

**FPT ACADEMY INTERNATIONAL**

**FPT – APTECH COMPUTER EDUCATION**

**ACE-HCMC-2-FPT**

**590 Cach Mang Thang 8, District 3, Ho Chi Minh City, Viet Nam**



travelola

Project Report Semester 4

|  |  |  |
| --- | --- | --- |
| **Supervisor** | Ngo Phuoc Nguyen | |
| **Batch** | T1.2008.M0 | |
| **Group** | 4 | |
| **Serial No.** | **Student Name** | **Enrollment No** |
| 1. | Nguyen Ngoc Hung | Student1273007 |
| 2. | Phung Hoai Bao | Student1273009 |
| 3. | Nguyen The Long | Student1273008 |
| 4. | Pham Van Nguyen | Student127300 |

Contents

[**ACKNOWLEDGEMENT** 6](#_heading=h.gjdgxs)

[**REVIEW 1** 7](#_heading=h.30j0zll)

[**INTRODUCTION** 8](#_heading=h.1fob9te)

[**PROBLEM DEFINITION** 10](#_heading=h.3znysh7)

[**CUSTOMER REQUIREMENT SPECIFICATION** 12](#_heading=h.2et92p0)

[**SCOPE OF WORK** 15](#_heading=h.tyjcwt)

[**SYSTEM REQUIREMENT SPECIFICATION** 17](#_heading=h.3dy6vkm)

[**Hardware** 17](#_heading=h.1t3h5sf)

[Web Server 17](#_heading=h.4d34og8)

[Client 17](#_heading=h.2s8eyo1)

[**Software:** 17](#_heading=h.17dp8vu)

[Web Server 17](#_heading=h.3rdcrjn)

[Client 18](#_heading=h.26in1rg)

[**TASK SHEET REVIEW 1** 19](#_heading=h.lnxbz9)

[**REVIEW 2** 20](#_heading=h.35nkun2)

[**DATA FLOW DIAGRAM** 21](#_heading=h.1ksv4uv)

[Architecture and Design of the System: 21](#_heading=h.44sinio)

[I. Context Level Diagram: 22](#_heading=h.2jxsxqh)

[II. First Level DFD: 23](#_heading=h.z337ya)

[III. DFD for Booking Cycle: 27](#_heading=h.3j2qqm3)

[**Sequence diagram** 28](#_heading=h.1y810tw)

[I. Login Function: (Users,Supplier) 28](#_heading=h.4i7ojhp)

[II. Signup Function for customers: (Users,Supplier) 30](#_heading=h.2xcytpi)

[III. Searching data function: 32](#_heading=h.1ci93xb)

[IV. View data function: 34](#_heading=h.3whwml4)

[V. Insert/update/delete data function: 35](#_heading=h.2bn6wsx)

[VI. Booking data function 36](#_heading=h.qsh70q)

[I. Entity Relation Diagram 39](#_heading=h.3as4poj)

[II. Main entity description 41](#_heading=h.1pxezwc)

[**ENTITY RELATIONSHIP DIAGRAM** 50](#_heading=h.49x2ik5)

[**TABLE DESIGN** 51](#_heading=h.2p2csry)

[I. Table Users: 51](#_heading=h.147n2zr)

[II. Table Account: 52](#_heading=h.3o7alnk)

[III. Table Airline: 53](#_heading=h.23ckvvd)

[IV. Table Flight: 54](#_heading=h.ihv636)

[V. Table District: 56](#_heading=h.32hioqz)

[VI. Table Feed Back: 56](#_heading=h.1hmsyys)

[VII. Table Image: 57](#_heading=h.41mghml)

[VIII. Table Flight Booking: 57](#_heading=h.2grqrue)

[IX. Table Flight Booking Details: 58](#_heading=h.vx1227)

[X. Table Hotel: 59](#_heading=h.3fwokq0)

[XI. Table Room: 59](#_heading=h.1v1yuxt)

[XII. Table Hotel Booking: 60](#_heading=h.4f1mdlm)

[XIII. Table Hotel Booking Detail: 61](#_heading=h.2u6wntf)

[XIV. Table Hotel Booking Room: 61](#_heading=h.19c6y18)

[XV. Table Hotel Feedback: 61](#_heading=h.3tbugp1)

[XVI. Table Location: 62](#_heading=h.28h4qwu)

[XVII. Table Passenger: 62](#_heading=h.nmf14n)

[XVII. Table Province: 63](#_heading=h.37m2jsg)

[XVII. Table Ward: 63](#_heading=h.1mrcu09)

[**Site Maps** 64](#_heading=h.46r0co2)

[I. User site: 64](#_heading=h.2lwamvv)

[II. Supplier: 64](#_heading=h.111kx3o)

[III. Admin: 65](#_heading=h.3l18frh)

[**TASK SHEET REVIEW 2** 66](#_heading=h.4k668n3)

[**REVIEW 3** 67](#_heading=h.2zbgiuw)

[**GUI DESIGN** 68](#_heading=h.1egqt2p)

[**I.**](#_heading=h.3ygebqi) **Home Page:** 68

[**II.**](#_heading=h.2dlolyb) **Search Hotel page:** 69

[**III.**](#_heading=h.sqyw64) **Search Flight page:** 70

[**IV.**](#_heading=h.3cqmetx) **Search Combo Flight and Hotel:** 71

[**V.**](#_heading=h.1rvwp1q) **Hotel Detail page:** 73

[**VI.**](#_heading=h.4bvk7pj) **Hotel booking page:** 75

[**VII.** **Flight booking page:** 76](#_heading=h.2r0uhxc)

[**VIII. Complete booking page:** 77](#_heading=h.1664s55)

[**IX.**](#_heading=h.3q5sasy) **Login:** 78

[**X.**](#_heading=h.25b2l0r) **Register Page:** 79

[**XI.**](#_heading=h.kgcv8k) **Admin:** 80

[**TASK SHEET REVIEW 3** 86](#_heading=h.1jlao46)

# **ACKNOWLEDGEMENT**

Our group would like to devote this chapter to express our appreciation to all those who always offered us valuable assistance during in carrying out this project.

First and foremost, we owe a debt of gratitude to our dedicated instructor, Mr. Ngo Phuoc Nguyen, whose suggestions and encouragement helped us to overcome not only technological challenges but also psychological tension. Without him, our work could not have been completed successfully.

Secondly, we highly appreciate the project Team at the Head Office which has provided us a golden opportunity to apply what we learned into practice. We are certain that the experiences we gained from this project will facilitate our future career greatly. Besides, we would also like to express my gratitude to all the staff members of my center for not only providing me with the opportunity to work with them on this project, but also for their support and encouragement throughout the process.

Finally, we wish to say a personal thank you to all our dear classmates and team members who are always willing to support us openly and enthusiastically.

# **REVIEW 1**

# **INTRODUCTION**

During the period of information technology and computer development people can search and find anything at home vie internet connection and use online services. Therefore, we are developing a service “Book any flight, any hotel online with more conveniences” named Travel Booking App.

Travel Booking App is a group of people who wants to develop an online site where the users who are too busy to think of how they can purchase ticket online for traveling and this is also a marketing channel for hotels and airlines who will supply products for sale. It will provide users and suppliers the following features.

**For Travel Booking App’s customers:**

* Provide an interactive GUI, on the World Wide Web for the user.
* User can sign up, login, logout, change password, reset password on the website.
* User can search any flight, hotel and the web will show them the flight time available and the hotel room available they refer before they book.
* User can review the flight and hotel room that they booked and used.
* User can purchasing online vie payment online method as Pay-pal, stripe…
* User receive ticket with QR Scan code.
* User can see booking history.
* User can send feedback to admin or any question they have about Website or about their booking.
* User can see recommendation information.

**For Travel Booking App’s suppliers:**

* Airlines Suppliers can create, update, remove flights.
* Hotel Suppliers can create, update, remove rooms.
* Supplies can sign up, login, logout, change password, reset password on the website.
* Suppliers can send mail to any feedback or question from customers.
* Suppliers reply review on website.
* Suppliers can check total booking, total purchased money from customers.

# **PROBLEM DEFINITION**

After reading the project specifications of Travel Booking App, our team had discussed about what should to be done to fulfil their requirements properly and we came to conclusion that in order to achieve the best result, we should deal with following queries:

**From the view point of users**

1. How can customer book the flight?
2. How can customer book the hotel’s room?
3. How can customer find the flight and room they want?
4. How does customer keep track their history booking that they booked?
5. How can customers review the flight and room they used?
6. How can customers use login, signup, and logout?
7. How can customers make a payment and receive ticket?
8. How can customers send feedback to our website?

**From the view point of the Supplier**

1. How can airline suppliers create, update, delete flight?
2. How can hotel suppliers create, update, delete room?
3. How can suppliers see bookings history? And total payments?
4. How can suppliers use login, and logout?
5. How can suppliers send e-mail directly to customers in order to answer questions and send ticket confirmation?
6. How can suppliers receive payment?

**From the view point of the Administrator**

1. Will the systems be able to check the validity of users’ information before saving into the database?
2. How can system generate QR Scan code for ticket?
3. How can Admin build recommendation system as if customers choose flight’s destination, it will suggest hotels at that destination?
4. How can Admin receive payment from customer and send % payment to suppliers?
5. How can Admin check feedback and reply feedback to customers?
6. How can Admin reply review from customers?
7. How can Admin check total bookings made from customers?

**CUSTOMER REQUIREMENT SPECIFICATION**

**Client:** Travel Booking App

**Business/Project Objective:** The project is to design a service online for booking flights/hotels and recommend hotels/flights on same destination.

**Input to the system:**

**Travel Booking App’s admin need to provide:**

* Information required in order to create account to access to the admin system.

**Travel Booking App’s website suppliers need to provide:**

* Company information to create supplier account.
* Information about hotel or flight going to be shown on website.

**Travel Booking App’s website customers need to provide:**

* Personal information to create an account.

**Output from the system:**

**Travel Booking App’s admin will be able to:**

* Login/Logout to the admin system.
* Manage (view, search, update, and delete) hotel’s rooms.
* Manage (view, search, update, and delete) flights.
* Manage (view, search, active/ deactive) registered customers.
* Manage (view, search, active/ deactive) suppliers.
* Manage (view, search, update, delete) booking tickets.
* Manage (view, search, reply, delete) reviews.
* Manage (view, search, reply, delete) feedbacks.

**Travel Booking App’s suppliers will be able to:**

* Login, Logout, Change password, Reset password.
* View, edit their information, flight/room’s images.
* For hotels: Manage (view, search, update, and delete) hotel’s rooms.
* For airlines: Manage (view, search, update, and delete) flights.
* Manage (view, reply) about flight/hotel’s room reviews.
* View their rating.
* View total bookings from customers.
* Manage (view, send email) reply feedback from customer.

**Travel Booking App’s customers will be able to:**

* If customers are guests:
  + Search flight/hotel, destination or date they want to book.
  + View flight or hotel information.
  + View review, rating about airline and hotel.
  + Can sign up to become member.
* If customers are registered members:
* Login, Logout, change Password, reset Password.
* Book hotel, book flight.
* View booking history.
* Search hotel/flight by date, destination.
* Review hotel/flight and make rating.
* Change personal information.
* Make a payment online.
* Receive booking confirmation and QR Scan code ticket.
* View Recommendation product that system suggest base on destination, date available or high rating.

# **SCOPE OF WORK**

The application should perform the following functions:

**Login:**

* There are 3 types of account in the application (“admin”, “supplier” and “user”). Admin will login as “admin”, Supplier will login as “supplier” and Customer will login as “user”.
* Three accounts must be registered and login for using the functionality of the application.
* For “admin”: can view/update customer profile, supplier information.
* For “supplier”: can update their information, update images, change their password, reset password.
* For “user”: can update profile, change password, reset password, register new account.

**Booking flight/hotel’s room:**

* When users choose a flight/hotel’s room available, website process them to payment checkout view for purchasing.

**Review flight or hotel:**

* When user search for a flight or hotel, they can see reviews and rating about that flight or hotel.
* Users can only review when they joined a flight from a airline before or stay at the hotel before, too.

**Database**

* The information about flights, hotels, bookings, reviews will be stored in the database, and users can view on the Web or Mobile app

**Security:**

* The user password will be encrypted and stored in the database.

# **SYSTEM REQUIREMENT SPECIFICATION**

## **Hardware**

### Web Server

|  |  |
| --- | --- |
| Processor | Intel Core i5 or higher. |
| Memory | 4 GB RAM or greater. |
| Modem/ADSL | Connecting to internet 24/24 |

### Client

|  |  |
| --- | --- |
| Processor | Intel Core i3 or higher. |
| Memory | 2 GB RAM or greater. |
| Monitor | Super VGA (1024x768) or higher resolution |
| Modem/ADSL | Connecting to internet |

## **Software:**

### Web Server

|  |  |
| --- | --- |
| Operation System | Windows 7 or higher |
| Browser | Chrome version 35 or Microsoft Edge. |
| Database | SQL Server |
| Software | Internet Information Service |
| JAVA |
| Spring Boot |

### Client

|  |  |
| --- | --- |
| Operation System | Windows 7 or higher |
| Browser | Chrome version 35 or Microsoft Edge. |

**2.1 Development software**

* + Microsoft Windows 10
  + SQL Server
  + Google Chrome or Microsoft Edge
  + Git
  + Visual Studio Code
  + Postman

**2.2 Technology**

* Angular
* Javascript
* jQuery
* Flutter
* Firebase
* Bootstrap
* Java Spring boot, Spring Web.
* JWT security.

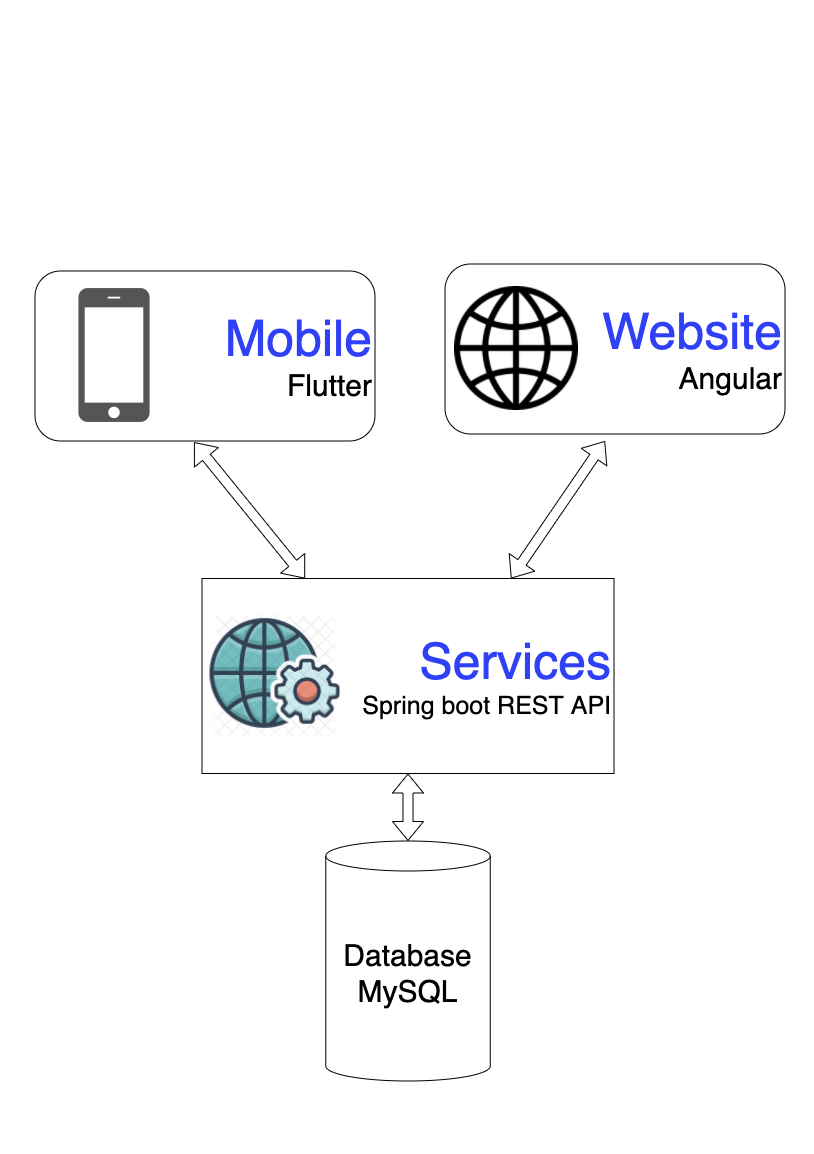
# **TASK SHEET REVIEW 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:**  **TRAVEL To VN** | | **Project Title:** | **Activity Plan Prepared By:** | **Date of Preparation of Activity Plan:** | | | |
| **Sr.No.** | **Task** | **Actual Start Date** | **Actual Days** | **Team Mate Names** | **Status** |
| 1 | Introduction and Problem Definition | travelola | Nguyen Ngoc Hung | 13/06/2022 | 1 | - Nguyen Ngoc Hung  - Phung Hoai Bao  - Nguyen The Long  - Pham Van Nguyen | 100% |
| 2 | Customer Requirement Specification (CRS) | 13/06/2022 | 1 | 100% |
| 3 | Functional Requirement | 13/06/2022 | 1 | 100% |
| 4 | System Requirement Specification | 100% |

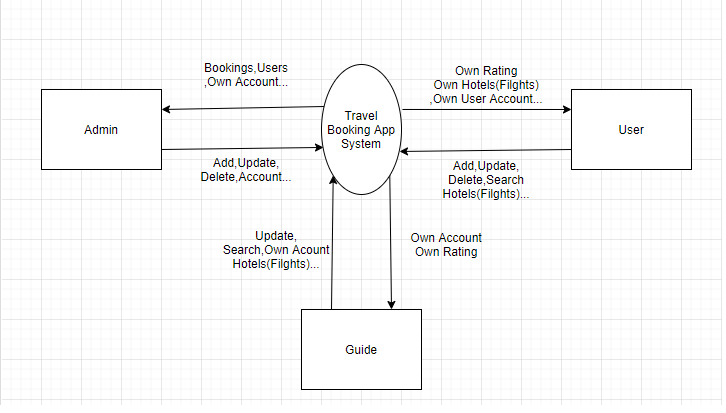
# **REVIEW 2**

# **DATA FLOW DIAGRAM**

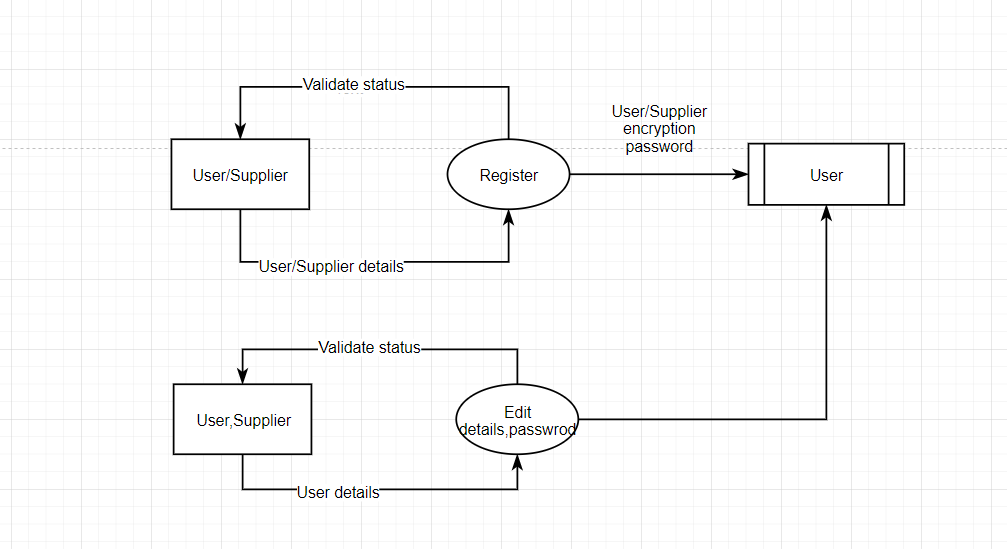
## Architecture and Design of the System:

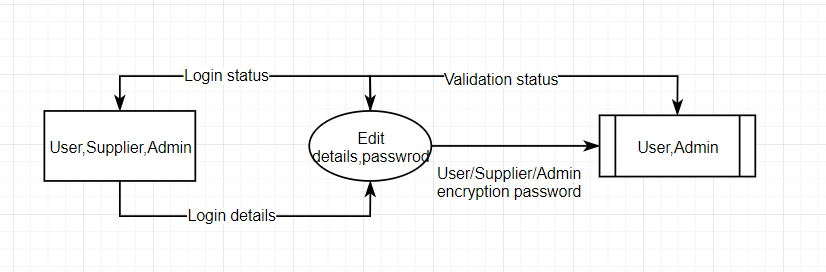


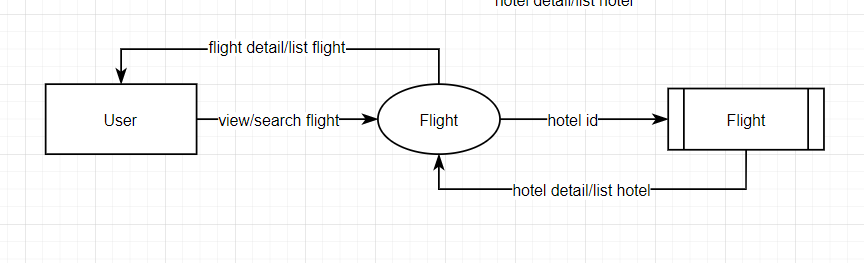
## Context Level Diagram:

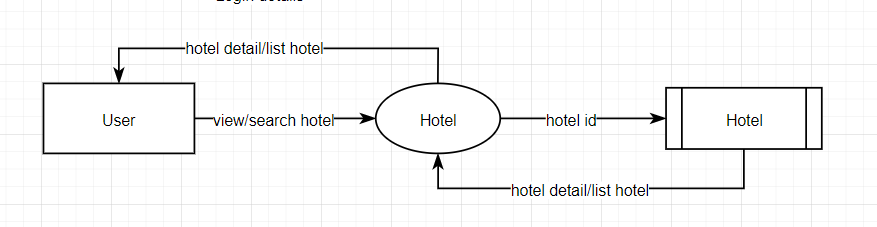


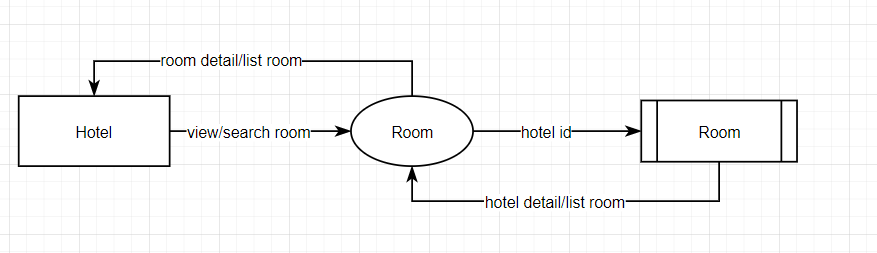
## First Level DFD:

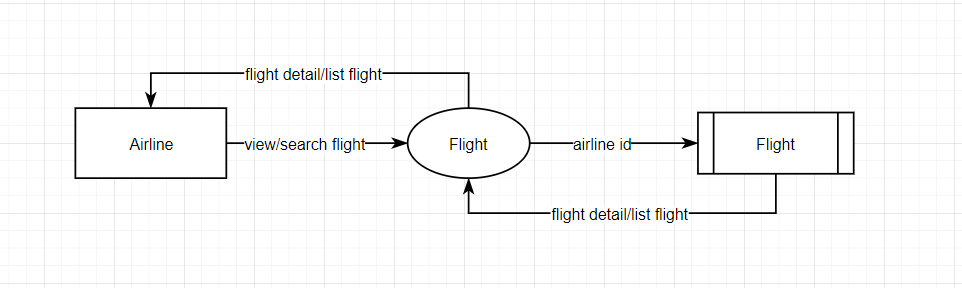


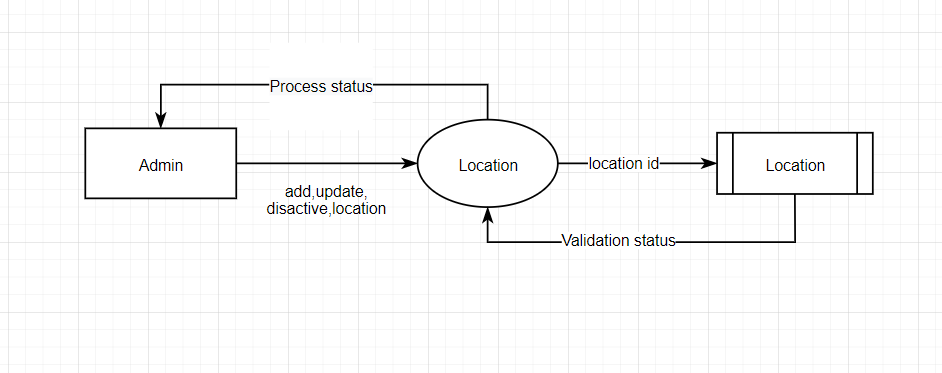
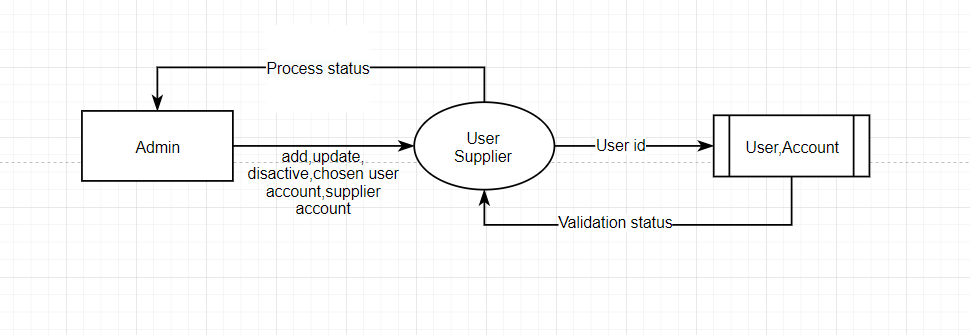
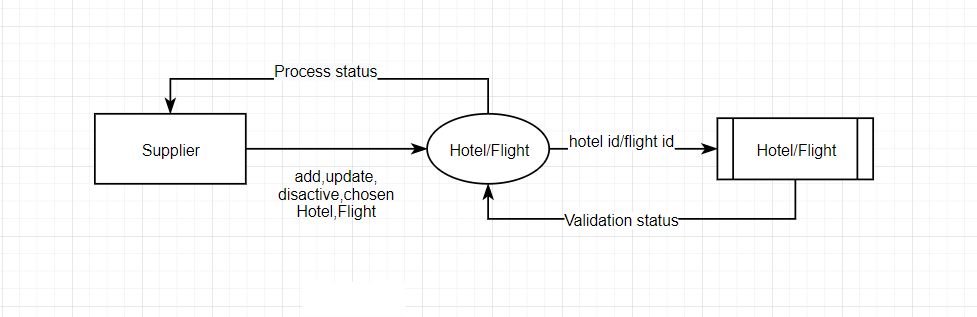








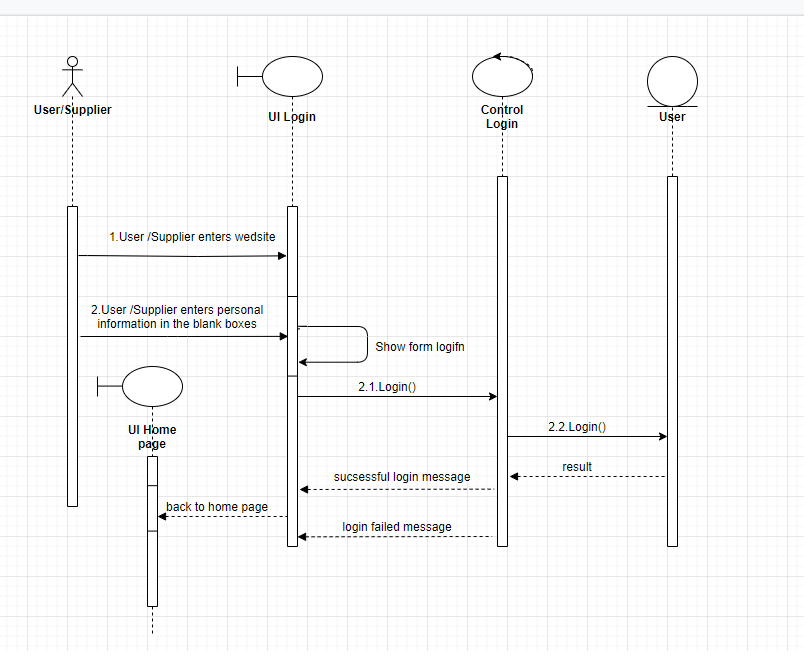




## DFD for Booking Cycle

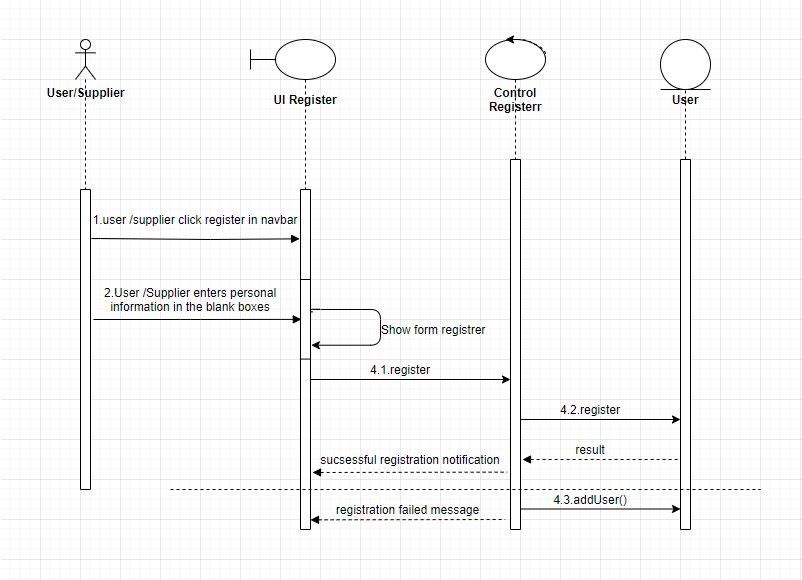
# **Sequence diagram**

## Login Function: (Users / Supplier)



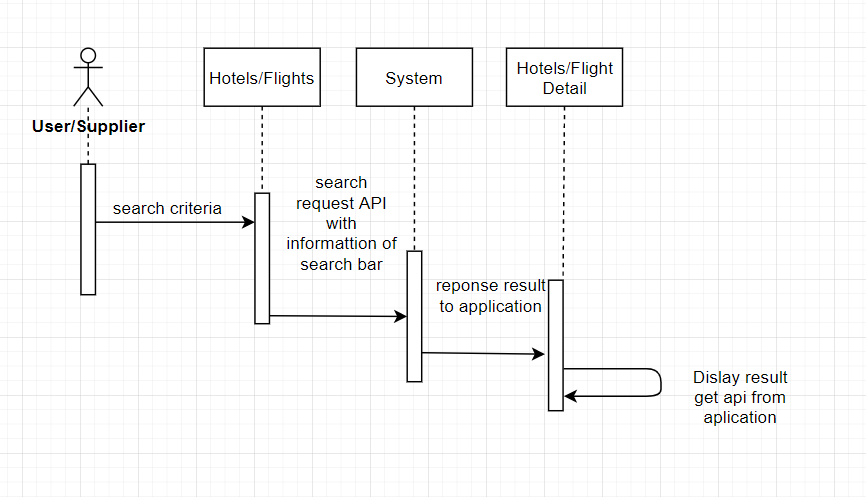
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Login – login button | |
| **Actors** | User and Supplier | |
| **Description** | When the user clicks the log button, Sparrow will open a popup page asking the user to enter login information | |
| **Requirements** | Username and Password can’t be blank. | |
|  | **User and Supplier Actions** | **System Responses** |
| **Basic flow** | 1. User and Supplier input user name and password | 2. System check the input string [Exception 1]  3. The system will check if, user exists or not and return the result  4.If yes, the screen will show login success, otherwise it will report invalid user. |
| **Exception** | 1. User or Supplier left the input Username empty | 1. An error message will be display: “Username can’t be blank” |
|  | 2. User or Supplier left the input Password empty | 1. An error message will be display: “Password can’t be blank” |

## Signup Function for customers: (User / Supplier)



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Register – Register button | |
| **Actors** | User and Supplier | |
| **Description** | When the user clicks register, it goes to the register page for the user to register, input register information | |
| **Requirements** | first name, last name, user name, email, password, confirm password can’t be blank. | |
|  | **User and Supplier Actions** | **System Responses** |
| **Basic flow** | 1. User and Supplier input first name, last name, user name, email, password, confirm password | 2. System check the input string [Exception 1]  3. The system will check if, user exists or not and return the result  4.If yes, the screen will show register success, otherwise it will report invalid user. |
| **Exception** | 1. User or Supplier left the input Username empty | 1. An error message will be display: “Username can’t be blank” |
|  | 2. User or Supplier left the input Password empty | 1. An error message will be display: “Password can’t be blank” |

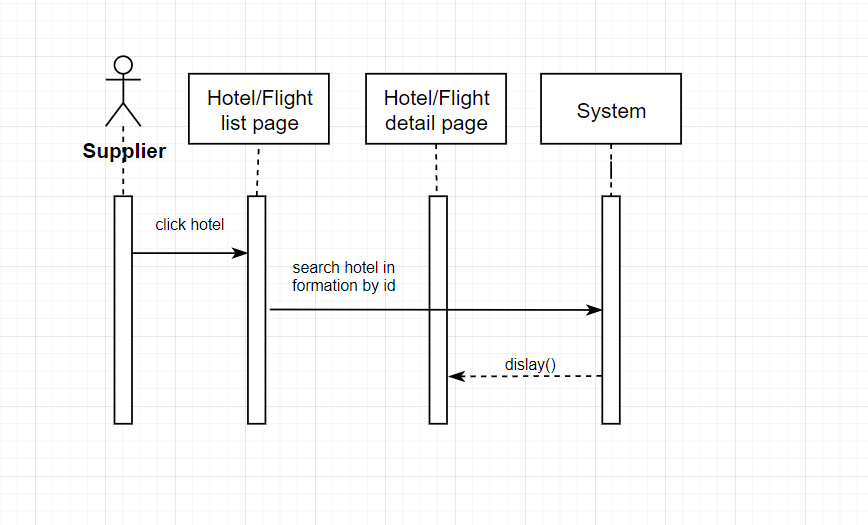
### Searching data function:



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Search – Search button | |
| **Actors** | User and Supplier | |
| **Description** | When search button is clicked, search result page will be displayed with result list of products and their information | |
| **Requirements** | Search string can’t be blank. | |
|  | **Actor Actions** | **System Responses** |
| **Basic flow** | 1. Actor input search string and input after that click on search button. | 2. System check the search string [Exception 1]  3. System create an empty list to contain search result.  4. Search in Products table to find products which have names contain the search string (ignore case) and add them to result list.  5. Display the result list in the search result page. |
| **Exception** | 1. Actor left the search string empty | 1. An error message will be display: “Search string can’t be empty” |

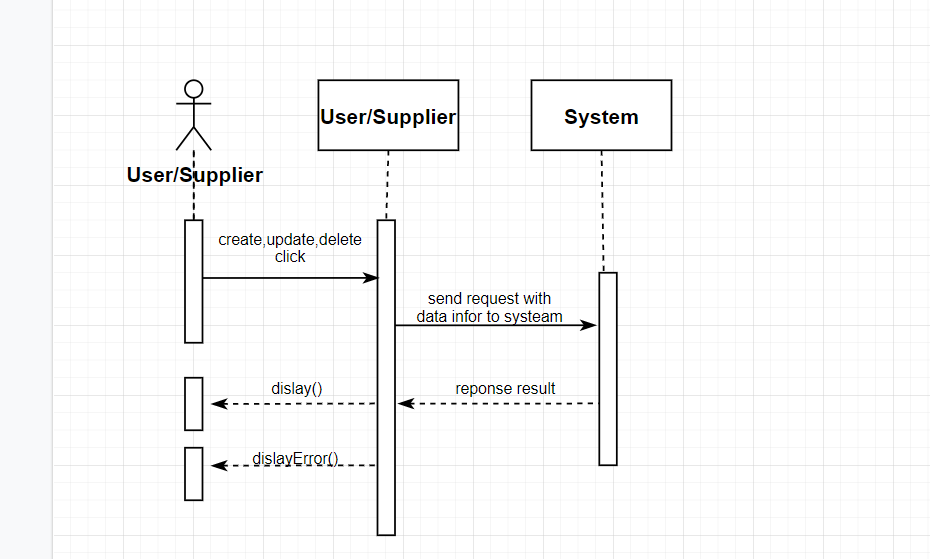
## 

## View data function:



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View supplier details | |
| **Actors** | User and Supplier | |
| **Description** | Allows users to view all hotels that match the user's criteria and have pagination for optimization | |
| **Requirements** | Hotels can click to go to the detail page to view hotel information | |
|  | **Actor Actions** | **System Responses** |
| **Basic flow** | 1. Clicks on supplier item | 2. Information search with the input data.  3. Display hotel in the hotel detail page. |

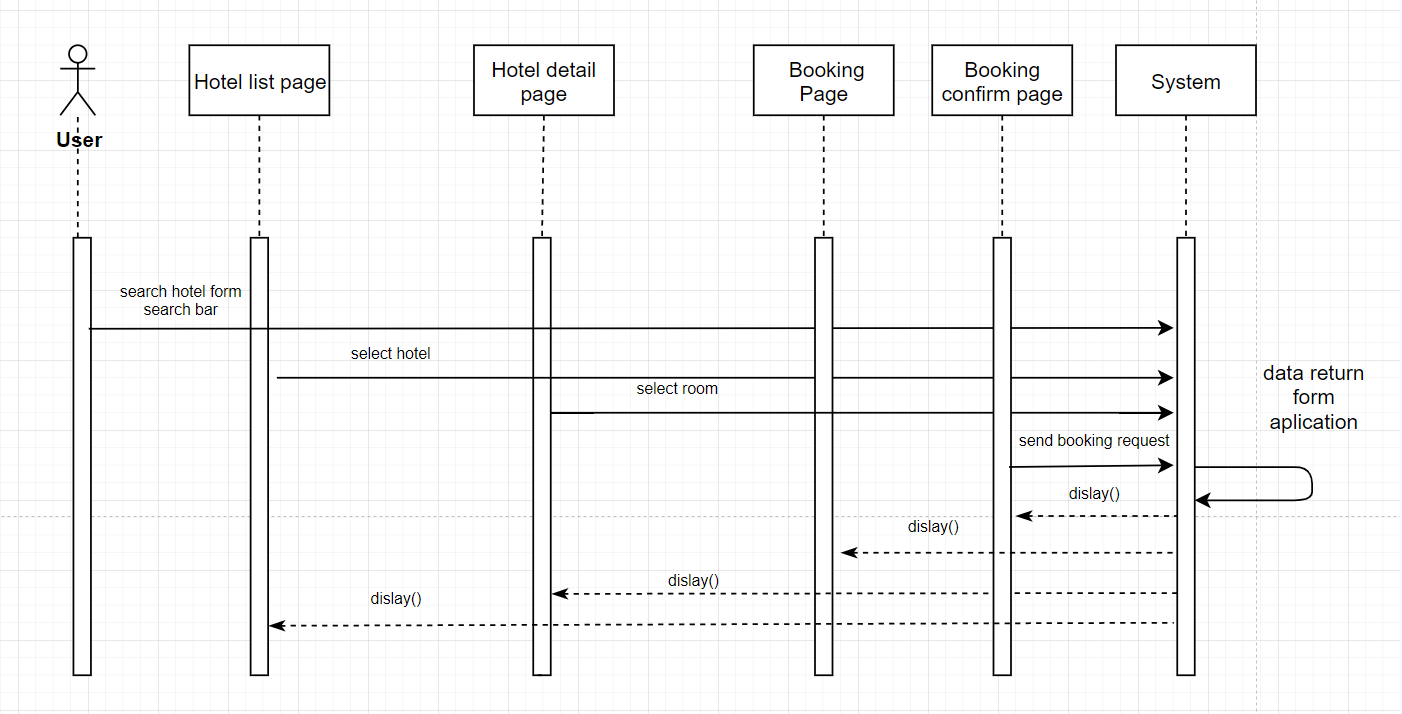
## Insert/update/delete data function:



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert, update, delete supplier details | |
| **Actors** | Supplier | |
| **Description** | Supplier can insert, update, delete their data here, in addition, there is also pagination and filter here. | |
| **Requirements** | Users can view old data as well as update and delete unnecessary data | |
|  | **Actor Actions** | **System Responses** |
| **Basic flow** | 1. Clicks on supplier item list button like edit or delete | 2.Get the data input and do action on the sever.  3. Check the action if it is created or update, then validate the require and columns to show the user, if it is deleted, the message will be displayed.  4. Display notification when success or error. |

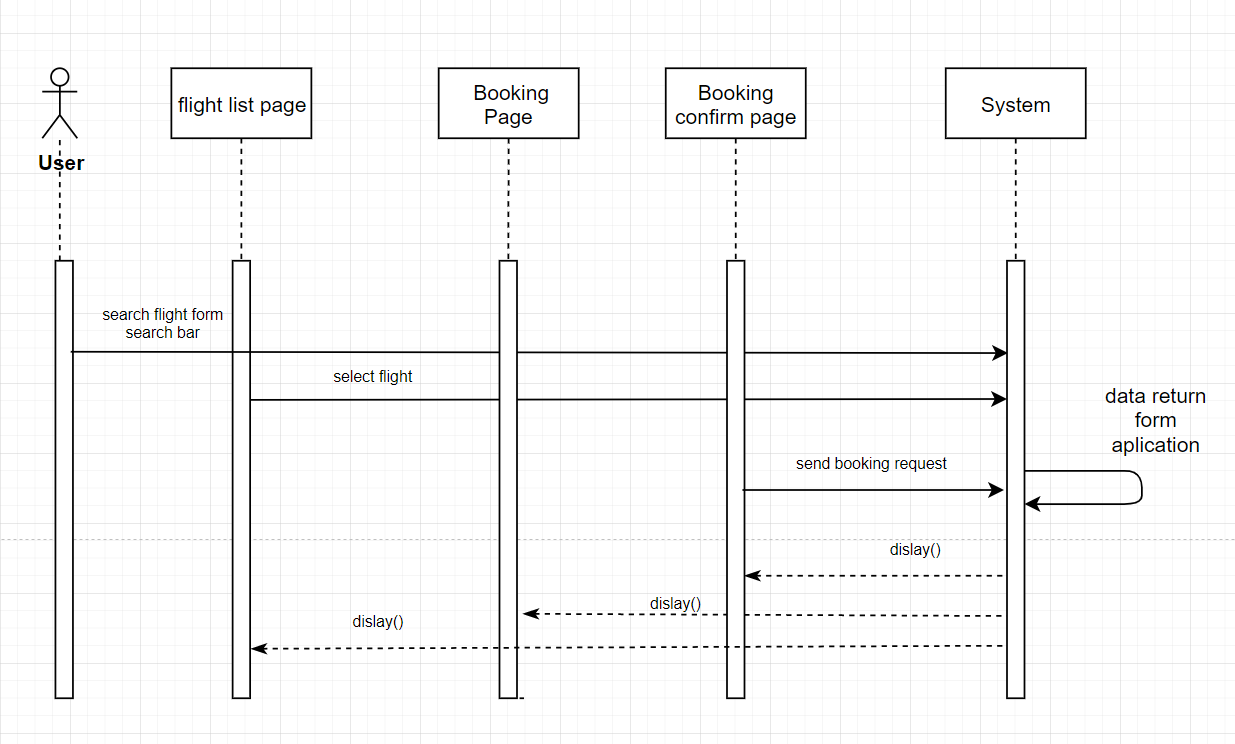
## Booking data function

1. Booking room



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Booking Room Action | |
| **Actors** | User only | |
| **Description** | Users can use the search engine on the home page to find the right hotel, then proceed to select the hotel and select the appropriate room to proceed with booking. | |
| **Requirements** | 1.Check in date, check out date and location can’t be blank.  2. must choose the number of rooms suitable for the number of people moving to the hotel.  3.Check out date greater than check in date.  4.Only user role can create booking | |
|  | **Actor Actions** | **System Responses** |
| **Basic flow** | 1. select location, check in date check out date, number adult and children to search  2.Into detail page click book now in room item select  3.into booking page click validate to validate information of user and open payment method | 1.Get the data input and do action search. Display item hotel result  2.Get data end search information of user. Display in Booking confirm page  3.Validate information of user and display payment method popup. After payment create book with data booking of user and show booking information at booking confirm page |

2. Booking flight

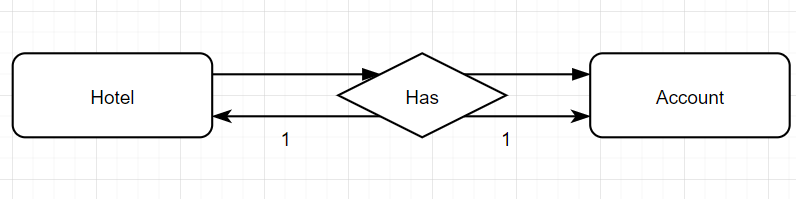


|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Booking Flight Action | |
| **Actors** | User only | |
| **Description** | Users can use the search engine on the home page to find the right flight Schedules, then proceed to select the schedules. | |
| **Requirements** | 1.Departure date and location can’t be blank.  2. Departure date less than Return date | |
|  | **Actor Actions** | **System Responses** |
| **Basic flow** | 1. Select location, departure date and search  2. Into detail page click select in flight item  3. Click add new passenger  4. Click remove passenger  5. Click travel with infant  6. Save personal info  7. Confirm booking | 1. Get the data input and do action search. Display item flight schedule result.  2. Get data and search data of user in the sever.  3. Add new passenger  4. Remove passenger  5. If get infant booking get more 10$  6. Save information of passenger  7. End request create booking |

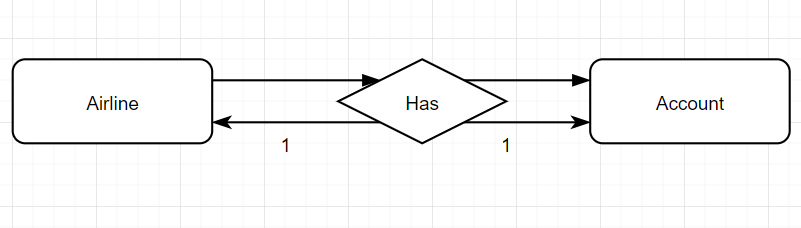
# **DATABASE DESIGN**

## Entity Relation Diagram

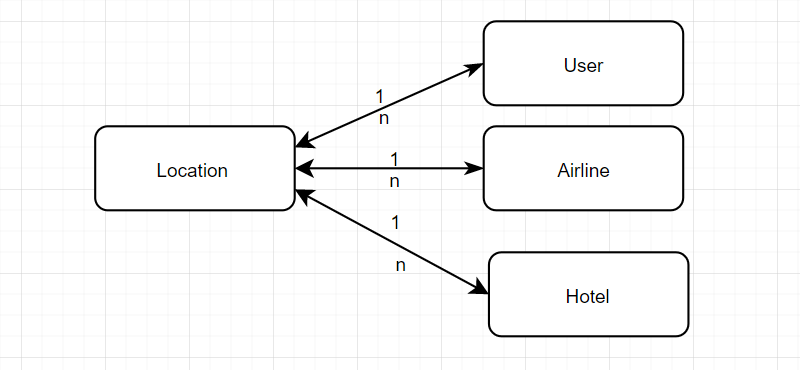
1. Hotel – Account



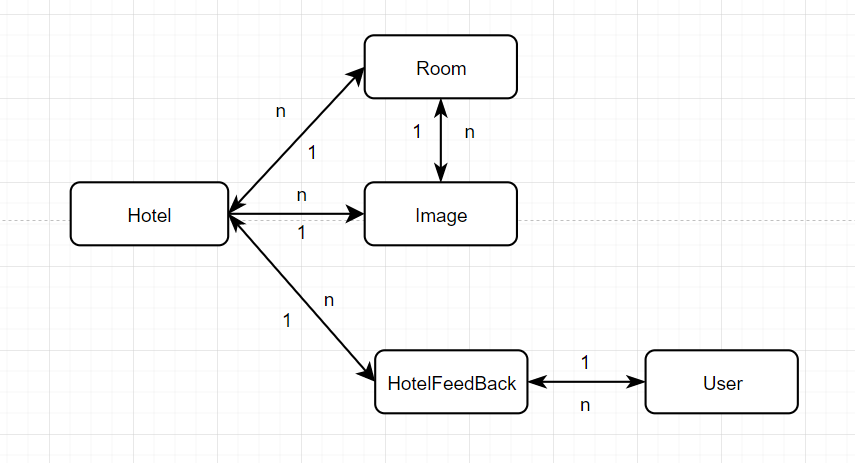
1. Airline – Account



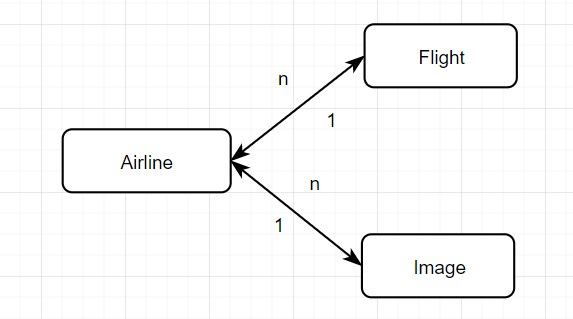
1. Location



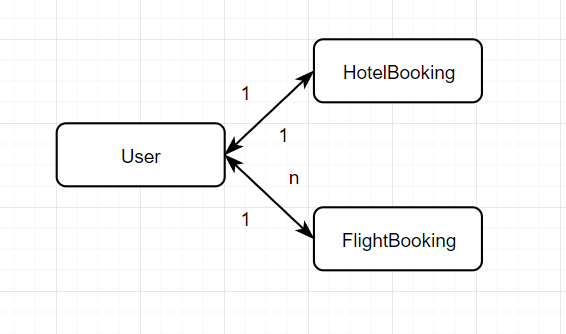
1. Hotel



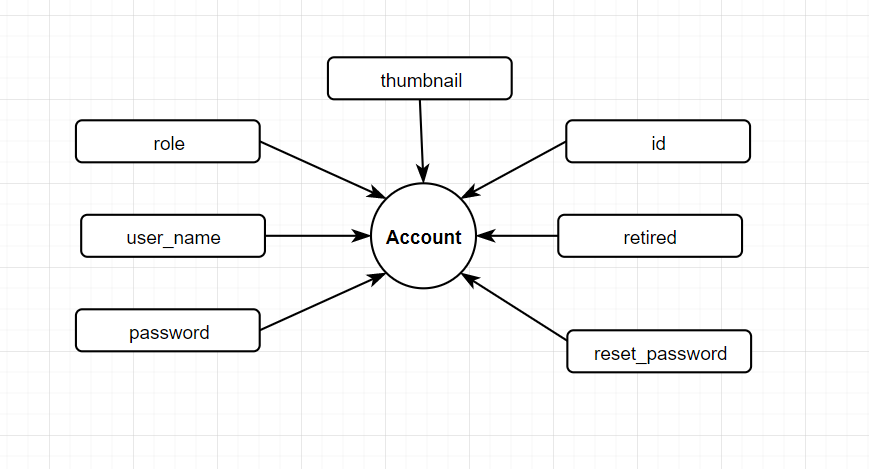
1. Airline

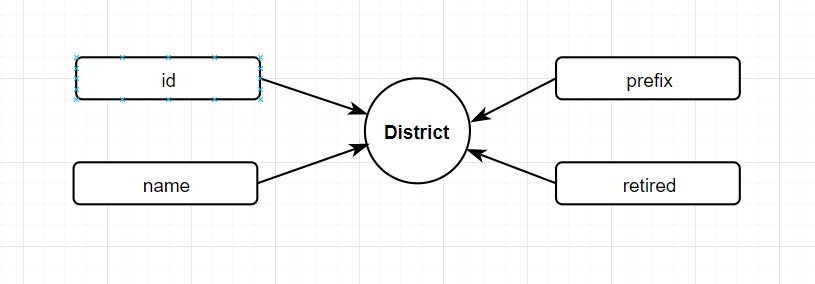


1. User

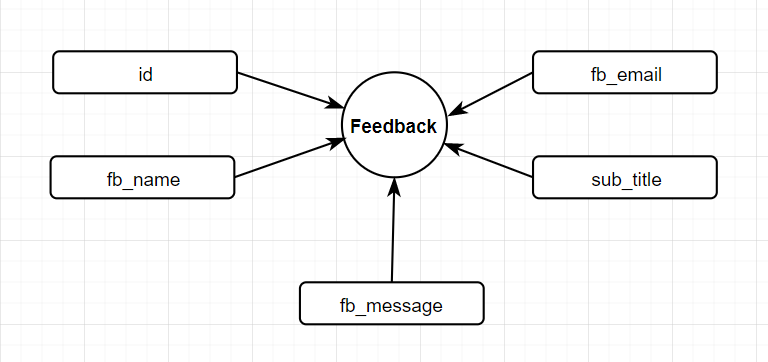


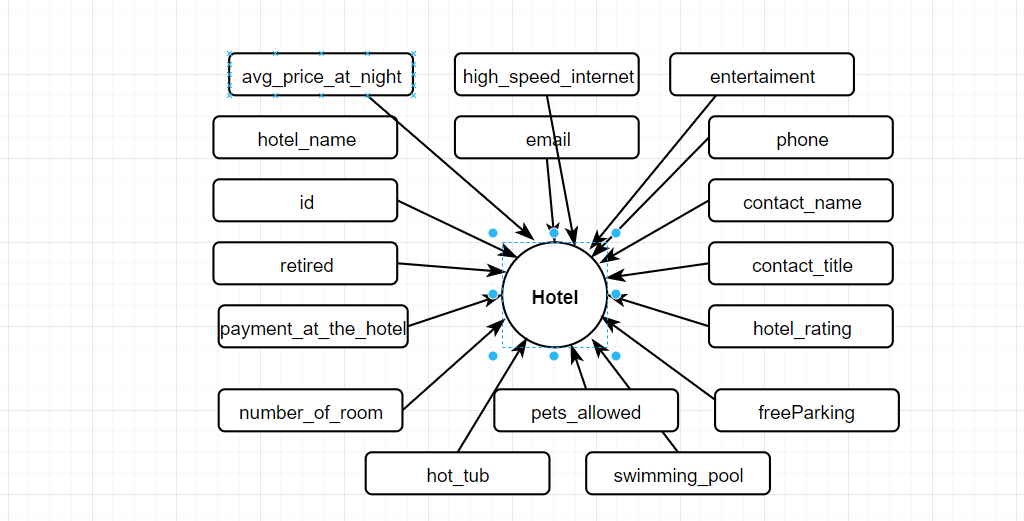
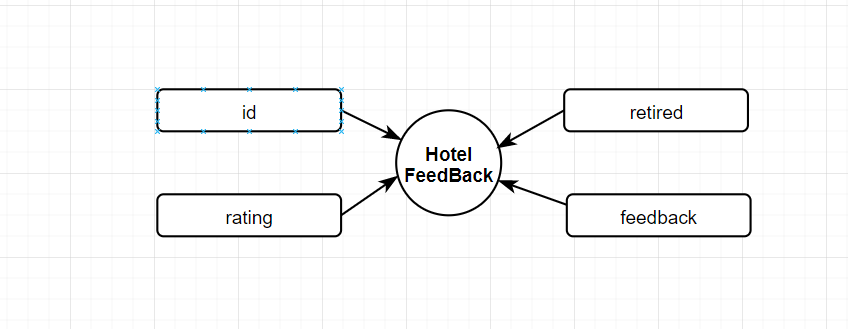
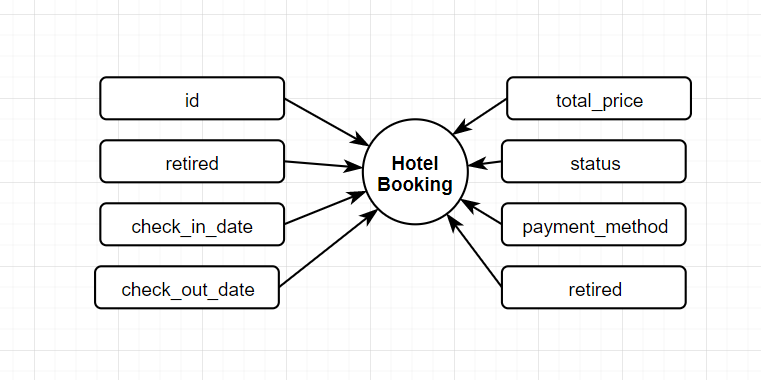
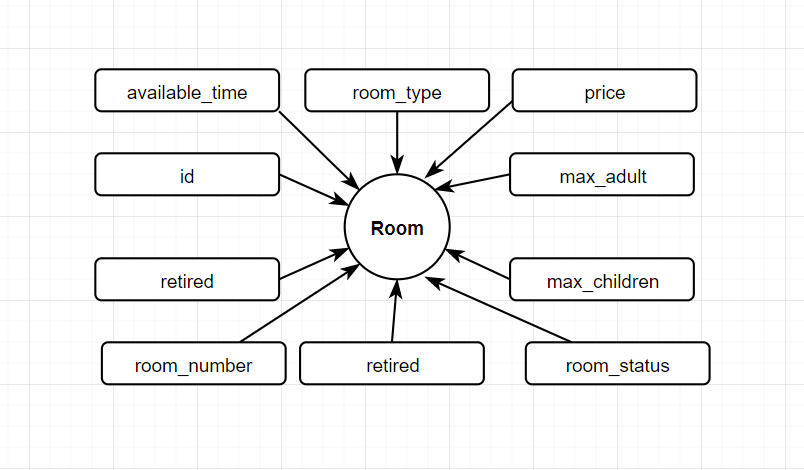
## Main entity description

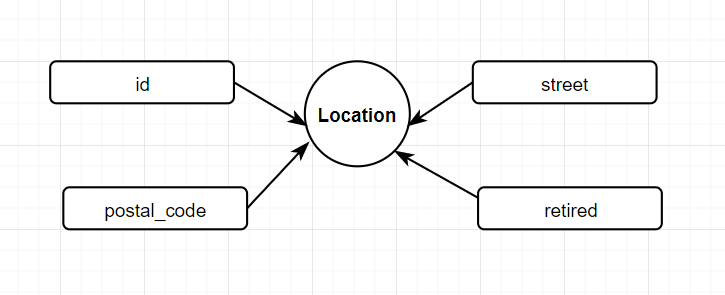
1. Account
2. District

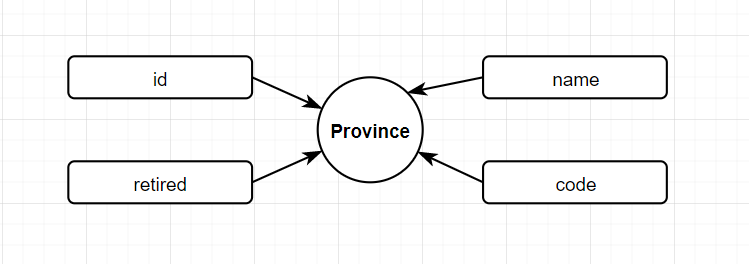


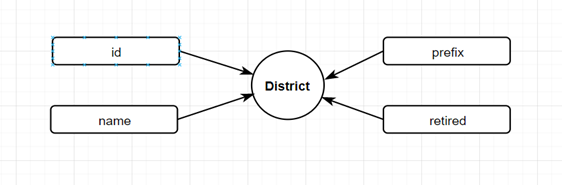
1. Feedback

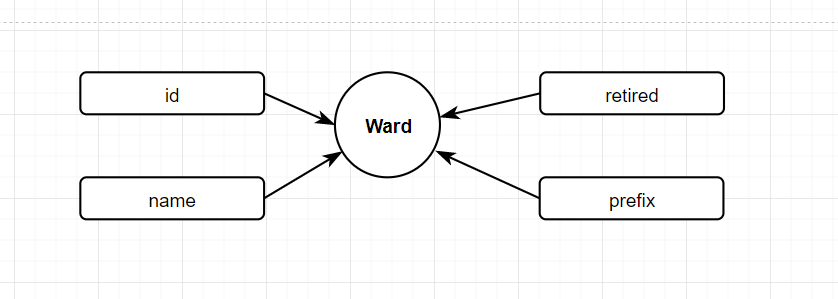
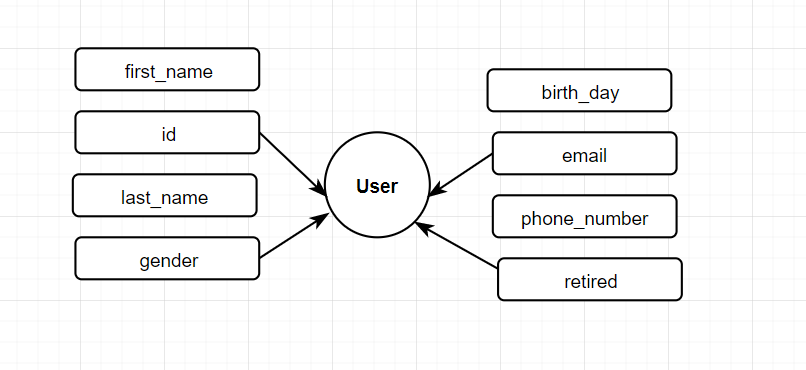
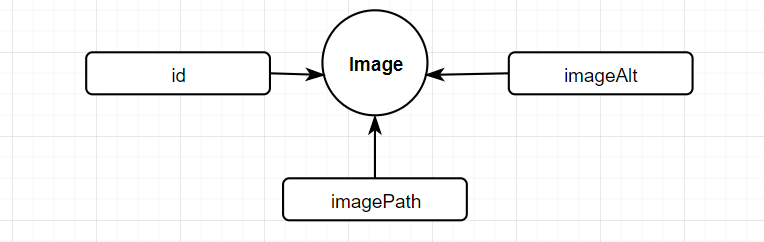


1. Hotel
2. Hotel Feedback
3. Hotel Booking
4. Room
5. Location

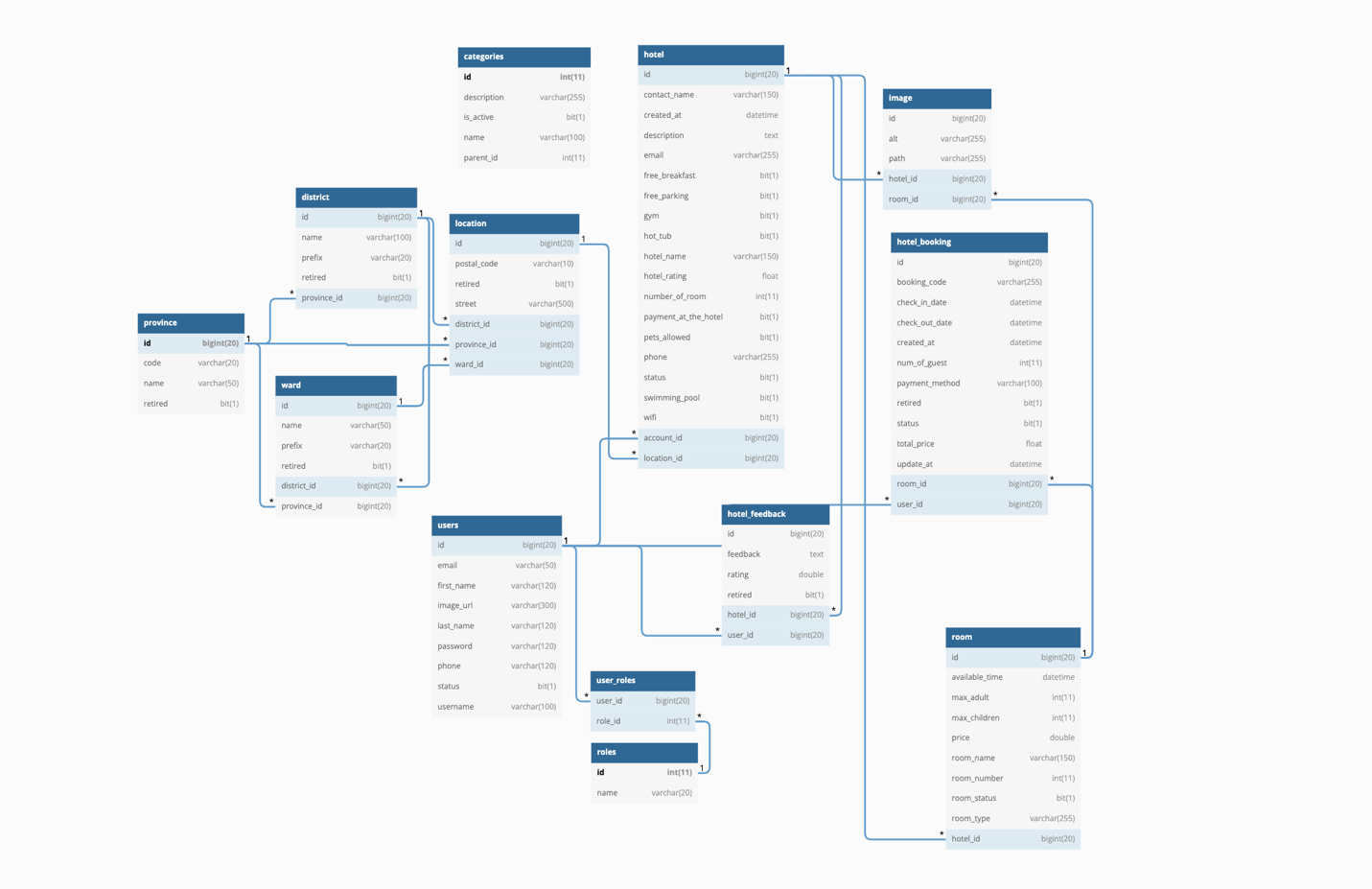


1. Province
2. District



1. Ward
2. User
3. Image

# **ENTITY RELATIONSHIP DIAGRAM**



# **TABLE DESIGN**

## Table Users:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Users** | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** |
| ID | bigint | No |  | PK | Identity | Stores user id |
| First Name | Character varying (100) | No |  |  |  | Stores user first name |
| Last Name | Character varying (100) | No |  |  |  | Stores last name |
| Phone | Character varying (255) | No |  |  |  | Stores user phone number |
| Status | boolean | No | true |  |  | Active check |
| Image url | bigint | No |  |  |  | Stores user image |
| Username | Character varying (255) | No |  | FK |  | Stores username of user |
| Password | Character varying (255) | No |  | FK |  | Stores Password of user |
| Email | Character varying (255) | No |  | FK |  | Stores Email of user |

## 

## Table District:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **District** | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** |
| ID | bigint | No |  | PK | Identity | Stores tour location id |
| Name | Character varying (255) | No |  |  |  | Stores name of district |
| Prefix | Character varying (20) | No |  |  |  | Stores Prefix of district |
| Retired | Boolean | Yes |  |  |  | Remove check |
| Province | bigint | No |  | FK |  | Stores province of district |
| Wards | bigint | No |  | FK |  | Stores ward of district |

## 

## Table Feed Back:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Feed Back** | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** |
| ID | bigint | No |  | PK | Identity | Stores tour image id |
| Name | Character varying (100) | No |  |  |  | Stores title of feedback |
| Email | Character varying (100) | No |  |  |  | Stores email of user who send feedback |
| Sub title | Character varying (100) |  |  |  |  | Stores sub title of feedback |
| Message | Character varying (255) |  |  |  |  | Stores description of feedback |

## Table Hotel:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hotel** | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** |
| ID | bigint | No |  | PK | Identity | Stores guide id |
| Hotel name | Character varying  (150) | No |  |  |  | Stores hotel name of hotel |
| Email | Character varying  (150) | No |  |  |  | Stores email of hotel |
| Phone | Int | No |  |  |  | Stores rating average |
| Contact name | Character varying  (150) | No |  |  |  | Stores contact name |
| Contact title | Character varying  (255) | No |  |  |  | Stores contact title of hotel |
| Address | Character varying  (255) | No |  |  |  | Stores address of hotel |
| Create at | timestamp | No |  |  |  | Stores create at of hotel |
| Status | Boolean | Yes |  |  |  | Remove check |
| Location | bigint | No |  | FK |  | Stores location of hotel |
| Account | bigint | No |  | FK |  | Stores account of hotel |

## 

## Table Room:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Room** | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** |
| ID | bigint | No |  | PK | Identity | Stores guide id |
| Room number | Int | No |  |  |  | Stores room number of room |
| Payment at the hotels | Boolean | No |  |  |  | Allow payment at the hotel |
| Available time | Date | Yes |  |  |  | Stores payment status |
| Room type | Character  varying (150) | No |  |  |  | Stores room type of room |
| Price | Double | No |  |  |  | Stores price of room |
| Max adult | Int | No |  |  |  | Stores max adult of room |
| Max children | Int | No |  |  |  | Stores max children of room |
| Status | Boolean | No |  |  |  | Remove check |
| Hotel | bigint | No |  | FK |  | Stores hotel of room |
| Hotel booking room | bigint | No |  | FK |  | Stores hotel booking room of room |

## 

## Table Hotel Booking:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hotel Booking** | | | | | | | | | | | |
| **Field Name** | **Data Type** | | **Null** | | **Default Value** | **Key** | | | **Table-Field-Ref** | **Description** | |
| ID | bigint | No | |  | | | PK | Identity | | | Stores guide id |
| Hotel type | Character  varying (150) | No | |  | | |  |  | | | Stores hotel type of hotel booking |
| Guest name | Character  varying (150) | No | |  | | |  |  | | | Stores guest name of hotel booking |
| Num of guest | Int | No | |  | | |  |  | | | Stores num of guest of hotel booking |
| Status | Boolean | No | |  | | |  |  | | | Approval type |
| Check in date | Date | No | |  | | |  |  | | | Stores check in date of hotel booking |
| Check out date | Date | No | |  | | |  |  | | | Stores check out date of hotel booking |
| Total price | Int | No | |  | | |  |  | | | Stores total price of hotel booking |
| Number of room | Int | No | |  | | |  |  | | | Stores number room of hotel booking |
| Create at | Timestamp | No | |  | | |  |  | | | Stores create at of hotel booking |
| Update at | Date | No | |  | | |  |  | | | Stores update at of hotel booking |
| Payment method | Character  varying (150) | No | |  | | |  |  | | | Stores payment method of hotel booking |
| Retired | Boolean | No | |  | | |  |  | | | Stores retired of hotel booking |
| Room Id | bigint | No | |  | | | FK |  | | | Stores hotel booking detail of booking |
| User | bigint | No | |  | | | FK |  | | | Stores user of booking |

## 

## XV. Table Hotel Feedback:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Hotel Feedback** | | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** | |
| ID | bigint | No |  | PK | Identity | | Stores guide id |
| Rating | Double | No |  |  |  | | Stores rating of hotel feed back |
| Feedback | Character  varying (255) | No |  |  |  | | Stores description of hotel feed back |
| User Id | bigint | No |  | FK |  | | Id of user feed back |
| Hotel | bigint | No |  | FK |  | | Stores hotel of hotel feedback |

## XVI. Table Location:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Location** | | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** | |
| ID | bigint | No |  | PK | Identity | | Stores guide id |
| Street | Character  varying (255) | No |  |  |  | | Stores street of location |
| Postal code | Character  varying (50) | No |  |  |  | | Stores postal code of location |
| Province | bigint | No |  | FK |  | | Stores province of location |
| District | bigint | No |  | FK |  | | Stores district of location |
| Ward | bigint | No |  | FK |  | | Stores ward of location |
| Retired | Boolean | Yes |  |  |  | | Remove check |

## XVII. Table Passenger:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Passenger** | | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** | |
| ID | bigint | No |  | PK | Identity | | Stores guide id |
| Firstname | Character  varying (200) | No |  |  |  | | Stores firstname of user |
| Lastname | Character  varying (200) | No |  |  |  | | Stores lastname of user |
| Brithday | Date | No |  |  |  | | Stores birthday of user |
| Gender | Boolean | No |  |  |  | | Stores bender of user |
| Card id number | Character  varying (200) | No |  |  |  | | Stores card id number of passenger |
| Card type | Int | No |  |  |  | | Stores card type of passenger |
| Card expired | Date | No |  |  |  | | Stores card expired of passenger |
| Retired | Boolean | Yes |  |  |  | | Remove check |

## XVII. Table Province:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Province** | | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** | |
| ID | bigint | No |  | PK | Identity | | Stores guide id |
| Name | Character  varying (50) | No |  |  |  | | Stores name of province |
| Code | Character  varying (50) | No |  |  |  | | Stores code of province |
| Retired | Boolean | Yes |  |  |  | | Remove check |

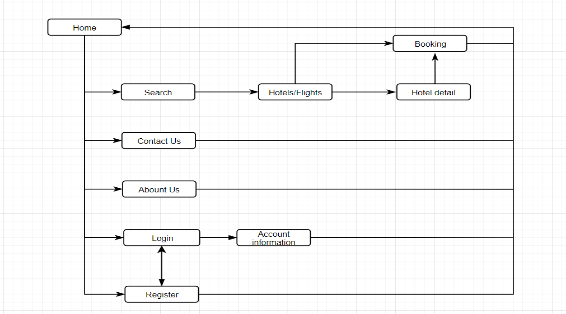
## XVII. Table Ward:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Ward** | | | | | | | |
| **Field Name** | **Data Type** | **Null** | **Default Value** | **Key** | **Table-Field-Ref** | **Description** | |
| ID | bigint | No |  | PK | Identity | | Stores guide id |
| Name | Character  varying (50) | No |  |  |  | | Stores name of ward |
| Prefix | Character  varying (20) | No |  |  |  | | Stores code of prefix |
| Province | bigint | No |  |  |  | | Stores province of ward |
| District | bigint | No |  |  |  | | Stores district of ward |
| Retired | Boolean | Yes |  |  |  | | Remove check |

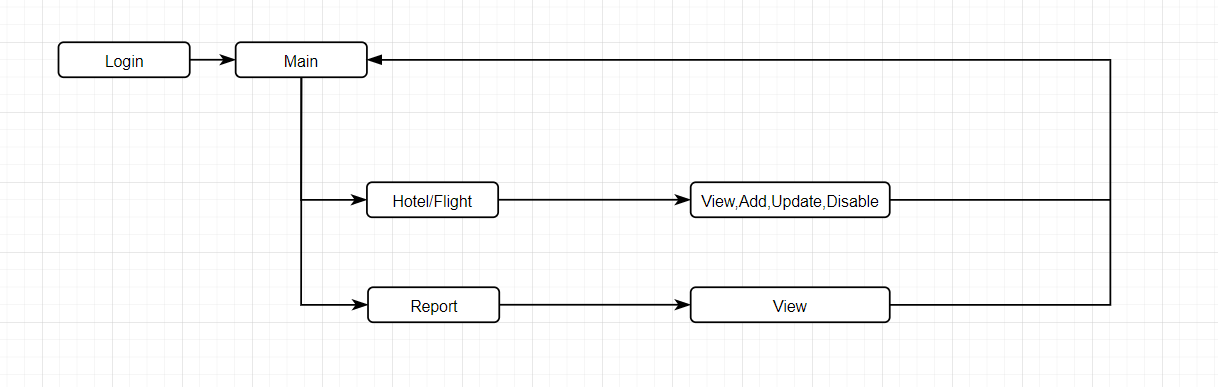
# 

# **Site Maps**

## User site:



## II. Supplier:



## III. Admin:

## 

# **TASK SHEET REVIEW 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:**  **TRAVEL ToVN** | | **Project Title:** | **Activity Plan Prepared By:** | **Date of Preparation of Activity Plan:** | | | |
| **Sr.No.** | **Task** | **Actual Start Date** | **Actual Days** | **Team Mate Names** | **Status** |
| 1 | DFD (Data Flow Diagram) | **travelola** | Nguyen Ngoc Hung | 17/06/2022 | 1 | Nguyen The Long | 100% |
| 2 | Flowcharts | 17/06/2022 | 1 | Phung Hoai Bao | 100% |
| 3 | ERD (Entity Relationship Diagram) | 17/06/2022 | 1 | Nguyen Ngoc Hung | 100% |
| 4 | Table Design | 17/06/2022 | 1 | Pham Van Nguyen | 100% |

|  |  |
| --- | --- |
| **Date: 17/06/2022** | |
| Signature of Instructor:  **MR. NGO PHUOC NGUYEN** | Signature of Team Leader:  **Nguyen Ngoc Hung** |