**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 5](#_Toc385642652)

[Report No.4: System Design Description (SDD) 6](#_Toc385642653)

[4.1. Design Overview 6](#_Toc385642654)

[4.2. System Architectural Design 7](#_Toc385642655)

[4.2.1. Choice of System Architecture 7](#_Toc385642656)

[4.2.2. Discussion of Alternative Designs 8](#_Toc385642657)

[4.2.3. Description of System Interface 8](#_Toc385642658)

[4.3. Component Diagram 9](#_Toc385642659)

[4.4. Detailed Description of Components 10](#_Toc385642660)

[4.4.1. Class Diagram Explanation 10](#_Toc385642661)

[4.5. Sequence Diagram 28](#_Toc385642662)

[4.5.1. Create Stadium 28](#_Toc385642663)

[4.5.2. Update Stadium 28](#_Toc385642664)

[4.5.3. Activate Stadium 29](#_Toc385642665)

[4.5.4. Deactivate Stadium 29](#_Toc385642666)

[4.5.5. View Stadiums 30](#_Toc385642667)

[4.5.6. Create Member Rank 30](#_Toc385642668)

[4.5.7. Update Member Rank 31](#_Toc385642669)

[4.5.8. Delete Member Rank 31](#_Toc385642670)

[4.5.9. View Member Ranks 32](#_Toc385642671)

[4.5.10. Create Advertising 32](#_Toc385642672)

[4.5.11. Update Advertising 33](#_Toc385642673)

[4.5.12. Delete Advertising 33](#_Toc385642674)

[4.5.13. View Advertisings 33](#_Toc385642675)

[4.5.14. Update User 34](#_Toc385642676)

[4.5.15. Activate User 34](#_Toc385642677)

[4.5.16. Deactivate User 35](#_Toc385642678)

[4.5.17. View Users 35](#_Toc385642679)

[4.5.18. Update Stadium Profile 36](#_Toc385642680)

[4.5.19. Add Field To Stadium 36](#_Toc385642681)

[4.5.20. Update Field 37](#_Toc385642682)

[4.5.21. Activate Field 37](#_Toc385642683)

[4.5.22. Deactivate Field 38](#_Toc385642684)

[4.5.23. View Fields 38](#_Toc385642685)

[4.5.24. Create Reservation 39](#_Toc385642686)

[4.5.25. Update Reservation 39](#_Toc385642687)

[4.5.26. Approve Reservation 39](#_Toc385642688)

[4.5.27. Deny Reservation 40](#_Toc385642689)

[4.5.28. View Reservations 40](#_Toc385642690)

[4.5.29. Create Promotion 40](#_Toc385642691)

[4.5.30. Update Promotion 41](#_Toc385642692)

[4.5.31. Activate Promotion 41](#_Toc385642693)

[4.5.32. Deactivate Promotion 41](#_Toc385642694)

[4.5.33. View Promotions 42](#_Toc385642695)

[4.5.34. Find Stadium 42](#_Toc385642696)

[4.5.35. Find Available Stadium 42](#_Toc385642697)

[4.5.36. View Stadium Detail 43](#_Toc385642698)

[4.5.37. Book Field 43](#_Toc385642699)

[4.5.38. Cancel Reservation 43](#_Toc385642700)

[4.5.39. Find Rival 44](#_Toc385642701)

[4.5.40. Join Match 44](#_Toc385642702)

[4.5.41. View Booking History 44](#_Toc385642703)

[4.5.42. Rate Stadium 45](#_Toc385642704)

[4.5.43. Review Stadium 45](#_Toc385642705)

[4.5.44. Update Account Profile 46](#_Toc385642706)

[4.5.45. Feedback Website 46](#_Toc385642707)

[4.5.46. Request Join System 46](#_Toc385642708)

[4.6. User Interface Design/ Hardware Interface Design (if any) 46](#_Toc385642709)

[4.7. Database Design 47](#_Toc385642710)

[4.7.1. Tables description 47](#_Toc385642711)

[4.7.2. Advertisement 48](#_Toc385642712)

[4.7.3. Configuration 48](#_Toc385642713)

[4.7.4. Field 49](#_Toc385642714)

[4.7.5. FieldPrice 49](#_Toc385642715)

[4.7.6. JoinSystemRequest 49](#_Toc385642716)

[4.7.7. MemberRank 50](#_Toc385642717)

[4.7.8. Notification 50](#_Toc385642718)

[4.7.9. Pricetable 50](#_Toc385642719)

[4.7.10. Promotion 50](#_Toc385642720)

[4.7.11. Punish Member 51](#_Toc385642721)

[4.7.12. Report user 51](#_Toc385642722)

[4.7.13. Reservation 51](#_Toc385642723)

[4.7.14. Role 53](#_Toc385642724)

[4.7.15. Stadium 53](#_Toc385642725)

[4.7.16. StadiumImage 54](#_Toc385642726)

[4.7.17. StadiumRating 54](#_Toc385642727)

[4.7.18. StadiumReview 54](#_Toc385642728)

[4.7.19. StadiumStaff 54](#_Toc385642729)

[4.7.20. User 55](#_Toc385642730)

[4.7.21. UserDistance 55](#_Toc385642731)

[4.8. Algorithm Of Business Intelligence 55](#_Toc385642732)

[4.8.1. Overview: 55](#_Toc385642733)

[4.8.2. Implement: 56](#_Toc385642734)

[4.8.3. Algorithm 57](#_Toc385642735)

## 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 | FPB | Football Pitches Booking |
| 3 | ERD | Entity Relationship Diagram |
| 4 | API | Application Programmable Interface |
| 5 |  |  |

## 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

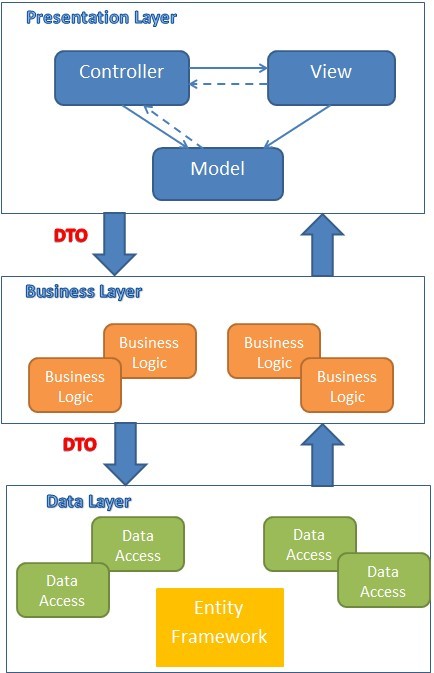
****

Figure 1: 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

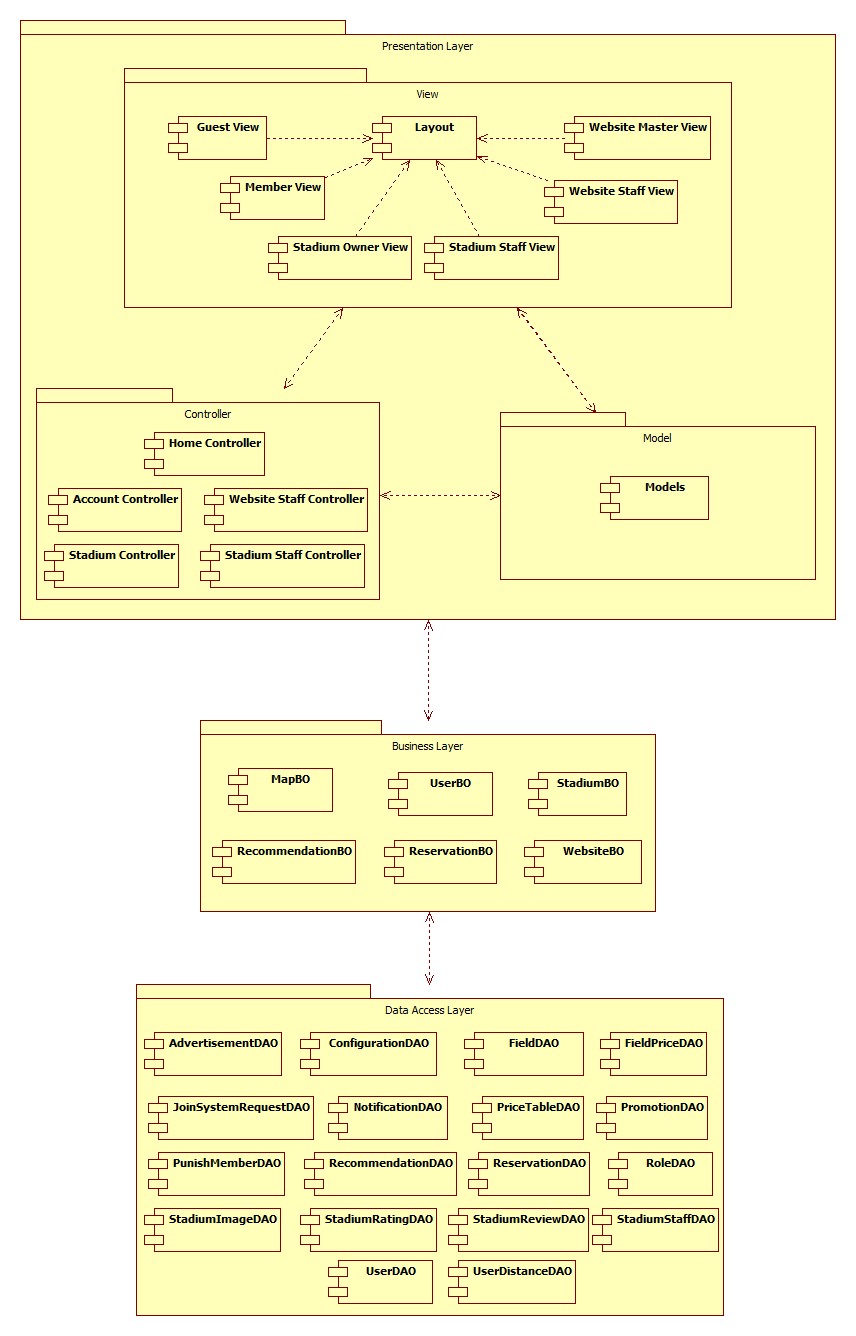


Figure 2: Component Diagram

### Detailed Description of Components

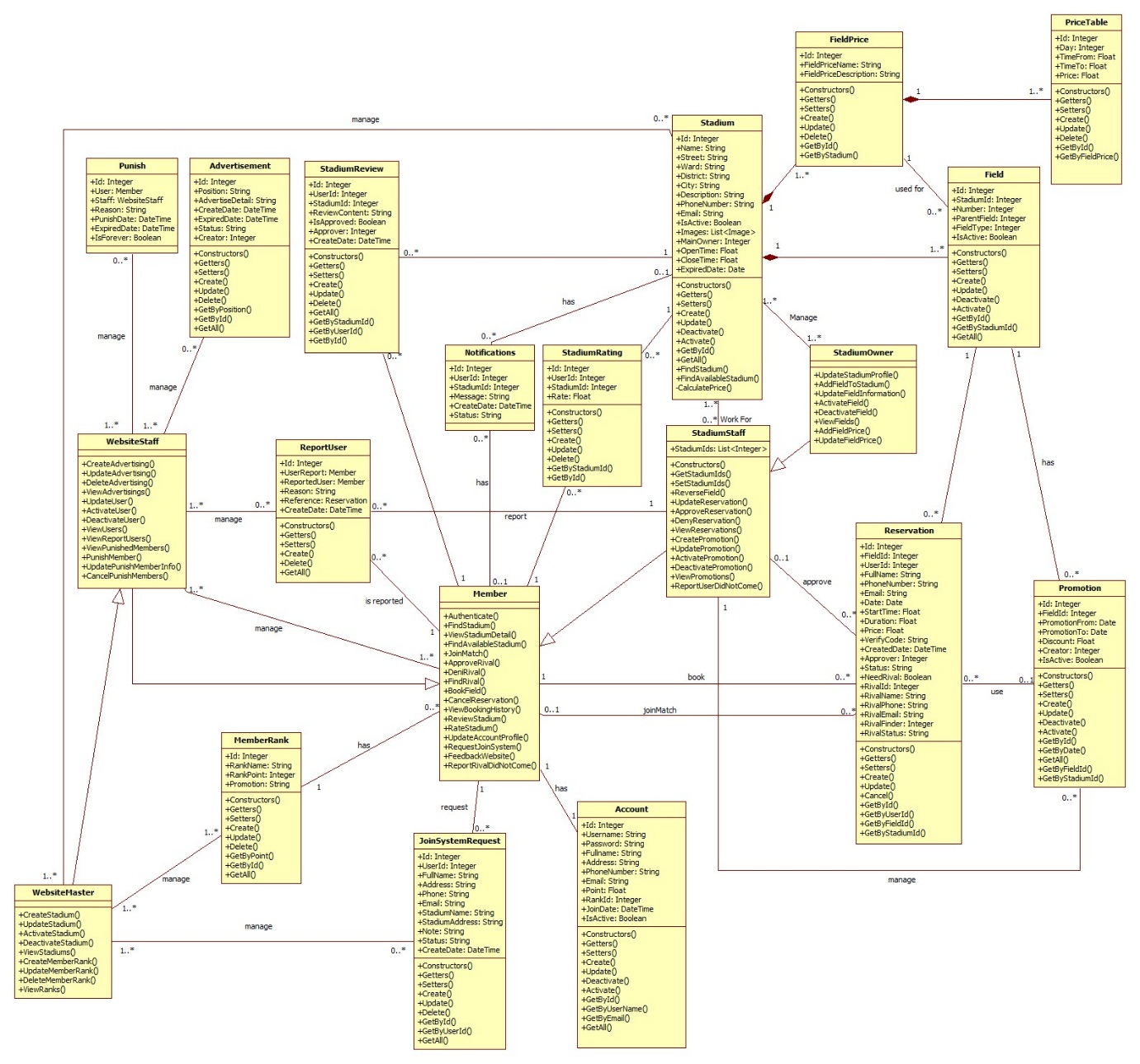


Figure 3: General Class Diagram

#### 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



Figure 4: Create Stadium

#### Suggest Stadium

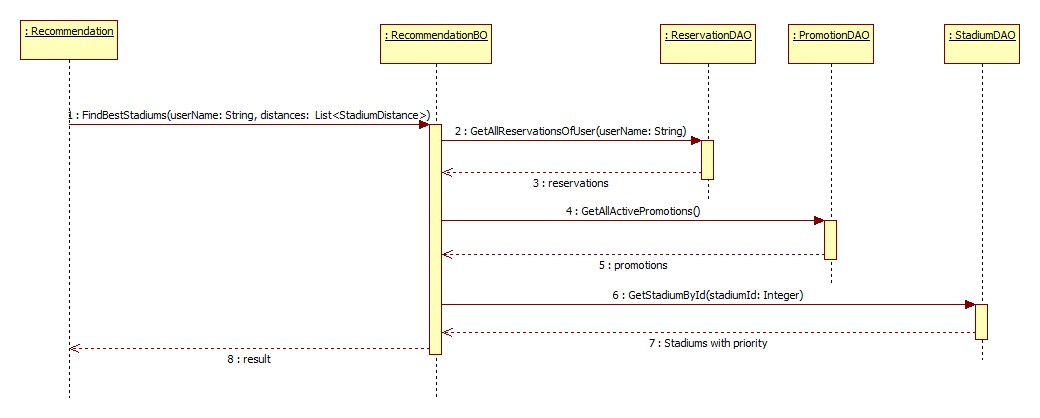


Figure 5: Suggest Stadium

#### Delete Member Rank

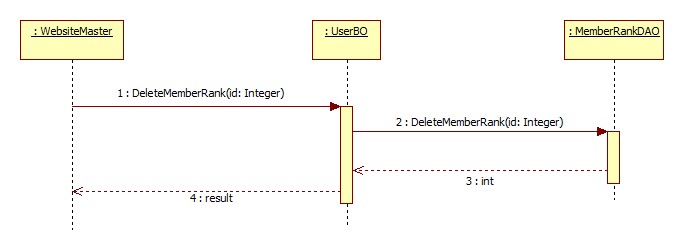


Figure 6: Delete Member Rank

#### Create Price Table

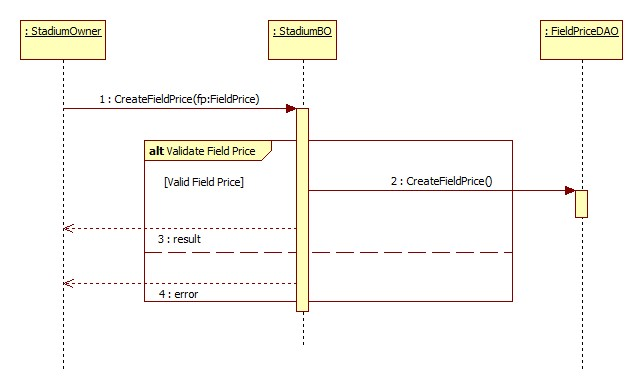


Figure 7: Create Price Table

#### Create Reservation

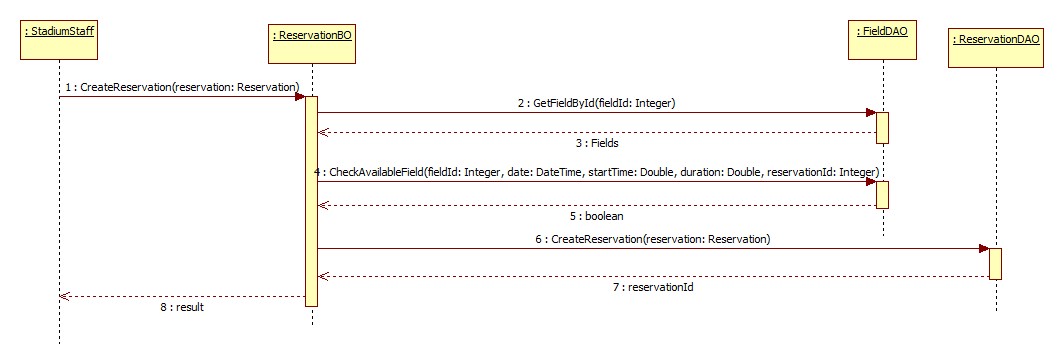


Figure 8: Create Reservation

#### Update Reservation

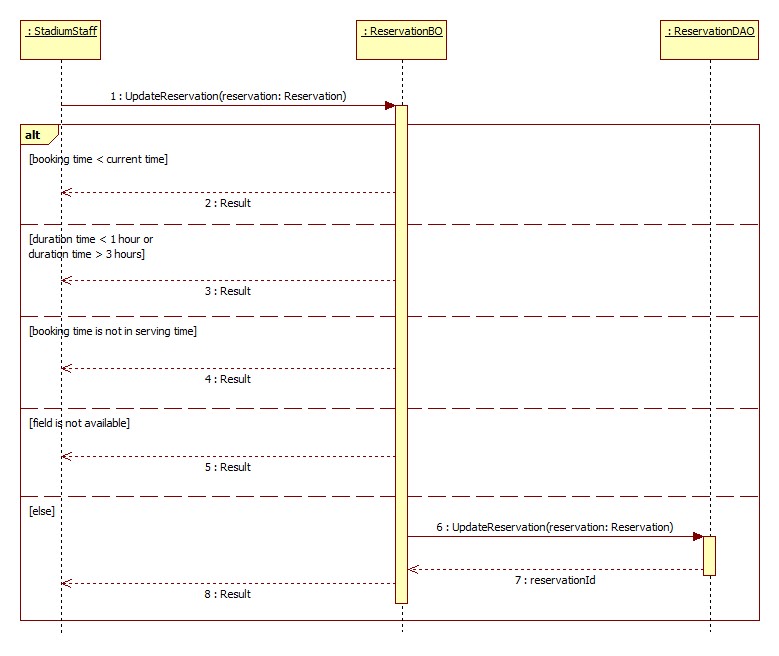


Figure 9: Update Reservation

#### Find Available Stadium

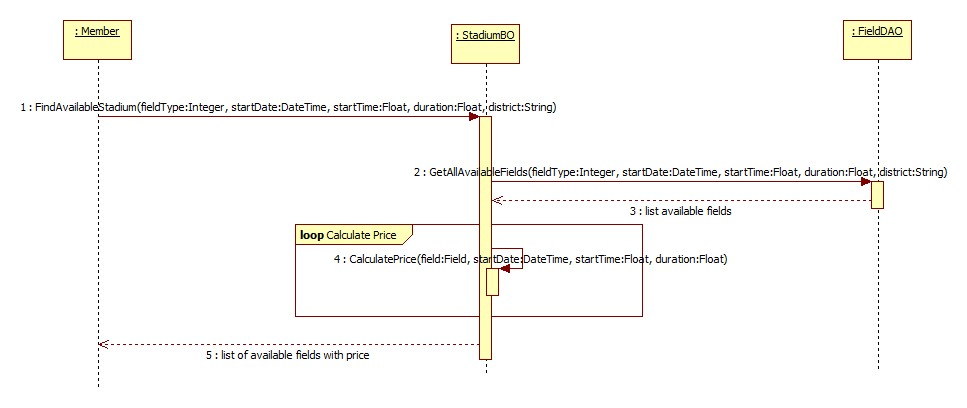


Figure 10: Find Available Stadium

#### Create Promotion

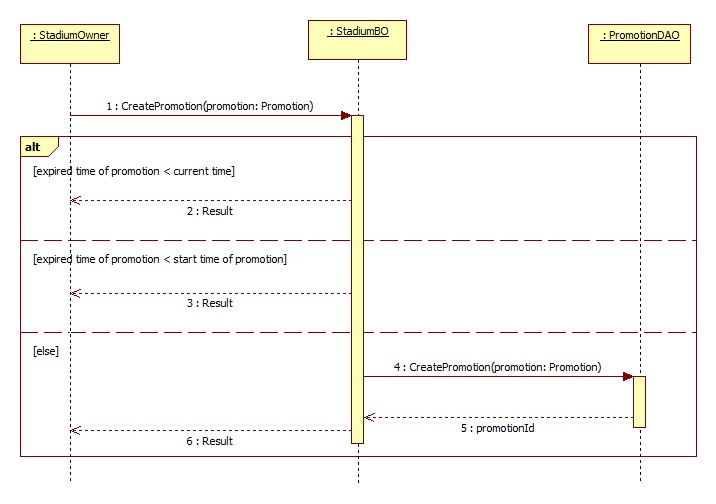


Figure 11: Create Promotion

#### Book Field

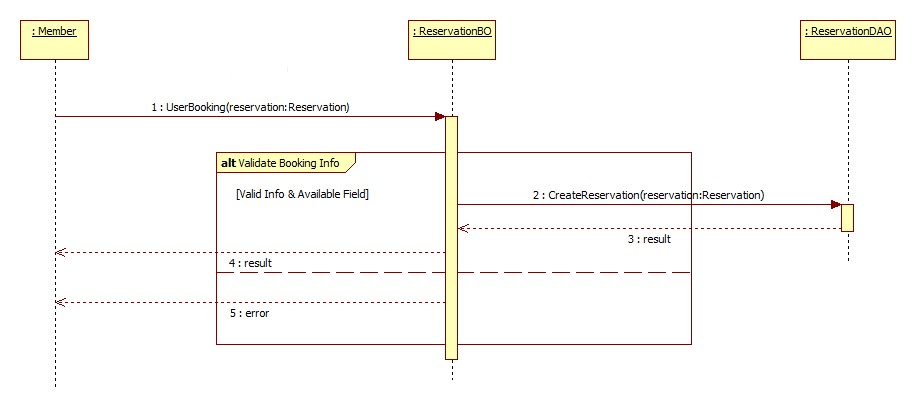


Figure 12: Book Field

#### Cancel Reservation

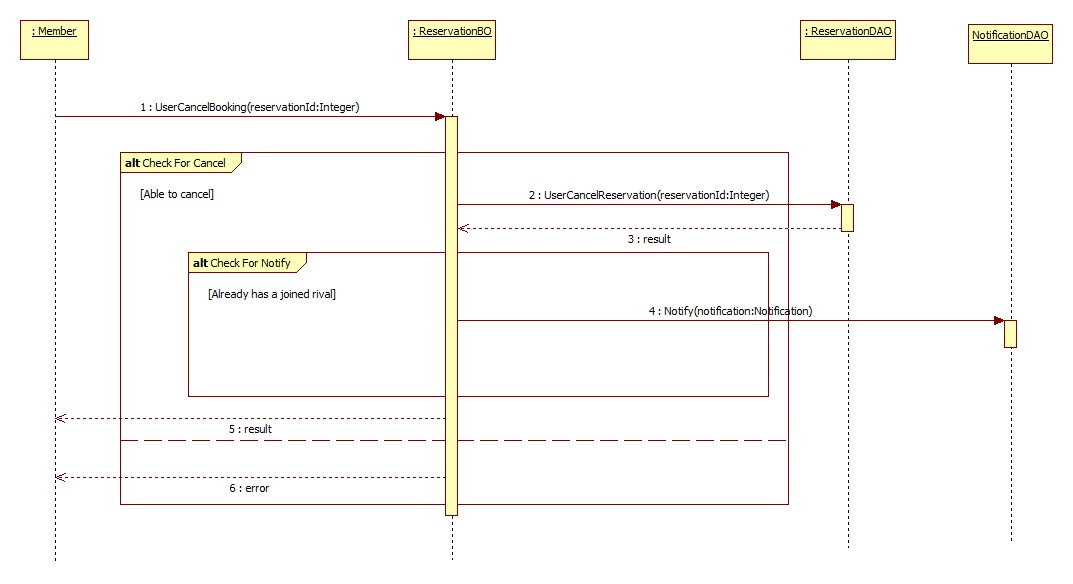


Figure 13: Cancel Reservaion

#### Find Rival

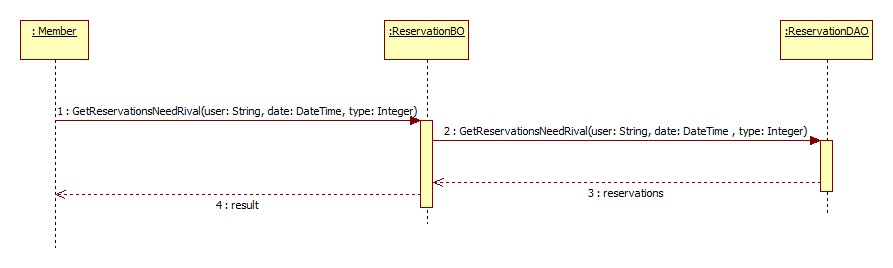


Figure 14: Find Rival

#### Join Match

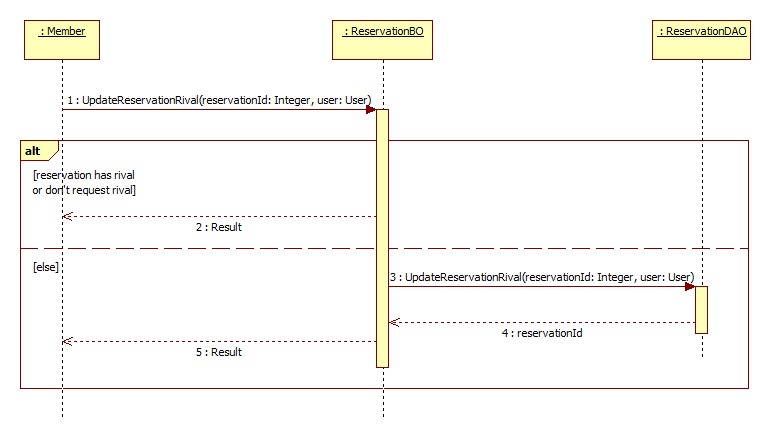


Figure 15: Join Match

#### View Booking History

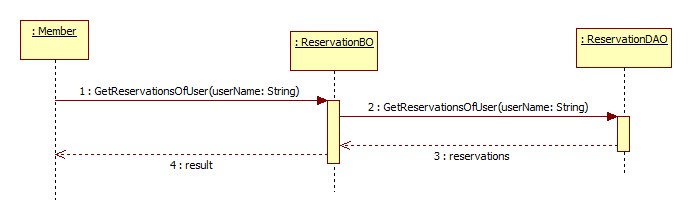


Figure 16: View Booking History

#### Rate Stadium

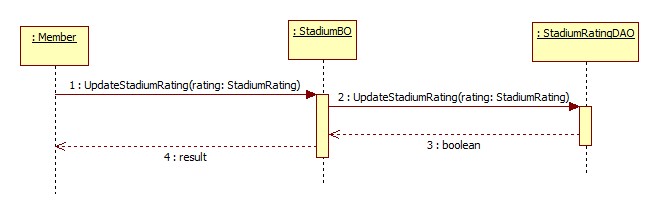


Figure 17: Rate Stadium

#### Review Stadium

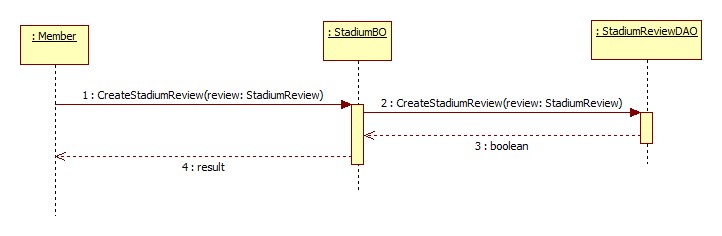


Figure 18: Review Stadium

#### Update Account Profile

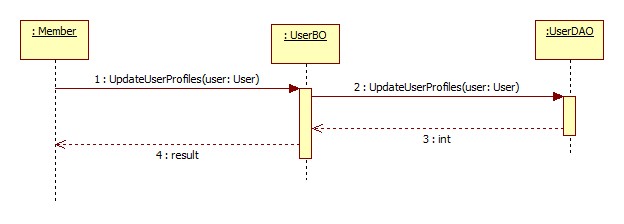


Figure 19: Update Account's Profile

#### Feedback Website

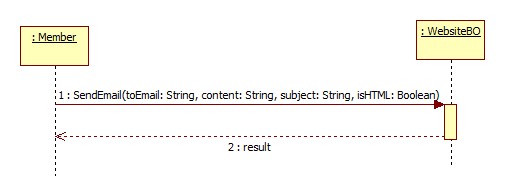


Figure 20: Feedback Website

#### Request Join System

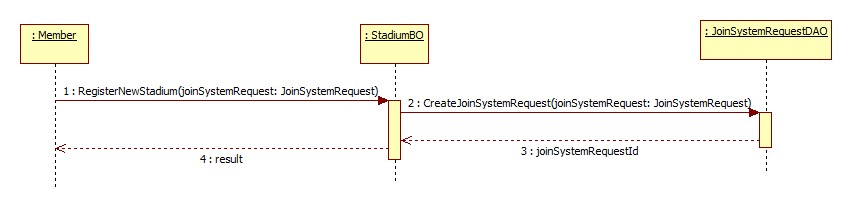


Figure 21: Request Join System

#### Update Stadium

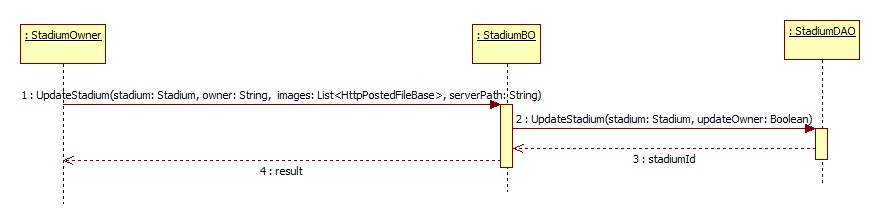


Figure 22: Update Stadium

#### Update Configuration

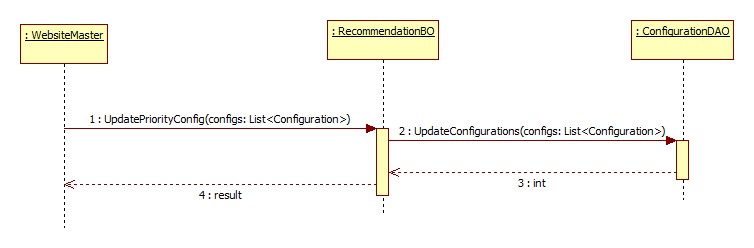


Figure 23: Update Configuration

#### Suggest Rival

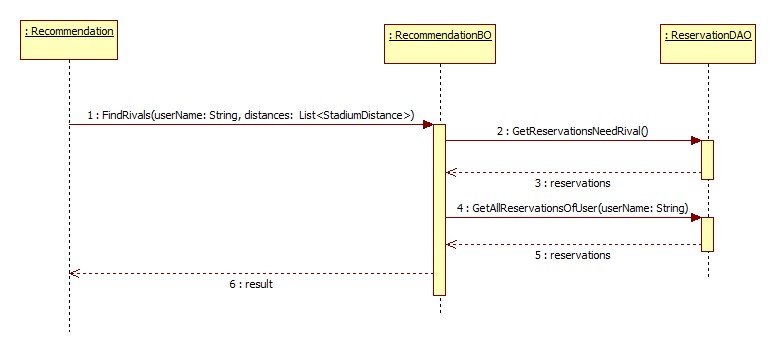


Figure 24: Suggest Rival

#### View Stadiums

#### Create Member Rank

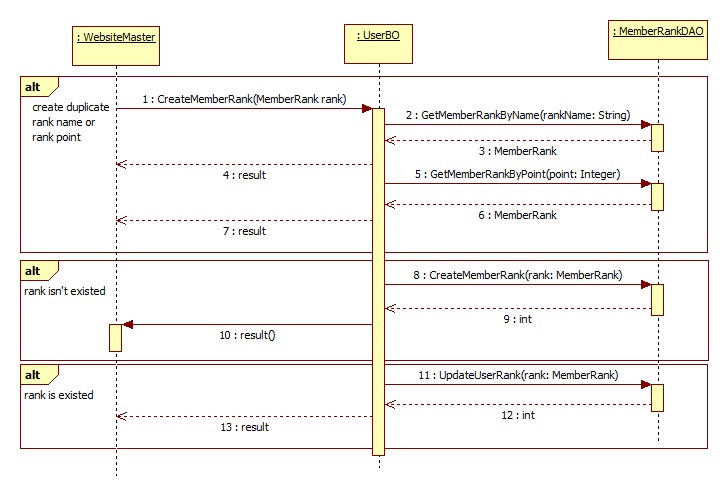


Figure 25: Create Member Rank

#### Update Member Rank

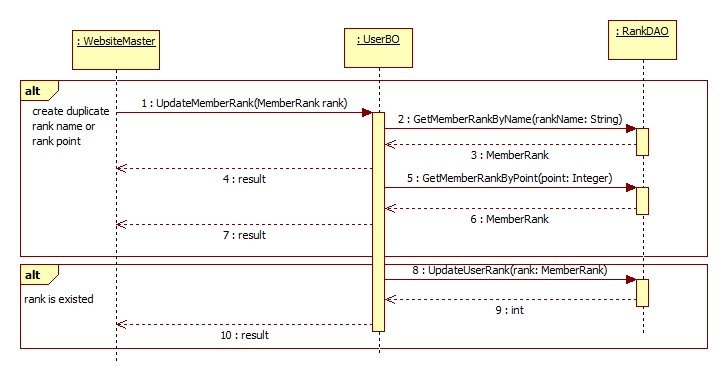


Figure 26: Update Member Rank

#### Create Advertising

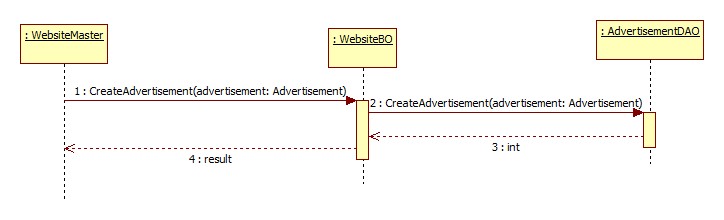


Figure 27: Create Advertising

#### Update Advertising

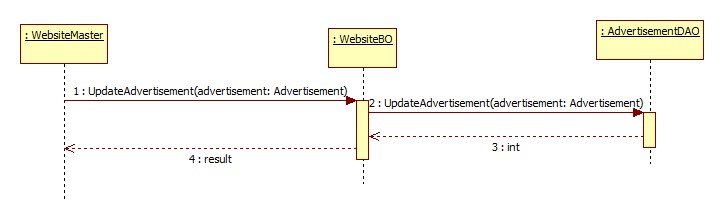


Figure 28: Update Advertising

#### Delete Advertising

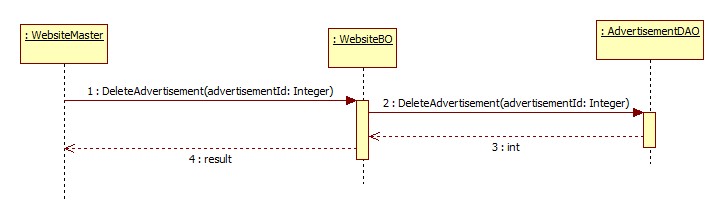


Figure 29: Delete Advertising

#### Punish User

#### Update User

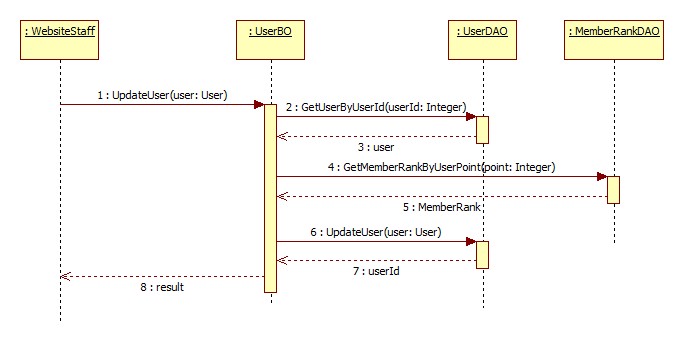


Figure 30: Update User

#### Update Stadium Profile

#### Update Field

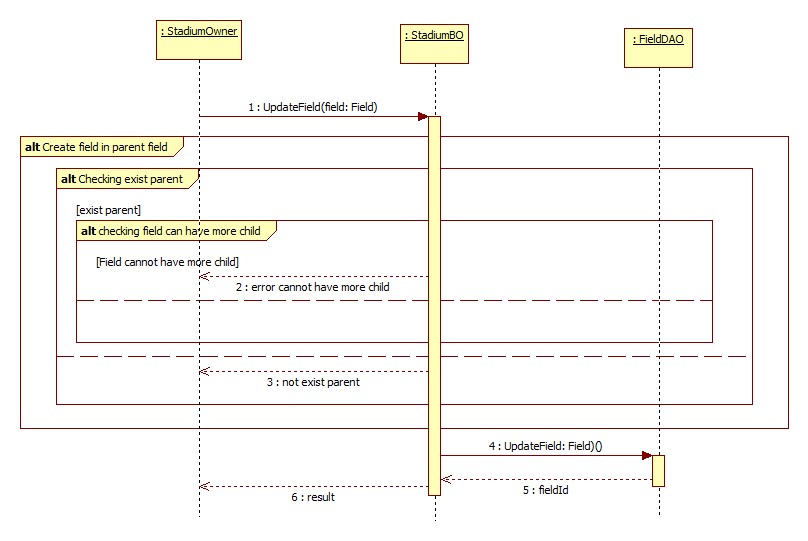


Figure 31: Update Field

#### Update Reservation Status

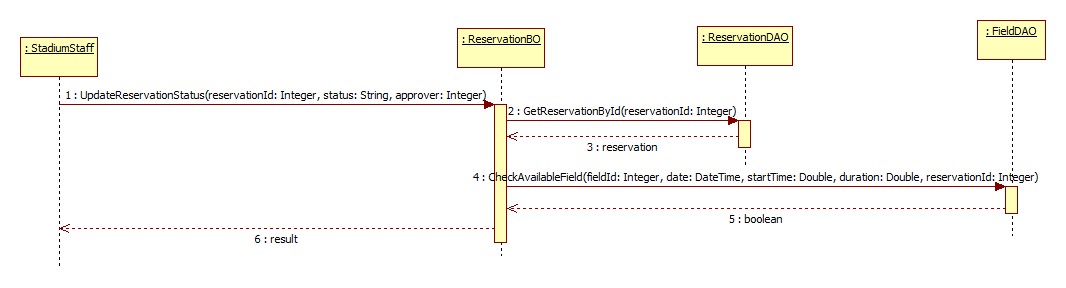


Figure 32: Update Reservation Status

#### Update Promotion

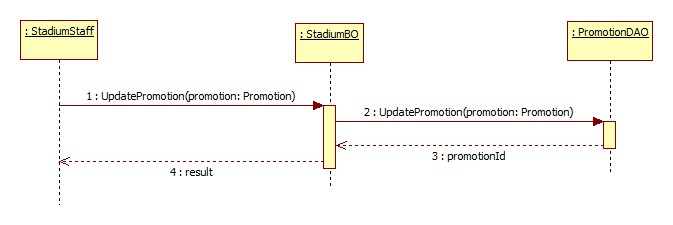


Figure 33: Update Promtion

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

Update Later

### Database Design

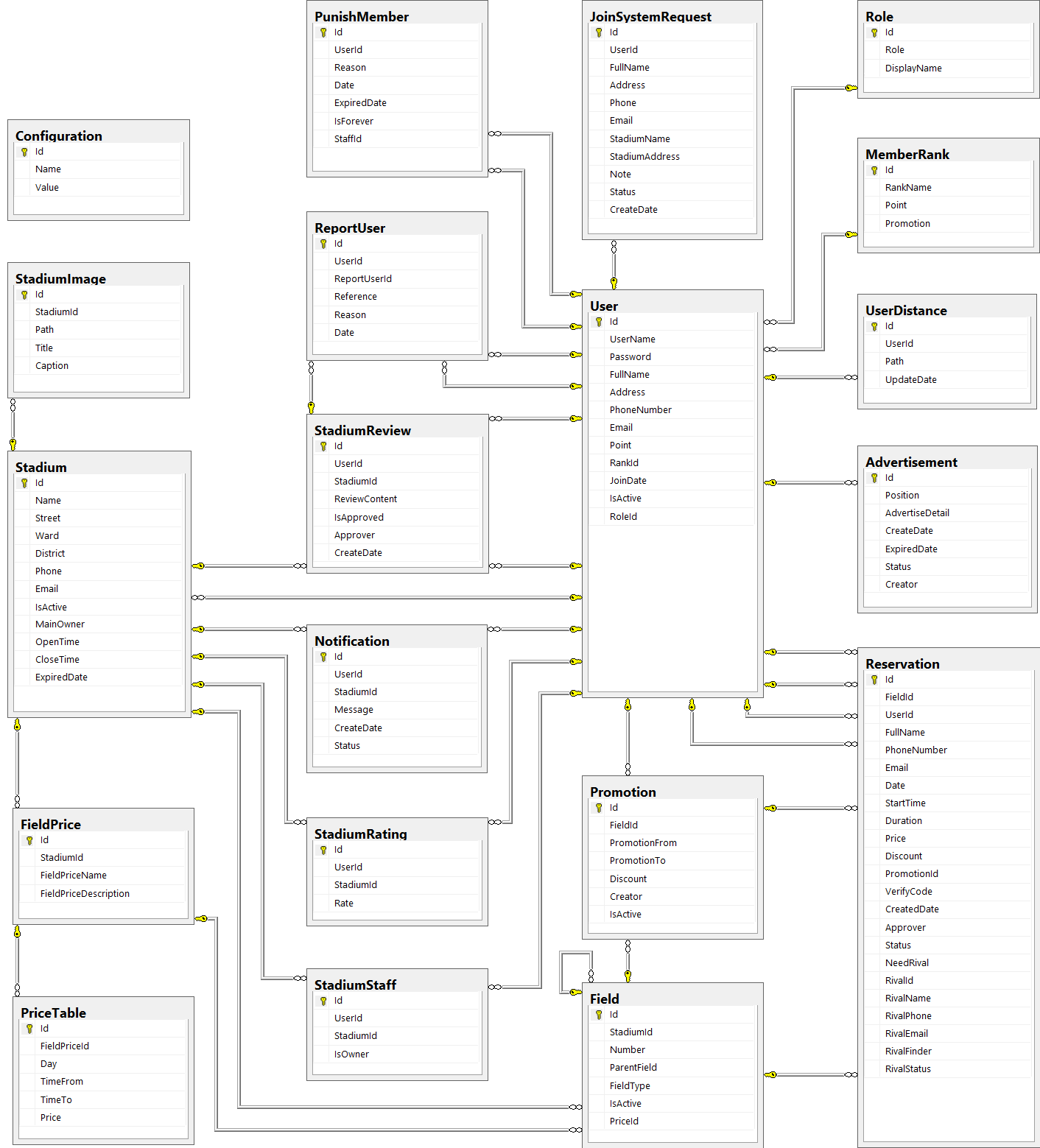


Figure 34: Database Diagram

#### Tables description

|  |  |  |
| --- | --- | --- |
| No. | Table name | Description |
| 1 | Advertisement | Store information about advertising of website |
| 2 | Configuration | Store information about configuration of website |
| 3 | Field | Store information about fields in system |
| 4 | FieldPrice | Store information about price of stadiums in system |
| 5 | JoinSystemRequest | Store information about join system requests in system |
| 6 | MemberRank | Store information about rank of member in system |
| 7 | Notification | Store information about notification of members and stadiums in system |
| 8 | PriceTable | Store informatino about price table of field price in system |
| 9 | Promotion | Store information about promotions of stadiums in system |
| 10 | PunishMember | Store information about punishing members in system |
| 11 | ReportUser | Store information about reporting users in system |
| 12 | Reservation | Store information about reservations of stadiums in system |
| 13 | Role | Store information about roles of user in system |
| 14 | Stadium | Store information about stadiums in system |
| 15 | StadiumImage | Store information about images of stadiums in system |
| 16 | StadiumRating | Store information about rating of stadiums in system |
| 17 | StadiumReview | Store information about reviews of stadiums in system |
| 18 | StadiumStaff | Store information about staffs of stadiums in system |
| 19 | User | Store information about users in system |
| 20 | UserDistance | Store information about the distance of 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 |

#### Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this configuration |
| Name | nvarchar(100) |  |  |  |  | The name of this configuration |
| Value | nvarchar(100) |  |  |  |  | The value of this configuration |

#### 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 |
| PriceId | int |  |  | FK |  | The index of price of this field |

#### FieldPrice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of price of field |
| StadiumId | int |  |  | FK |  | The index of stadium |
| FieldPriceName | nvarchar(200) |  |  |  |  | Price name of this field |
| FieldPriceDescription | nvarchar(200) |  |  |  |  | Description of the field price |

#### 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) |  |  |  |  | The name of this rank |
| Point | int |  |  |  |  | This point is condition to have this user rank |
| Promotion | nvarchar(500) | true |  |  |  | The promotion for user who have this rank |

#### Notification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this notification |
| UserId | int |  |  | FK |  | Index of this user |
| StadiumId | int |  |  | Fk |  | Index of the stadium |
| Message | nvarchar(2000) |  |  |  |  | Content of message |
| CreateDate | datetime |  |  |  |  | Time when notification appear |
| Status | Nvarchar(50) |  |  |  |  | Status of this notification |

#### Pricetable

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this price table |
| FieldPriceId | int |  |  | FK |  | The name of this rank |
| Day | int |  |  |  |  | Day of this price table |
| TimeFrom | float |  |  |  |  | Start time of this price table |
| TimeTo | float |  |  |  |  | End time of this price table |
| Price | float |  |  |  |  | Price of each price table |

#### 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 |

#### Punish Member

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this punishing |
| UserId | int |  |  | FK |  | The index of user who will be punished |
| Reason | Nvarchar(200) |  |  |  |  | Reason of this punishing |
| Date | datetime |  |  |  |  | Punish time of this punishing |
| ExpiredDate | datetime |  |  |  |  | Unbanned time of this punishing |
| IsForever | bit |  |  |  |  | Check is it forever punish |
| StaffId | int | true |  | FK |  | Index of the staff who is punish this user |

#### Report user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | Null | Unique | P/F key | Default | Description |
| Id | int |  |  | PK |  | The index of this promotion |
| UserId | int |  |  | FK |  | The index of the user who is reported |
| ReportUserId | int |  |  |  |  | The index of the user who reports |
| Reason | nvarchar(50) |  |  |  |  | Reason of the report |
| Reference | int |  |  | FK |  | Reference of this report |
| Date | date |  |  |  |  | Create date of this report |

#### 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 |
| 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 |
| NeedRival | 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 |
| RivalStatus | nvarchar(20) | true |  |  |  | The status about approving of this rival |

#### 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 |
| DisplayName | | nvarchar(50) | | true | |  | |  | |  | | The display 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 |
| 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 |
| MainOwner | int |  |  | FK |  | The owner of this stadium |
| OpenTime | float |  |  |  |  | Opened time of this stadium |
| CloseTime | float |  |  |  |  | Closed time of this stadium |
| ExpiredDate | date |  |  |  |  | Limited date 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 |
| 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 |

#### UserDistance

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type | | Null | | Unique | | P/F key | | Default | | Description | |
| Id | | int | |  | |  | | PK | |  | | The index of the user stadium |
| UserId | | int | |  | |  | | FK | |  | | The index of the user |
| Path | | nvarchar(50) | |  | |  | |  | |  | | The path of this user distance |
| IsOwner | | bit | |  | |  | |  | |  | | The role of this staff |

### Algorithm Of Business Intelligence

#### Overview:

This automatic function helps system to suggest stadium or rival when users login. It makes the website becomes more friendly and convenience for user. We apply Google API to calculate the distance from user’s address to stadium’s address, storage the reservation history of user to understand their habits, and get the promoted information of stadium to give user best advices.

Therefore we use three norms to recommend the stadium for user:

* Most booked stadium
* The nearest stadium
* Stadium which is discounted

#### Implement:

* User logins system
* System gets the reservation history of this user, all the available promoted stadiums and the distance from user to each stadium.
* Each recommended stadium is calculated by three norms: the stadium which is most booked, the stadium which is nearest to user’s address and the most discount stadium.
* System calculates the priority of each norms and combines it into final priority to decide the order of stadium which is displayed on Homepage.
* Homepage has 4 categories of stadium: The best stadium (balance rate of priority), most booked stadium(highest priority rate for the most booked stadium), most nearest stadium (highest priority rate for the nearest stadium), and most discount stadium (highest priority rate for the most discounted stadium).
* The best stadium, most booked stadium, most nearest stadium, and most discount stadium have the same method to be suggested but they have the difference value of priority rate.
* For example, Most Booked Stadium is suggested by: the highest priority rate for most booked stadium (60 percent), the lower priority rate for near stadium (20 percent) and discounted stadium (20 percent).
* In addition, the value of priority rate is able to be changed. It depend on taste of customer. If they interest more about near stadium, website master can raise the rate of nearest stadium priority, and decrease the others dow.

Beside that we apply the same way to suggest rival for player who wants to find competitors or have not enough players for a match.

Suggest Stadium

The Best Stadium

The Nearest Stadium

The Most

Booked Stadium

The Most

Discounted Stadium

Nearest priority

Most booked

priority

Most discounted

priority

Nearest priority

Most booked

priority

Most discounted

priority

Nearest priority

Most booked

priority

Most discounted

priority

Nearest priority

Most booked

priority

Most discounted

priority

Figure 35: Stadium Suggestion

#### Algorithm

**Most Booked Stadium Point (MBSP):**

* Get all reservation, we assume number of stadiums were booked
* Reduced Point = 10 / number of stadiums were booked
* Stadium with max reservation: set 10 mark
* Next Stadium mark =

10 – [(Max reservation number – Next reservation number) x Reduced Point]

* Calculate to the end of booked stadiums list
* We have mark of each booked stadium

**Most Discounted Stadium Point(MDSP):**

* Get all available promoted stadium, assume number of stadium are being discounted
* Reduced Point = 10 / number of stadium are being discounted.
* Stadium with the highest discount: set 10 mark
* Next Stadium mark =

10 – [(highest discount amount – next discount amount) x Reduced Point]

* Calculate to the end of the discounted stadiums list.
* We have mark of each promoted satdium

**Nearest Priority Stadium Point(NPSP):**

* Distance from user’s address to stadium < 2km: set 10 mark
* Distance from user’s address to stadium < 4km: set 9 mark
* Distance from user’s address to stadium < 8km: set 8 mark
* ….
* Distance from user’s address to stadium out of 20 km: 0 mark

We have mark of each stadium near user’s address

**Summarise:**

* Most booked priority: **MostBooked**
* Most discount priority: **MostPromoted**
* Distance priority: **Nearest**

Final priority of each stadium =

(**MostBooked** x **MBSP) + (MostPromoted** x **MDSP)** + (**Nearest** x **NPSP)**

Note: Value of MostBooked, MostPromoted and Nearest is store in Tbl.Configuration. It’s able to change.

Result: The stadium with the highest priority is display firstly in each category on Home page.