
| Rate | float |  |  |  |  | The point of this rating |

#### StadiumReview

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | | | Null | | Unique | | P/F key | | Default | | Description |
| Id | | int |  | |  | | PK | |  | | The index of this review | |
| UserId | | int |  | |  | | FK | |  | | The index of user who review the stadium | |
| StadiumId | | int |  | |  | | FK | |  | | The index of stadium which is reviewed | |
| ReviewContent | | nvarchar(500) |  | |  | |  | |  | | The content of this review | |
| IsApproved | | bit |  | |  | |  | |  | | The status about approving of this review | |
| Approver | | int | true | |  | | FK | |  | | The index of stadium staff who approve this review | |
| CreateDate | | datetime |  | |  | |  | |  | | The date when this review is created | |

#### StadiumStaff

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | | Null | | Unique | | P/F key | | Default | | Description | |
| Id | | int | |  | |  | | PK | |  | | The index of this stadium staff |
| UserId | | int | |  | |  | | FK | |  | | The index of user who is staff of stadium |
| StadiumId | | int | |  | |  | | FK | |  | | The index of stadium |
| IsOwner | | bit | |  | |  | |  | |  | | The role of this staff |

#### User

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this user |
| UserName | nvarchar(50) |  |  |  |  | The user name of this user |
| Password | nvarchar(50) |  |  |  |  | The password of this user |
| FullName | nvarchar(50) | true |  |  |  | The full name of this user |
| Address | nvarchar(100) | true |  |  |  | The address of this user |
| PhoneNumber | nvarchar(20) | true |  |  |  | The phone number of this user |
| Email | nvarchar(50) |  |  |  |  | The email of this user |
| Point | int |  |  |  |  | The point of this user |
| RankId | int |  |  | FK |  | The index of rank of this user |
| IsReceivedReward | bit |  |  |  |  | The status about receiving reward of this user |
| JoinDate | date |  |  |  |  | The date when user register at website |
| IsActive | bit |  |  |  |  | The status of this user |
| RoleId | int |  |  |  |  | The index of role of this user |

### Other material (if any)

N/A