

**FOOTBALL FIELD BOOKING SYSTEM**

**Software Design Document**

– HCM, August 2022 –

**Table of Contents**

[I. Overview](#_heading=h.gjdgxs) **3**

[1. Code Packages/Namespaces](#_heading=h.30j0zll) 3

[2. Database Schema](#_heading=h.1fob9te) 3

[II. Code Designs](#_heading=h.3znysh7) **5**

[1. <Feature/Function Name1>](#_heading=h.2et92p0) 5

[a. Class Diagram](#_heading=h.tyjcwt) 7

[b. Class Specifications](#_heading=h.3dy6vkm) 8

[XYZ Class](#_heading=h.gtv3gs42fiiv) 8

[ABC Class](#_heading=h.cvlkrhhn86sy) 8

[c. Sequence Diagram(s)](#_heading=h.1t3h5sf) 8

[d. Database queries](#_heading=h.4d34og8) 8

[2. <Feature/Function Name2>](#_heading=h.2s8eyo1) 9

[III. Database Tables](#_heading=h.17dp8vu) **9**

[1. tblUsers](#_heading=h.3rdcrjn) 9

[2. TblRoles](#_heading=h.26in1rg) 9

[3. TblFeedback](#_heading=h.udbkz57q2rjf) 9

[4. TblCity](#_heading=h.fvrf23ngk5ku) 9

[5. TblLocation](#_heading=h.lop5b7kzr27x) 10

[6. TblFields](#_heading=h.k0uu4fw4xm2q) 10

[7. TblFieldCategory](#_heading=h.9sc0utw51n3u) 10

[8. TblSlot](#_heading=h.df64n600y2) 10

[9. TblSlotDetail](#_heading=h.rt18z3b19qhq) 11

[10. TblFoodCategory](#_heading=h.8tx7fgaxhnix) 11

[11. TblFoods](#_heading=h.ge1j7sk9svjw) 11

[12. TblFoodDetail](#_heading=h.etu86ins4uq5) 11

[13. TblBooking](#_heading=h.ihu8tcf2v1kd) 11

[14. TblBookingDetail](#_heading=h.bfbt27171yug) 12

# I. Overview

## 1. Code Packages/Namespaces

*[Provide the package diagram for each sub-system. The content of this section including the overall package diagram, the explanation, package and class naming conventions in each package. Please see the sample and description table format below – following Java project naming convention]*



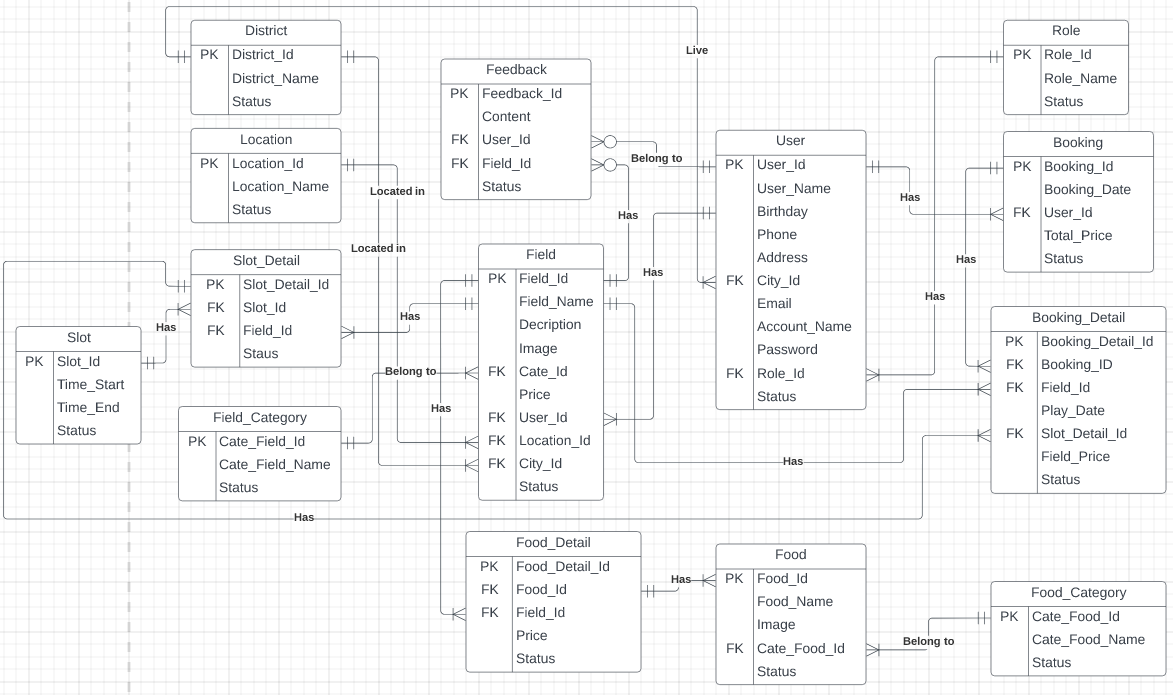
***Package descriptions & package class naming conventions***

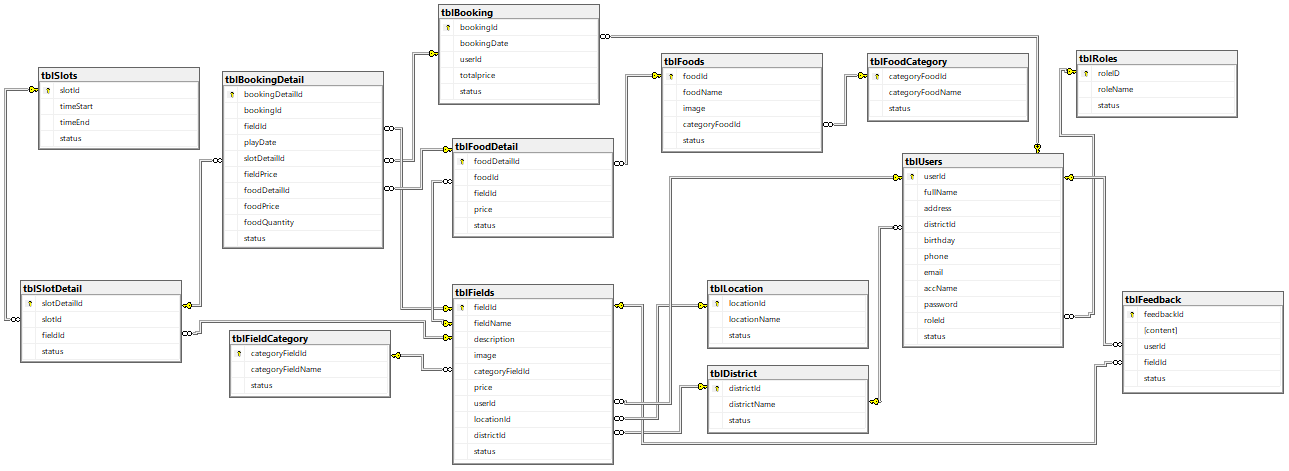
| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *controllers* | *<Description of the package>*  *-Contains servlets to receive user requests from jsp pages to process requests and return results.*  *<Class naming convention>*  *-Name in such a way that all the first letter of each word is capitalized* |
| *02* | *daos* | *-Contains functions that perform database interactions for controllers when they are called*  *–Named in such a way that the first letter of each word is capitalized with a DAO* |
| *03* | *utis* | *-Make java project link with database*  *-Naming with initials 2 (DB) and capitalize first word of utils* |
| *04* | *dto* | *-Contains project objects*  *-Name in such a way that all the first letter of each word is capitalized* |
| *05* | *stackjava.com.accessgoogle.common* | *-Make a connection with google*  *-Contains the object that google returns*  *-Contains methods to be able to log in with google accounts*  *-Name in such a way that all the first letter of each word is capitalized* |
| *06* | *stackjava.com.accessgoogle.servlet* | *-Servlet handle connection with google*  *-Name in such a way that all the first letter of each word is capitalized* |

## 

## 2. Database Schema

ERD + Database Diagram





***Table descriptions & package class naming conventions are as below***

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | tblUsers | The table contains the user's information.  - Primary keys: userId  - Foreign keys: roleId, cityId |
| 02 | tblFields | The table contains the field's information  - Primary keys: fieldId  - Foreign keys: categoryId, userId, locationId, cityId |
| 03 | tblRole | The table contains the role's information  - Primary keys: roleId |
| 04 | tblBooking | The table contains the user's booking information  - Primary keys: bookingId  - Foreign keys: userId |
| 05 | tblBookingDetail | The table contains the user's booking detail information  - Primary keys: bookingDetailId  - Foreign keys: bookingId, fieldId, slotDetailId |
| 06 | tblLocation | The table contains the location's information  - Primary keys: locationId |
| 07 | tblCity | The table contains the city's information  - Primary keys: cityId |
| 08 | tblFieldCategory | The table contains the field category's information  - Primary keys: categoryFieldId |
| 09 | tblFeedback | The table contains the feedback's information  - Primary keys: feedbackId  - Foreign keys: userId, fieldId |
| 10 | tblSlot | The table contains the slot's information  - Primary keys: slotId |
| 11 | tblSlotDetail | The table contains the slot detail's information  - Primary keys: slotDetailId  - Foreign keys: slotId, fieldId |
| 12 | tblFood | The table contains the food's information  - Primary keys: foodId  - Foreign keys: categoryFoodId |
| 13 | tblFoodDetail | The table contains the food detail's information  - Primary keys: foodDetailId  - Foreign keys:foodId, fieldId |
| 14 | tblFoodCategory | The table contains the food category's information  - Primary keys: categoryFoodId |

# II. Code Designs

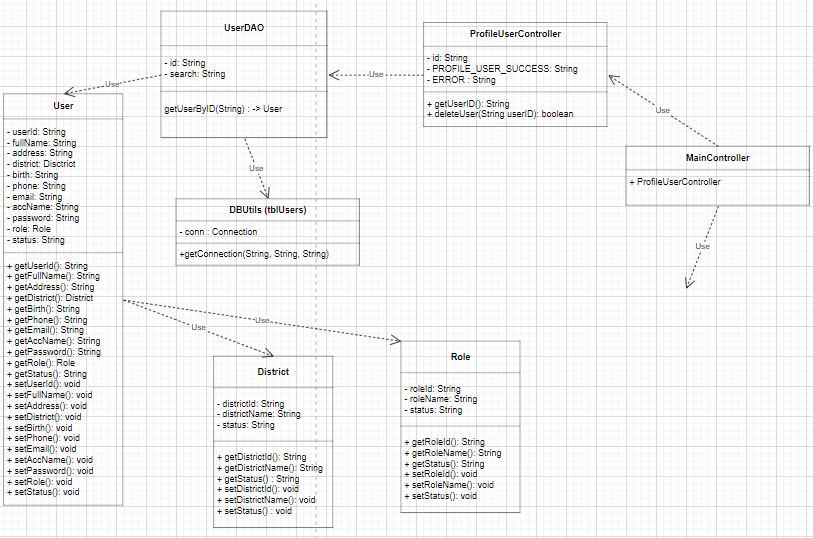
## 1. <Feature/Function Name1>

*[Provide the detailed design for the function <Feature/Function Name1>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

1.<View contact information/View contact information> :It include

2.<View home page/View landing page>

3.<Profile of user/View account’s profile>



4.<Update profile of user/View account’s profile>

5.<Login (google)/Authentication>

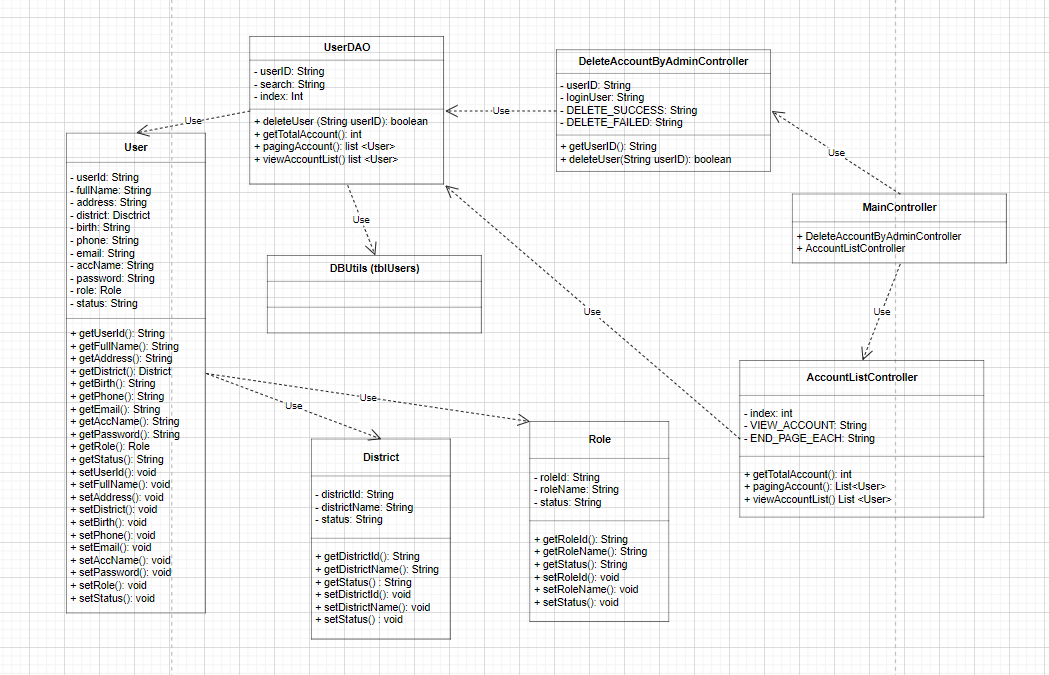
6.<Create account for Customer/Sign up account>

7.<Forgot Password/Reset password>

8.<View admin page/View admin page>

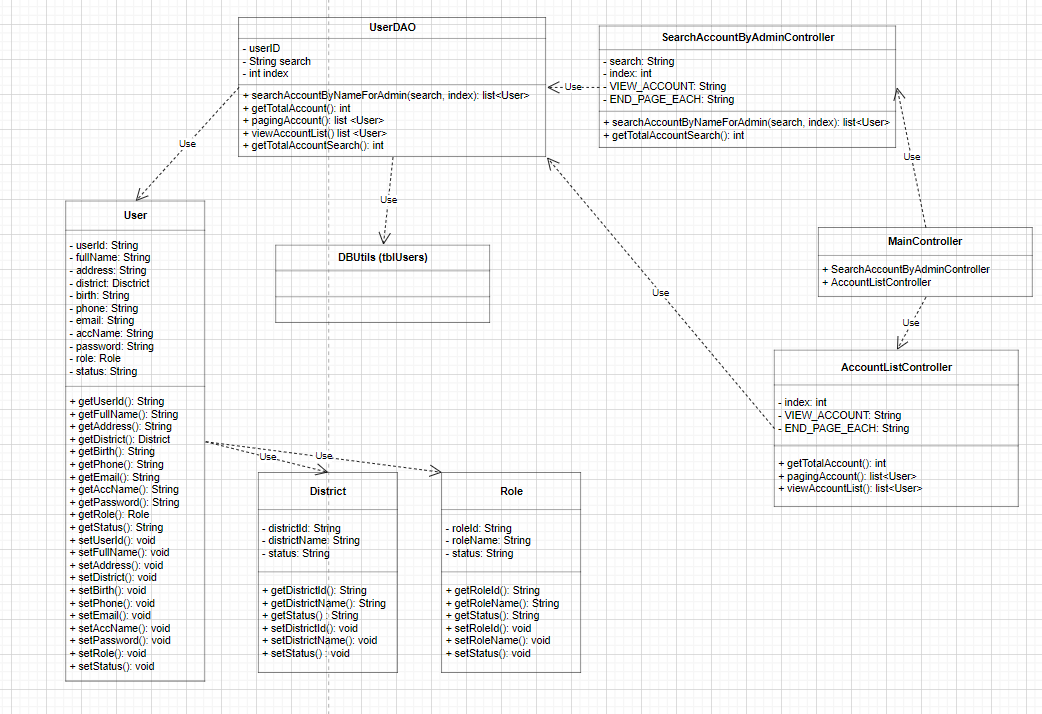
9.<Delete account/Manage list of accounts>

1. Class Diagram



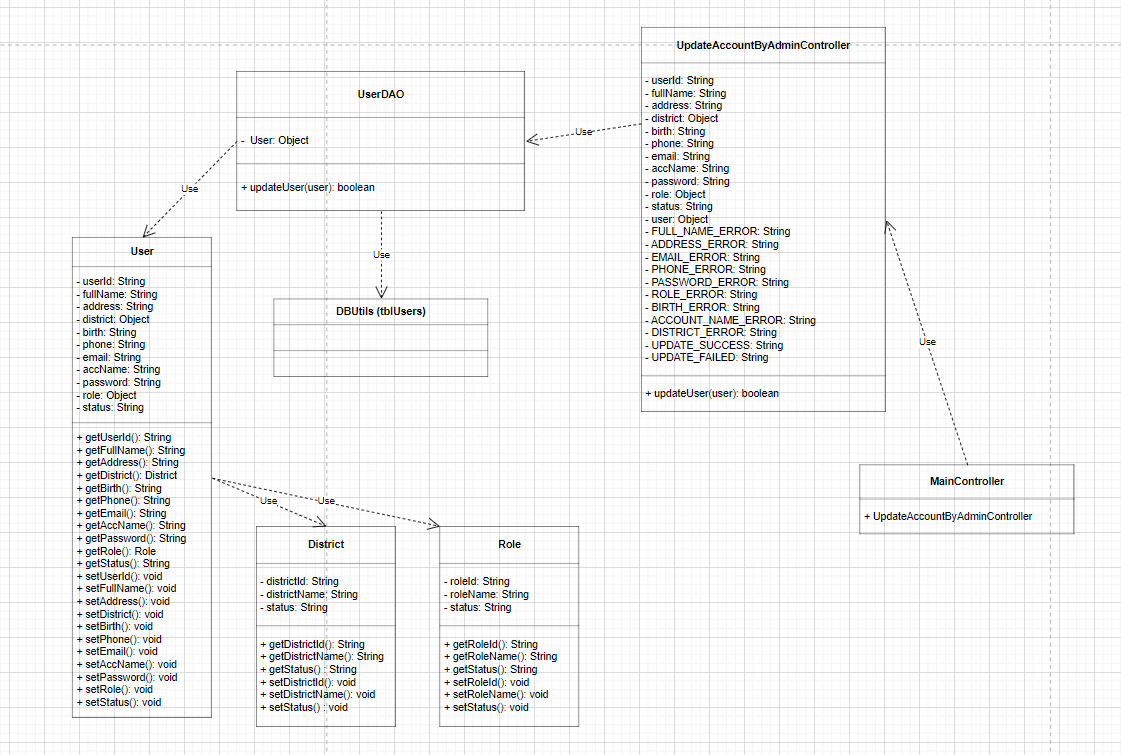
10.<Search account/Manage list of accounts>

1. Class Diagram



11.<Update account/Manage list of accounts>

1. Class Diagram



12.<Update status Field/Manage list of fields>

13.<Search Field/Manage list of fields>

14.<Delete Field/Manage list of fields>

15.<Search info history Booking field/Manage Booking field>

16.<Detail info history booking/Manage Booking field>

17.<View Screen Customer/View Screen Customer>

18.<Search Active Field/View list of field>

19.<View detail of Field/View field detail>

20.<Feedback/View field detail>

21.<View Similar Field/View field detail>

22.<Update Order Field/Booking field>

23.<Delete Order Field/Booking field>

24.<Create information Order Field/Booking field>

25.<Booking confirm/Booking field>

26.<View QR code/Pay money>

27.<Check out/Pay money>

28.<View all Order of customer/View history of booking>

29.<Search order of customer/View history of booking>

30.<View detail booked fields/View detail history of booking>

31.<View list field of Manager/Manage list owner’s field>

32.<Search field of manager/Manage list owner’s field>

33.<Sort Field of manager/Manage list owner’s field>

34.<Delete Field of manager/Manage list owner’s field>

35.<Create information of new Field/Create new field>

36.<View information detail of Field/Manage owner’s field detail>

37.<View all booking field of customer/ View booking history of owner’s field>

38.<Search booked Field/View booking history of owner’s field>

39.<Delete Customer Booking/View booking history of owner’s field>

40.<View detail booking/View booking history of owner’s field>

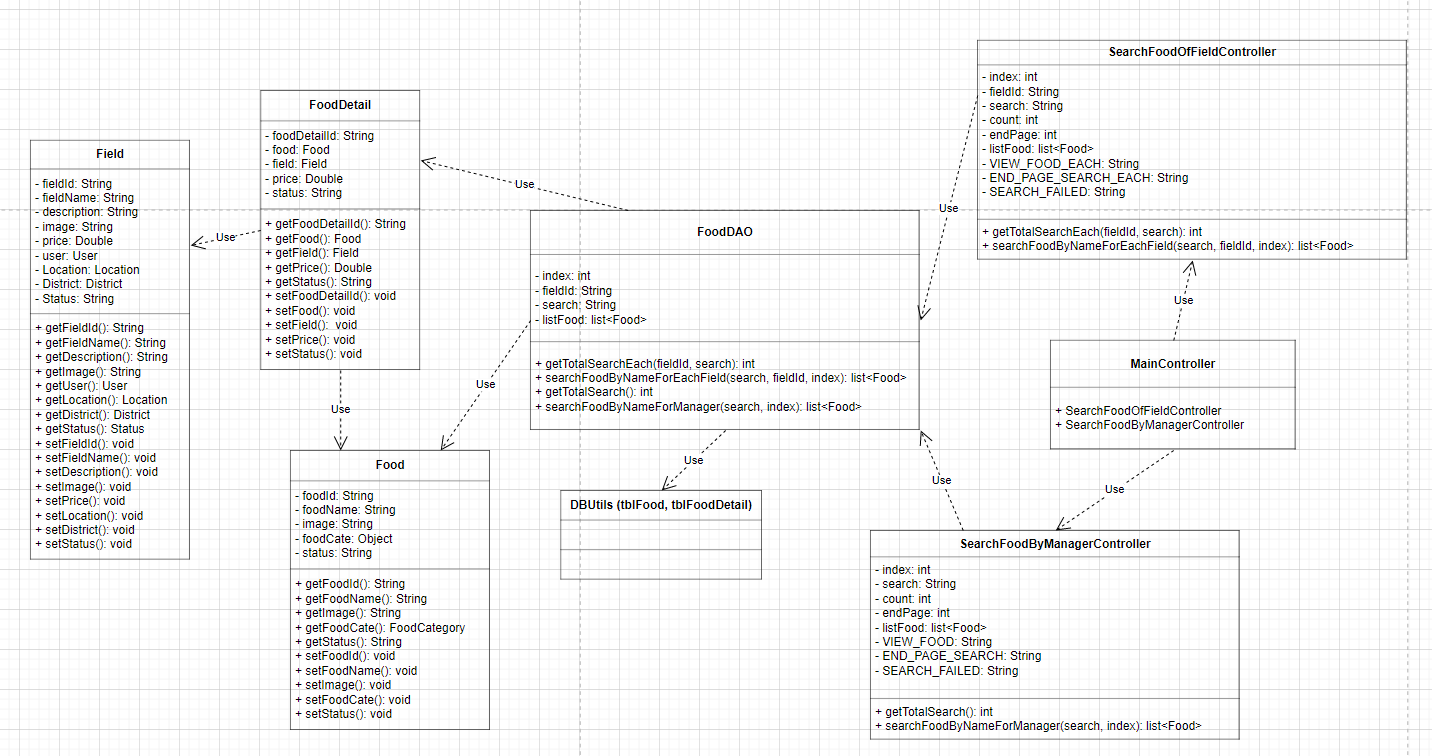
41.<Search order /Manage order >

42.<Delete Order /Manage order >

43.<Income of all order/Manage order >

44.<Search Food/Manage food>

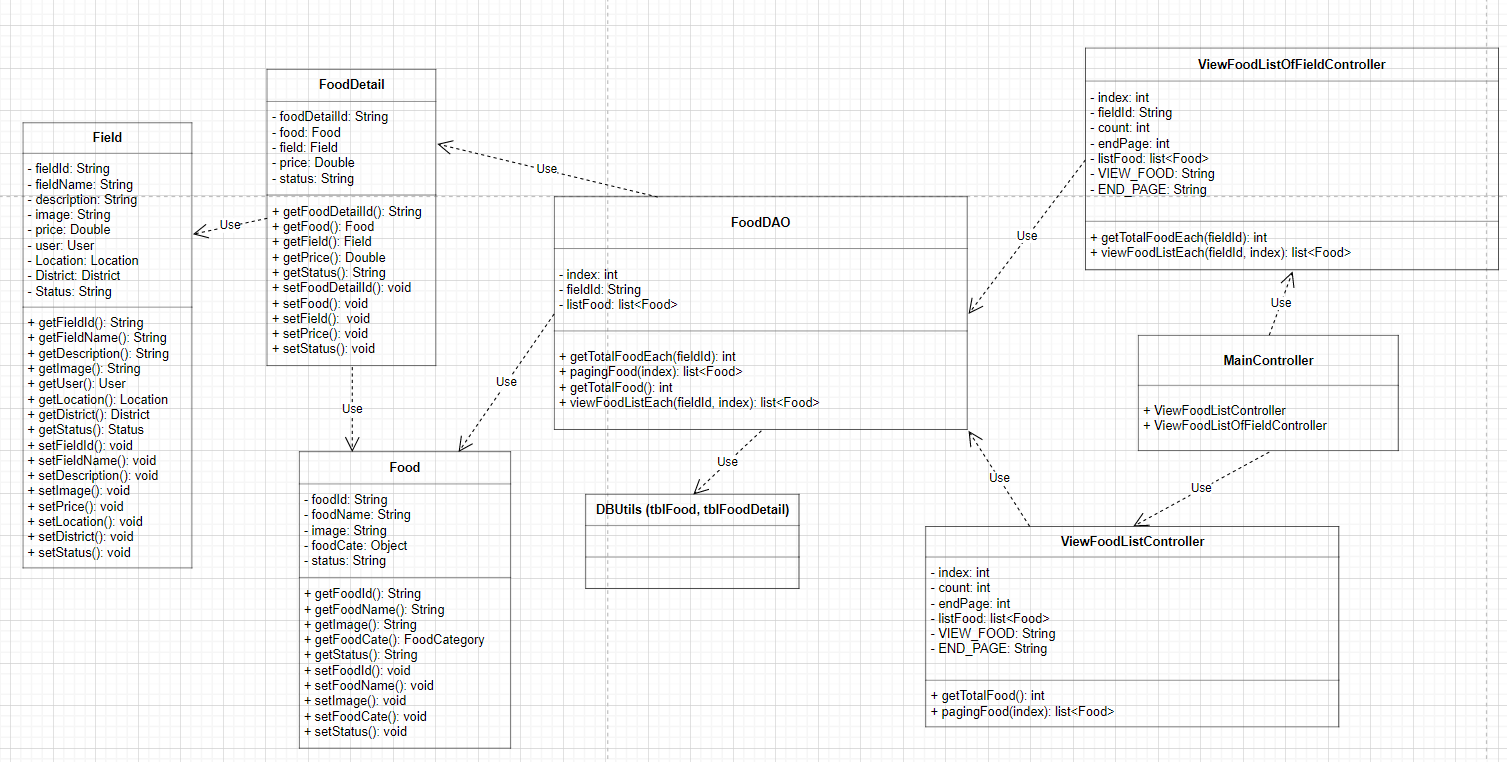
1. Class Diagram:



45.<Delete Food/Manage food>

46.<View Food/Manage food>

1. Class Diagram:



47.<Update Food/Manage food>

48.<Create Food/Manage food>

49.<Change password/Manage account>: It include controllers (MainController, ResetPasswordController) , daos(UserDAO), dto(user)

### a. Class Diagram

*[This part presents the class diagram for the relevant feature]*



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

*[Provide the sequence diagram(s) for the feature, see the sample below]*



### d. Database queries

**UserDAO:**

CHECK\_DUPLICATE = "SELECT fullName FROM tblUsers WHERE userID = ?";

CREATE = "INSERT INTO tblUsers(userID, fullName, cityId, birthday, phone, accName,email, roleId, status, password) VALUES (? ,?, ?, ?, ?, ?, ?, ?, ?, ?)";

LOGIN = "SELECT userID, fullName, address, cityId, birthday, phone, accName, roleID, status FROM tblUsers WHERE email = ? AND password = ?";

CHECK\_PASS = "SELECT email, password FROM tblUsers WHERE email = ? AND password = ?";

UPDATE\_PASS = "UPDATE tblUsers SET password = ? WHERE email = ? AND password = ?";

SEARCH\_ACCOUNT\_BY\_NAME\_FOR\_ADMIN = "SELECT userID, fullName, address, cityId, birthday, phone, email, accName, password, roleId, status FROM tblUsers WHERE fullName LIKE ? ";

SEARCH\_ACCOUNT\_BY\_ID\_FOR\_ADMIN = "SELECT userID, fullName, address, cityId, birthday, phone, email, accName, password, roleId, status FROM tblUsers WHERE userID LIKE ? ";

DELETE\_USER = "UPDATE tblUsers SET status = 0 WHERE userID = ?";

UPDATE\_USER = "UPDATE tblUsers SET fullName = ?, address = ?, cityId = ?, birthday = ?, phone = ?, email = ?, accName = ?, password = ?, roleId = ?, status = ? WHERE userID = ?";

GET\_USER\_BY\_ID = "SELECT userID, fullName, address, cityId, birthday, phone, email, accName, status, roleID FROM tblUsers WHERE userID = ?";

GET\_ALL\_USER = "SELECT userID, fullName, address, cityId, birthday, phone, email, accName, status, roleID FROM tblUsers";

CHECK\_USER\_ID = "SELECT userID FROM tblUsers WHERE userID = ?";

UPDATE\_PROFILE\_USER = "UPDATE tblUsers SET fullName = ?, birthday = ?, phone = ?, email = ?, address = ? WHERE userID = ?”;

VIEW\_ACCOUNT\_LIST = "SELECT userID, fullName, address, cityId, birthday, phone, email, accName, password, roleId, status FROM tblUsers";

CHECK\_EMAIL\_EXISTED = "SELECT email FROM tblUsers WHERE email = ?";

PAGING = "SELECT \* FROM tblUsers ORDER BY userId OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY;

**SlotDetailDAO:**

GET\_ALL\_INFO = "SELECT slotDetailID, slotID, fieldID, status FROM tblSlotDetail WHERE slotDetailID like ? ";

**SlotDAO:**

GET\_ALL\_INFO = "SELECT slotID, timeStart, timeEnd, status FROM tblSlots WHERE slotID like ?";

COUNT\_ALL\_SLOT = "SELECT COUNT(\*) as totalSlot FROM tblSlots WHERE slotID like ? ";

PAGING\_LIST\_BOOKING = "SELECT \* FROM tblSlots WHERE slotID like ? ORDER BY slotID OFFSET ? ROWS FETCH NEXT 12 ROWS ONLY ";

**RoleDAO:**

GET\_ROLE = "SELECT roleID FROM tblRoles WHERE roleID = ?";

GET\_ROLE\_BY\_ID = "SELECT roleID, roleName, status FROM tblRoles WHERE roleID = ?";

GET\_ALL\_ROLE = "SELECT roleID, roleName, status FROM tblRoles";

**LocationDAO:**

GET\_LOCATION\_BY\_ID = "SELECT locationID, locationName, status FROM tblLocation WHERE locationID like ?";

GET\_ALL\_LOCATION = "SELECT locationID, locationName, status FROM tblLocation";

CHECK\_LOCATION\_ID = "SELECT locationID FROM tblLocation WHERE locationID = ?";

CHECK\_LOCATION\_NAME = "SELECT locationName FROM tblLocation WHERE locationName = ?";

SEARCH\_LOCATION\_BY\_ADMIN = "SELECT locationID, locationName, status FROM tblLocation WHERE locationName like ? AND status like ?";

DELETE\_LOCATION\_BY\_ADMIN = "UPDATE tblLocation SET [status] = 'false' WHERE locationID = ?";

CHECK\_EXIST\_LOCATION = "SELECT locationID FROM tblFields WHERE locationID = ?";

UPDATE\_STATUS\_LOCATION\_BY\_ADMIN = "UPDATE tblLocation SET [status] = ? WHERE locationID = ?";

UPDATE\_LOCATION\_BY\_OWNER = "UPDATE tblLocation SET locationName = ? WHERE locationID = ?";

CREATE\_LOCATION = "INSERT INTO tblLocation(locationID, locationName) VALUES(?,?)";

**BookingDAO:**

INSERT\_BOOKING ="INSERT INTO tblBooking(bookingId, bookingDate, userId, totalprice, [status]) VALUES (?, ?, ?, ?, ?)";

GET\_BOOKING\_BY\_BOOKING\_ID = "SELECT bookingID, bookingDate, userID, totalPrice, status FROM tblBooking WHERE bookingID like ? ";

DELETE\_BOOKING\_BY\_BOOKING\_ID = "UPDATE tblBooking SET status = ? WHERE bookingID like ? ";

UPDATE\_BOOKING\_STATUS\_BY\_ID = "UPDATE tblBooking SET status = ? WHERE bookingID = ? ";

GET\_BOOKING\_ID = "Select bookingID from tblBooking WHERE bookingID = ?";

COUNT\_ALL\_BOOKING = "SELECT COUNT(\*) as totalBooking FROM tblBooking WHERE userID = ? AND status like ? AND bookingDate BETWEEN ? AND ? ";

COUNT\_ALL\_BOOKING\_ADMIN = "SELECT COUNT(\*) as totalBooking FROM tblBooking WHERE userID like ? AND status like ? AND bookingDate BETWEEN ? AND ? ";

PAGING\_LIST\_ALL\_BOOKING = "SELECT \* FROM tblBooking WHERE userID = ? AND status like ? ORDER BY status DESC, bookingDate DESC OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY ";

PAGING\_LIST\_ALL\_BOOKING\_ADMIN = "SELECT \* FROM tblBooking WHERE userID like ? AND status like ? ORDER BY status DESC, bookingDate DESC OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY ";

COUNT\_LIST\_BOOKING = "SELECT COUNT(\*) as totalBooking FROM tblBooking WHERE userID = ? AND status like ? ";

COUNT\_LIST\_BOOKING\_ADMIN = "SELECT COUNT(\*) as totalBooking FROM tblBooking WHERE userID like ? AND status like ? ";

PAGING\_LIST\_BOOKING = "SELECT \* FROM tblBooking WHERE userID = ? AND status like ? AND bookingDate BETWEEN ? AND ? ORDER BY bookingId OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY ";

PAGING\_LIST\_BOOKING\_ADMIN = "SELECT \* FROM tblBooking WHERE userID like ? AND status like ? AND bookingDate BETWEEN ? AND ? ORDER BY bookingId OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY ";

PAGING\_LIST\_ALL\_BOOKING\_MANAGER = "SELECT \* FROM tblBooking WHERE bookingID = ? ORDER BY bookingId OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY ";

PAGING\_LIST\_BOOKING\_MANAGER = "SELECT \* FROM tblBooking WHERE bookingID = ? AND status like ? AND bookingDate BETWEEN ? AND ? ORDER BY bookingId OFFSET ? ROWS FETCH NEXT 10 ROWS ONLY ";

**BookingDetailDAO:**

GET\_BOOKING\_DETAIL = "SELECT bookingDetailID, bookingID, fieldID, slotDetailID, fieldPrice, playDate, status FROM tblBookingDetail WHERE bookingID like ? ";

GET\_LIST\_BOOKING\_DETAIL\_BY\_FIELD\_ID = "SELECT bookingDetailID, bookingID, fieldID, slotDetailID, fieldPrice, playDate, status FROM tblBookingDetail WHERE fieldID = ? AND playDate >= ? AND slotDetailID = ? ";

GET\_LIST\_BOOKING\_DETAIL\_BY\_FIELD\_ID\_2 = "SELECT bookingDetailID, bookingID, fieldID, slotDetailID, fieldPrice, playDate, status FROM tblBookingDetail WHERE fieldID = ? ";

INSERT\_BOOKING\_DETAIL = "INSERT INTO tblBookingDetail(bookingDetailId, bookingId, fieldId, playDate, slotDetailId, fieldPrice, [status]) VALUES (?, ?, ?, ?, ?, ?, 1)";

GET\_ALL\_BOOKING\_DETAIL = "SELECT bookingDetailID, bookingID, fieldID, playDate, slotDetailID, fieldPrice, status FROM tblBookingDetail";

COUNT\_ALL\_BOOKING\_DETAIL = "SELECT COUNT(\*) as totalBookingDetail FROM tblBookingDetail";

GET\_ALL\_BOOKING\_DETAIL\_PAGING = "SELECT bookingDetailID, bookingID, fieldID, playDate, slotDetailID, fieldPrice, status FROM tblBookingDetail ORDER BY bookingDetailID OFFSET ? ROWS FETCH NEXT 5 ROWS ONLY";

GET\_BOOKING\_DETAIL\_ID = "SELECT bookingDetailI tblBookingDetail WHERE bookingDetailID like ? ";

**FoodDetailDAO:**

GET\_ALL\_INFO = "SELECT foodDetailID, foodID, fieldID, price, quantity, status FROM tblFoodDetail WHERE foodDetailID like ?";

**FoodDAO:**

GET\_ALL\_INFO = "SELECT foodID, foodName, image, categoryFoodId, status FROM tblFoods WHERE foodID like ?";

SEARCH\_FOOD\_BY\_NAME\_FOR\_MANAGER = "SELECT foodId, foodName, image, categoryFoodId, status FROM tblFoods WHERE foodName LIKE ? ";

SEARCH\_FOOD\_BY\_ID\_FOR\_MANAGER = "SELECT foodId, foodName, image, categoryFoodId, status FROM tblFoods WHERE foodId LIKE ? ";

DELETE\_FOOD = "UPDATE tblFoods SET status = 0 WHERE foodId = ?";

VIEW\_FOOD\_LIST = "SELECT foodId, foodName, image, categoryFoodId, status FROM tblFoods";

UPDATE\_FOOD = "UPDATE tblFoods SET foodName = ?, image = ?, categoryFoodId = ?, status = ? WHERE foodId = ?";

COUNT = "select count(\*) from tblFoods";

PAGING = "SELECT \* FROM tblFoods ORDER BY foodId OFFSET ? ROWS FETCH NEXT 5 ROWS ONLY";

**FoodCategoryDAO:**

GET\_ALL\_INFO = "SELECT categoryFoodID, categoryFoodName, status FROM tblFoodCategory WHERE categoryFoodID like ?";

**FieldDAO:**

GET\_ALL\_INFO = "SELECT fieldID, fieldName, description, image, categoryFieldID, price, UserID, LocationID, cityID, status FROM tblFields WHERE fieldID like ? ";

PRINT\_FIELD\_DETAIL\_BY\_NAME = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, cityId, status FROM tblFields WHERE fieldName like ?";

PRINT\_ALL\_OWNER\_FIELD = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, cityId, status FROM tblFields WHERE userID like ? ";

PRINT\_ALL\_FIELD\_BY\_ADMIN = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, cityId, status FROM tblFields";

PRINT\_FIELD\_DETAIL\_BY\_ADMIN = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, cityId, status FROM tblFields WHERE fieldId like ?";

UPDATE\_STATUS\_FIELD\_BY\_ADMIN = "UPDATE tblFields SET [status] = ? WHERE fieldId = ?";

UPDATE\_FIELD\_BY\_OWNER = "UPDATE tblFields SET fieldName = ?, [description] = ?, [image] = ?, categoryFieldId = ?, price = ?, userId = ?, locationId = ?, cityId = ? WHERE fieldId = ?";

DELETE\_FIELD\_BY\_ADMIN = "UPDATE tblFields SET [status] = 'false' WHERE fieldId = ?";

CHECK\_EXIST\_FIELD = "SELECT fieldId FROM tblBookingDetail WHERE fieldId = ?";

SEARCH\_FIELD\_BY\_NAME = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, cityId, status FROM tblFields WHERE fieldName like ? AND status like ?";

SEARCH\_FIELD\_BY\_FIELD\_CATE = "SELECT fieldId, fieldName, description, image, f.categoryFieldId, price, userId, locationId, cityId, f.status, fc.categoryFieldName FROM tblFields f LEFT JOIN tblFieldCategory fc ON f.categoryFieldId = fc.categoryFieldId WHERE fc.categoryFieldName like ? AND f.status like ?";

SEARCH\_FIELD\_BY\_FIELD\_OWNER = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, f.userId, locationId, f.cityId, f.status FROM tblFields f LEFT JOIN tblUsers us ON f.userId = us.userId WHERE us.fullName like ? AND f.status like ?";

SEARCH\_FIELD\_BY\_CITY = "SELECT fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, f.cityId, f.status FROM tblFields f LEFT JOIN tblCity ci ON f.cityId = ci.cityId WHERE ci.cityName like ? AND f.status like ?";

CHECK\_FIELD\_ID = "SELECT fieldId FROM tblFields WHERE fieldId = ?";

CREATE\_FIELD = "INSERT INTO tblFields(fieldId, fieldName, description, image, categoryFieldId, price, userId, locationId, cityId) VALUES(?,?,?,?,?,?,?,?,?)";

**FieldCategoryDAO:**

GET\_ALL\_INFO = "SELECT categoryFieldID, categoryFieldName, status FROM tblFieldCategory WHERE categoryFieldId like ?";

GET\_ALL\_CATEGORY = "SELECT categoryFieldID, categoryFieldName, status FROM tblFieldCategory";

CHECK\_FIELD\_CATE\_ID = "SELECT categoryFieldID FROM tblFieldCategory WHERE categoryFieldID = ?";

CHECK\_FIELD\_CATE\_NAME = "SELECT categoryFieldName FROM tblFieldCategory WHERE categoryFieldName = ?";

SEARCH\_FIELD\_CATE\_BY\_ADMIN = "SELECT categoryFieldID, categoryFieldName, status FROM tblFieldCategory WHERE categoryFieldName like ? AND status like ?";

DELETE\_FIELD\_CATE\_BY\_ADMIN = "UPDATE tblFieldCategory SET [status] = 'false' WHERE categoryFieldID = ?";

CHECK\_EXIST\_FIELD\_CATE = "SELECT categoryFieldID FROM tblFields WHERE categoryFieldID = ?";

UPDATE\_STATUS\_FIELD\_CATE\_BY\_ADMIN = "UPDATE tblFieldCategory SET [status] = ? WHERE categoryFieldID = ?";

UPDATE\_FIELD\_CATE\_BY\_OWNER = "UPDATE tblFieldCategory SET categoryFieldName = ? WHERE categoryFieldID = ?";

CREATE\_FIELD\_CATE = "INSERT INTO tblFieldCategory(categoryFieldID, categoryFieldName) VALUES(?,?)";

**CityDAO:**

GET\_ALL\_INFO = "SELECT cityID, cityName, status FROM tblCity WHERE cityID = ?";

GET\_ALL\_CITY = "SELECT cityID, cityName, status FROM tblCity";

CHECK\_CITY\_ID = "SELECT cityID FROM tblCity WHERE cityID = ?";

CHECK\_CITY\_NAME = "SELECT cityName FROM tblCity WHERE cityName = ?";

SEARCH\_CITY\_BY\_ADMIN = "SELECT cityID, cityName, status FROM tblCity WHERE cityName like ? AND status like ?";

UPDATE\_STATUS\_CITY\_BY\_ADMIN = "UPDATE tblCity SET [status] = ? WHERE cityID = ?";

UPDATE\_CITY\_BY\_OWNER = "UPDATE tblCity SET cityName = ? WHERE cityID = ?";

DELETE\_CITY\_BY\_ADMIN = "UPDATE tblCity SET [status] = 'false' WHERE cityID = ?";

CHECK\_EXIST\_CITY = "SELECT cityID FROM tblFields WHERE cityID = ?";

CREATE\_CITY = "INSERT INTO tblCity(cityID, cityName) VALUES(?,?)";

## FeedbackDAO

CHECK\_FEEDBACK\_ID = "SELECT feedbackId from Feedback where feedbackId like ?";

INSERT\_FEEDBACK = "INSERT INTO tblFeedback(feedbackId, title, content, userId, fieldId, status) VALUES(?,?,?,?,?,?) ";

PRINT\_ALL\_FEEDBACK = "SELECT feedbackId, title, content, userId, fieldId, status FROM tblFeedback";

SEARCH\_FEEDBACK\_BY\_ADMIN = "SELECT feedbackId, title, content, us.userId, fi.fieldId, f.status" + " FROM tblFeedback f left join tblUsers us on f.userId = us.userId left join tblFields fi on f.fieldId = fi.fieldId " + " where us.fullName like ? AND fi.fieldName like ?";

**NewPassword**

## UPDATE\_PASSWORD = "UPDATE tblUsers SET password = ? WHERE email = ?";

## 2. <Feature/Function Name2>

…

# III. Database Tables

## 1. tblUsers

The table contains the user's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | userId | varchar | 10 | Yes | Yes | PK |  |
| 2 | fullName | nvarchar | 50 | No | Yes | No |  |
| 3 | address | nvarchar | 100 | No | No | No |  |
| 4 | cityId | varchar | 10 | No | No | FK |  |
| 5 | birthday | date | No | No | No | No |  |
| 6 | phone | varchar | 11 | No | No | No |  |
| 7 | email | varchar | 30 | No | Yes | No |  |
| 8 | accName | nvarchar | 50 | No | Yes | No |  |
| 9 | password | nvarchar | 50 | No | Yes | No |  |
| 10 | roleId | varchar | 2 | No | No | FK |  |
| 11 | status | bit | No | No | No | No |  |

## 2. TblRoles

The table contains the role's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | roleId | varchar | 10 | Yes | Yes | PK |  |
| 2 | roleName | nvarchar | 50 | No | Yes | No |  |
| 3 | status | bit | No | No | No | No |  |

## 3. TblFeedback

The table contains the feedback's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | feedbackId | varchar | 10 | Yes | Yes | PK |  |
| 2 | content | nvarchar | 500 | No | Yes | No |  |
| 3 | userId | varchar | 10 | No | Yes | FK |  |
| 4 | fieldId | varchar | 10 | No | No | FK |  |
| 5 | status | bit | No | No | No | No |  |

## 4. TblCity

The table contains the city's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | cityId | varchar | 10 | Yes | Yes | PK |  |
| 2 | cityName | nvarchar | 50 | No | Yes | No |  |
| 3 | status | bit | No | No | No | No |  |

## 5. TblLocation

The table contains the location's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | locationId | varchar | 10 | Yes | Yes | PK |  |
| 2 | locationName | nvarchar | 50 | No | Yes | No |  |
| 3 | status | bit | No | No | No | No |  |

## 6. TblFields

The table contains the field's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | fieldId | varchar | 10 | Yes | Yes | PK |  |
| 2 | fieldName | nvarchar | 50 | No | Yes | No |  |
| 3 | description | nvarchar | 500 | No | No | No |  |
| 4 | image | nvarchar | max | No | Yes | No |  |
| 5 | categoryFieldId | varchar | 10 | No | No | FK |  |
| 6 | price | money | No | No | Yes | No |  |
| 7 | userId | varchar | 10 | No | No | FK |  |
| 8 | locationId | varchar | 10 | No | No | FK |  |
| 9 | cityId | varchar | 10 | No | No | FK |  |
| 10 | status | bit | No | No | No | No |  |

## 7. TblFieldCategory

The table contains the field category's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | categoryFieldId | varchar | 10 | Yes | Yes | PK |  |
| 2 | categoryFieldName | nvarchar | 50 | No | Yes | No |  |
| 3 | status | bit | No | No | No | No |  |

## 8. TblSlots

The table contains the slot's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | slotId | varchar | 10 | Yes | Yes | PK |  |
| 2 | timeStart | varchar | No | No | Yes | No |  |
| 3 | timeEnd | varchar | No | No | Yes | No |  |
| 4 | status | bit | No | No | No | No |  |

## 9. TblSlotDetail

The table contains the slot detail's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | slotDetailId | varchar | 10 | Yes | Yes | PK |  |
| 2 | slotId | varchar | 10 | No | No | FK |  |
| 3 | fieldId | varchar | 10 | No | No | FK |  |
| 4 | status | bit | No | No | No | No |  |

## 10. TblFoodCategory

The table contains the food category's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | categoryFoodId | varchar | 10 | Yes | Yes | PK |  |
| 2 | categoryFoodName | nvarchar | 50 | No | Yes | No |  |
| 3 | status | bit | No | No | No | No |  |

## 11. TblFoods

The table contains the food's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | foodId | varchar | 10 | Yes | Yes | PK |  |
| 2 | foodName | nvarchar | 50 | No | Yes | No |  |
| 4 | image | nvarchar | max | No | Yes | No |  |
| 5 | categoryFoodId | varchar | No | No | No | FK |  |
| 6 | status | bit | No | No | No | No |  |

## 12. TblFoodDetail

The table contains the food detail's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | foodDetailId | varchar | 10 | Yes | Yes | PK |  |
| 2 | foodId | varchar | 50 | No | No | FK |  |
| 3 | fieldId | varchar | 10 | No | No | FK |  |
| 4 | price | money | No | No | Yes | No |  |
| 5 | status | bit | No | No | No | No |  |

## 13. TblBooking

The table contains the booking's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | bookingId | varchar | 10 | Yes | Yes | PK |  |
| 2 | bookingDate | date | No | No | Yes | No |  |
| 3 | userId | varchar | 10 | No | No | FK |  |
| 4 | totalPrice | money | No | No | No | No |  |
| 5 | status | bit | No | No | No | No |  |

## 14. TblBookingDetail

The table contains the booking detail's information.

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | bookingDetailId | varchar | 10 | Yes | Yes | PK |  |
| 2 | bookingId | varchar | 50 | No | Yes | FK |  |
| 3 | fieldId | varchar | 10 | No | No | FK |  |
| 4 | playDate | date | No | No | Yes | No |  |
| 5 | slotDetailId | varchar | 10 | No | No | FK |  |
| 6 | fieldPrice | money | No | No | Yes | No |  |
| 7 | status | bit | No | No | No | No |  |

# 