FPT ACADEMY INTERNATIONAL

FPT – APTECH COMPUTER EDUCATION

**Centre Name: ACE-HCMC-2-FPT.**

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

# FLIGHT BOOKING

| | | |
|---|---|---|
| **Supervisor:** | Nguyen Hoang Thao | |
| **Semester:** | 04 | |
| **Batch No:** | F2-13-1704-T0 | |
| **Group No:** | 1 | |
| **Order:** | **Full name** | **Roll No.** |
| 1. | Pham Le Tuan Lam | student1025615 |
| 2. | Vo Minh Phat | student1022128 |
| 3. | Nguyen Huu Tri | student1021966 |

Month 02 Year 2019

# Table content


**1 Problem Definition** ..... 1

**2 Customer Requirement Specification** ..... 1

**2.1 User use System** ..... 1

**2.2. Functional Requirements.** ..... 2

**2.2.1 Customer** ..... 2

**2.2.2 Admin** ..... 3

**2.3 System Requirements** ..... 4

**2.3.1 Hardware Requirements** ..... 4

**2.3.2. Software Requirements.** ..... 5

**4. Use – case** ..... 7

**5. Data Flow Diagram (DFD)** ..... 7

**5.1 Context diagram** ..... 7

**5.2 DFD Level 1** ..... 8

**5.2.1 Sign up** ..... 8

**5.2.2 Logon** ..... 8

**5.2.3 Search Flight** ..... 8

**5.2.4 Trace Booking** ..... 8

**5.2.5 Booking** ..... 9

**5.2.6 Dashboard** ..... 10

**5.2.7 Report** ..... 10

**5.2.7 Category** ..... 11

**6. Sequence** ..... 12

**6.1 Booking** ..... 12

**6.2 Sign Up** ..... 14

**6.3 Logon** ..... 16

**6.4 Search Flight** ..... 17

**6.5 Trace Booking** ..... 18

**6.6 DashBoard** ..... 19

**6.7 Category** ..... 21

**6.8 Report** ..... 23

**7. Entity Relationship (E-R) Diagram** ..... 25

**7.1 E-R Diagram** ..... 25

**7.2 Table** ..... 26

**7.2.1 Region** ..... 26

**7.2.2 Airport** ..... 26

**7.2.3 Airline** ..... 27
**7.2.4 Flight** ..... 27
**7.2.5 Class** ..... 27
**7.2.6 ClassDetail** ..... 27
**7.2.7 ClassFlight** ..... 27
**7.2.8 Ticket** ..... 28
**7.2.9 Passenger** ..... 28
**7.2.10 Payment** ..... 29
**7.2.11 TicketDetailPrice** ..... 29
**7.2.12 Account** ..... 30
**7.2.13 Role** ..... 30
**7.2.14 AccountRole** ..... 30
**8. Site map** ..... 31
**8.1 Passenger** ..... 31
**8.2 Admin** ..... 32
**9. UI** ..... 33
**9.1 Home Page** ..... 33
**9.2 Sign Up** ..... 34
**9.3 Logon** ..... 35
**9.4 Search Flight** ..... 36
**9.5 Booking** ..... 37
**9.6 Passenger Info** ..... 38
**9.7 Payment** ..... 39
**9.8 Trace Booking** ..... 40
**9.9 Dashboard** ..... 41
**9.10 Report** ..... 42
**9.11 Category** ..... 43

# REVIEW 1

# 1 Problem Definition

In recent years, people's mobility and demand for tourism has increased. More and more vehicles contribute to the development of transport network problems. Types of vehicles such as buses, coaches, taxis, trains ... only meet short travel distances and don't bring high safety to customers. Airlines in Vietnam is growing with many famous brands such as Vietnamairlines, VietJet, JetStar ... bringing high competitive.

With the purpose of serving the quality of life, serving the needs of customers, our FTA airline company was born to meet the increasing demand and travel of domestic customers. In the context of increasingly busy and bustling life, the issue of air ticket booking is quite a problem. Suppose we had a quick flight tomorrow but today the job is too busy and we can't go to the ticket counter directly to buy it, or buy online tickets online but we don't know where to buy, the form and the procedure is too complicated, the price is too high or sometimes there is no suitable flight for us. To solve this problem, our airline has developed the "Flight Tickets Anywhere (FTA)" website, which helps customers to book flight tickets anytime, anywhere and most quickly.

With a friendly, simple, easy-to-use interface, experienced and enthusiastic online support team, high speed website to meet the needs of access, especially on peak occasions, FTA will help customers book tickets The ball machine is fast with the simplest procedure, especially the price is cheap compared to other firms.

# 2 Customer Requirement Specification

## 2.1 User use System

- **Customer:**
  - Who use the webpage to search flight, Book flight, payment, and Tracing their booking (View detail of history booking). Customers can sign up to become a meber of a webpage.
- **Admin:**
  - Who manage the web: Can Create, update: Passenger, Account , Airline, Airport, Region and other information of the web and make report.

## 2.2. Functional Requirements.

### 2.2.1 Customer

- **Search Flight:**
  - Customer choice departure, destination, depart day, return day, number of customer, airline on web for searching the flight that they want.
  - Departure, destination, depart day, number of customer is required, cusomter choise in view. Customer can choise depart day from present to one year in future. Number customer default is one if customer do not input. Maximum every booking is nine include Child, Infant, Adult.
  - View must have notifications to customer if they input invalid imfomation.
  - The web will return all flight corresponding with search condition of passenger.
- **Book Flight:**
  - Passenger can choice Flight from list of after they search. The web will display imfomation of the the flight choosen. Include: Class, terms of class, Depart time, Arrive time, Aircraff, Departure, Destination.
  - After choice Flight passeger will see Price sumary include fare, Tax, Fee carrier charges.
  - Continue to fill information detail: Title, given name, Last name, date of birth, gender, Contact info: phone number, Email address.
  - Accept Term then select payment online with Card number, Name on card, Expiration date.
  - Customer can edit departure, destination, depart day, return day, number of customer, airline in booking process before make payment. The web will give new flights according to new information that customer input.
  - After payment successful, system will send email ticket in PDF format to customer, Customer can print this ticket to check-in in airport.
  - In Booking process must have notifications when input value is not valid.
- **Trace Booking**
  - Customer can view their trip information by enter following exactly imfomation: Reservation, Last Name, Email Address or logon by account to see all history

booking. If customers are not the member, they can only view history of booking by Reservation, Last Name, Email Address for every finding.

- Web will display about list of flight that customer booked: , Depart time, Arrive time, Aircraff, Departure, Destination.

❖ **Sign up**

- Customer can sign up to system to become a member of system.
- Customer fill account id, account name, email, phone, location, date of birth, gender into form that system provide to sign up.
- Account id, account name, email, phone is required.
- View have notifications if imformation customer input is not valid or sign up success.

❖ **Login**

- If Customer signed up they can logon to system by filling account Id, password registried.
- Customer can view, update their profile information ( Account name, date of birth, gender, email, phone).
- Customer can view history booking: Depart time, Arrive time, Aircraff, Departure, Destination,….
- View have to display message if logon fail or display name of customer in webpage.

❖ **View News, Contact info of the web.**

- In home page display recommend about cheap fligh from airline, Customer can choise derect from this to book flight.
- When customer need help they can view contact information of the web: Email, phone to contact.

### 2.2.2 Admin

❖ **Dashboard**

- Admin can view Sale overview: Total sale in days, in month, Total profit.
- View list of lastest Booking include imformation: Airline, Departure, destination,…
- View list of new member, total member.

❖ **Make Report**

- The admin can make report by input filter: Airline, Airport, Region in range of time. Print report with PDF Format.

❖ **User management**

- Create, update, view, delete information of user in system by input Account ID, Account Name.
- Grant, revoke role to user.

❖ **Manage information about Web**:

About info displayed in view that customer can choise direct when search flight. Or admin use to filter when make report, view dashboard. All value input has to valid and has nofifications

- **Airline:** This is information about provider flight. Admin can create Airline by input AirlineID, Airline Name. Update Region by input AirlineName.
- **Airport:** This is information about where passenger can depart or arrive. Admin can create Airport by in put Airport ID, Airport Name, Region ID the Airport belong.
- **Region:** This is imformation about that Airport located. Admin can Create region by input RegionID, Region Name. Update Region by input Region name.
- **Contact info:** admin can create or edit contact info of the web: Phone, email, time work.

## 2.3 System Requirements

### 2.3.1 Hardware Requirements

**Server:**

| Component | Requirement |
|---|---|
| **CPU** | Processor type:<br>Core-compatible processor or faster<br>Processor speed:<br>Recommended: 2.0 GHz or faster |
| **OS** | Microsoft Windows 7 with IIS 7 or higher |

| | |
|---|---|
| **Memory (RAM)** | RAM:<br>Minimum: 3 GB<br>Recommended: 4 GB or more |
| **Hard Drive** | Free space:<br>Minimum: 20 GB<br>Recommended: 50 GB or more<br>Maximum: Operating system maximum |

**Client:**

| **Component** | **Requirement** |
|---|---|
| **CPU** | Processor type:<br>Core-compatible processor or faster<br>Processor speed:<br>Recommended: 2.0 GHz or faster |
| **OS** | All OS(Window ,Linux ,Android ,Mac OS …) |
| **Memory (RAM)** | RAM:<br>Minimum: 3 GB<br>Recommended: 4 GB or more<br>Maximum: Operating system maximum |
| **Hard Drive** | Free space: Minimum: 20GB |

### 2.3.2. Software Requirements.

**Server:**

| **Component** | **Requirement** |
|---|---|

| | |
|---|---|
| **Visual Studio** | Visual Studio 2013 or higher |
| **Microsoft .NET Framework** | Version 4.5 or higher |
| **RDBMS** | Microsoft SQL Server 2012 or higher |
| **Front-End** | HTML5CSS3(sass, Bootstrap 3), JavaScript(JQuery), Ajax |
| **Back-End** | ASP.NET MVC 5, ASP.NET WEB API 2, Entity Framework |

**Client:**

| **Component** | **Requirement** |
|---|---|
| **CPU** | Processor type:<br>Core -compatible processor or faster<br>Processor speed:<br>Recommended: 2.0 GHz or faster |
| **OS** | All OS(Window ,Linux) |
| **Memory (RAM)** | RAM:<br>Minimum: 2 GB<br>Recommended: 4 GB or more<br>Maximum: Operating system maximum |
| **Hard Drive** | Free space:<br>Minimum: 20GB |
| **Web Browser** | Firefox 60+,Chrome 60+ |

**Review 1 - Task Sheet**

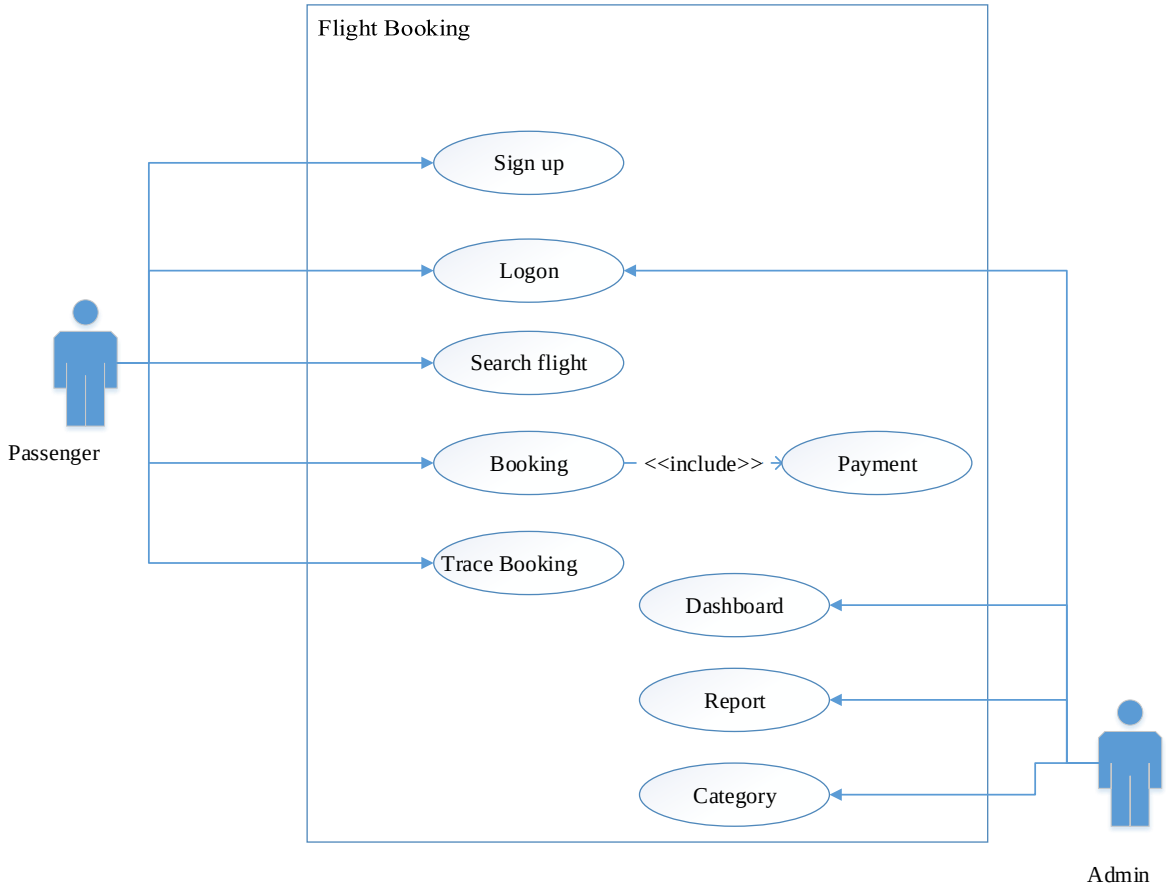
| **Project Ref. No.:**<br>**Project title: Flight Booking** | | **Activity Plan Prepared by: Trí** | | **Date of Preparation of Activity Plan** | | |
|---|---|---|---|---|---|---|
| **Sr.No.** | **Task** | **Start Date** | **End Date** | **Actual day** | **Member's Name** | **status** |
| 1 | Problem Definition | 15/02/2019 | 17/02/2019 | 3 | All members | Complete |
| 2 | Customer Requir Specification | | | | | |
| 3 | Functional Requirement Specification | | | | | |
| 4 | System Requirements | | | | | |

**Group Leader** **Faculty**

Nguyen Huu Tri Nguyen Hoang Thao

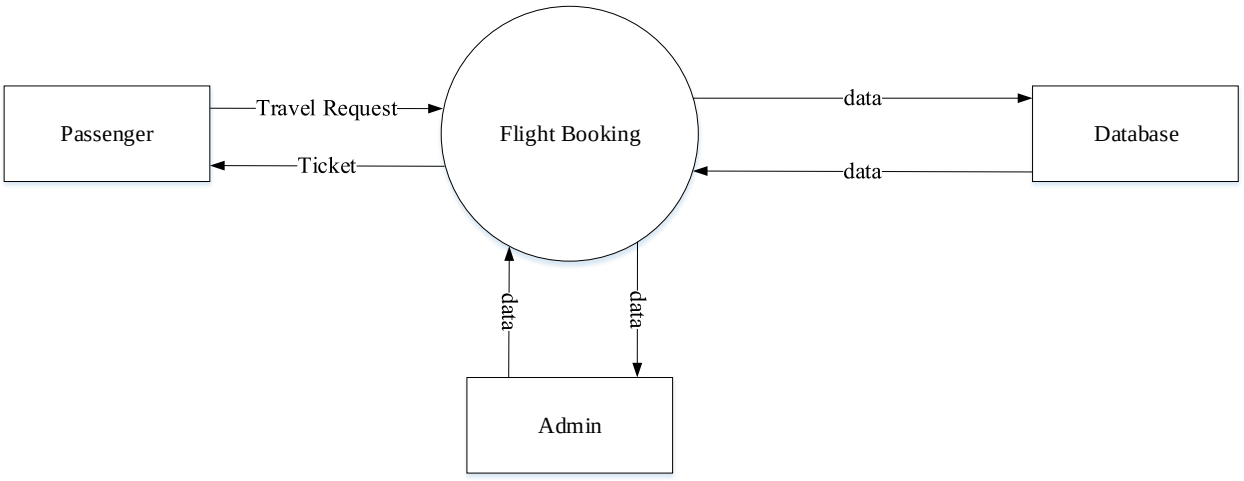
# REVIEW 2

# 4. Use – case



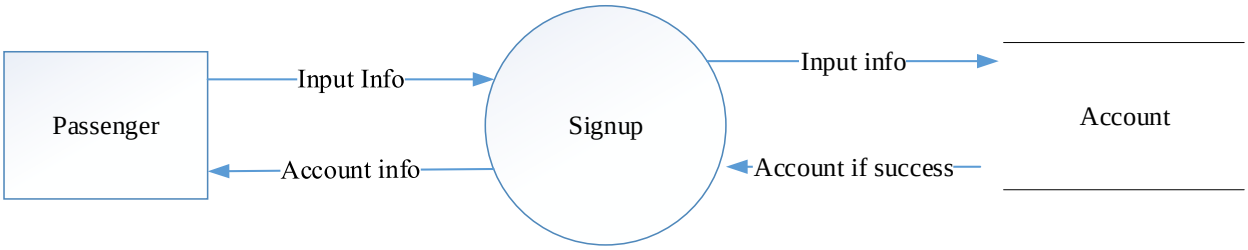

# 5. Data Flow Diagram (DFD)

## 5.1 Context diagram

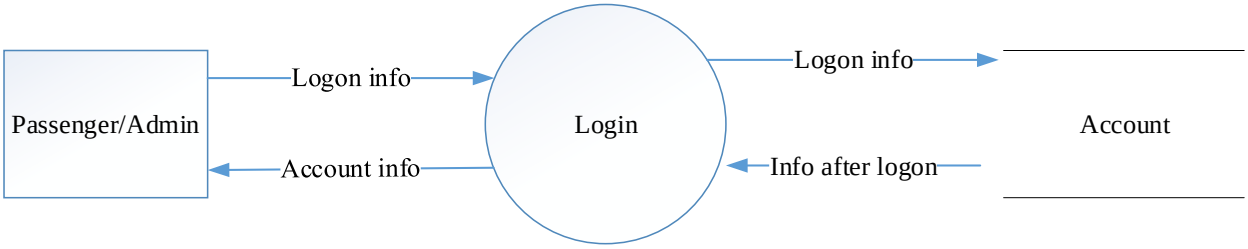


## 5.2 DFD Level 1

### 5.2.1 Sign up

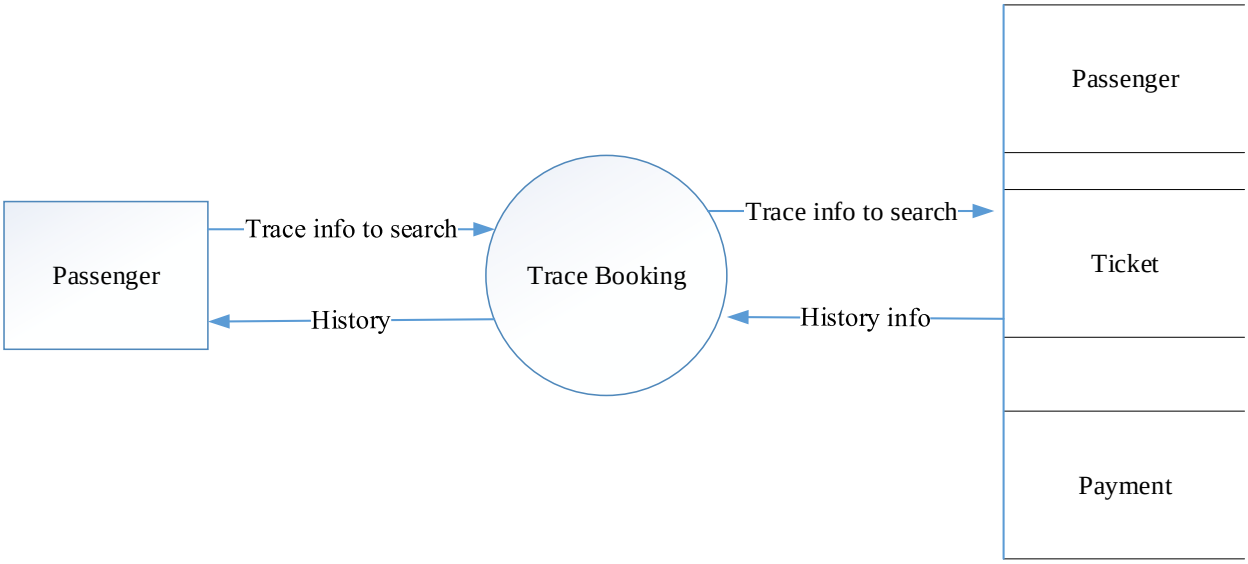



### 5.2.2 Logon

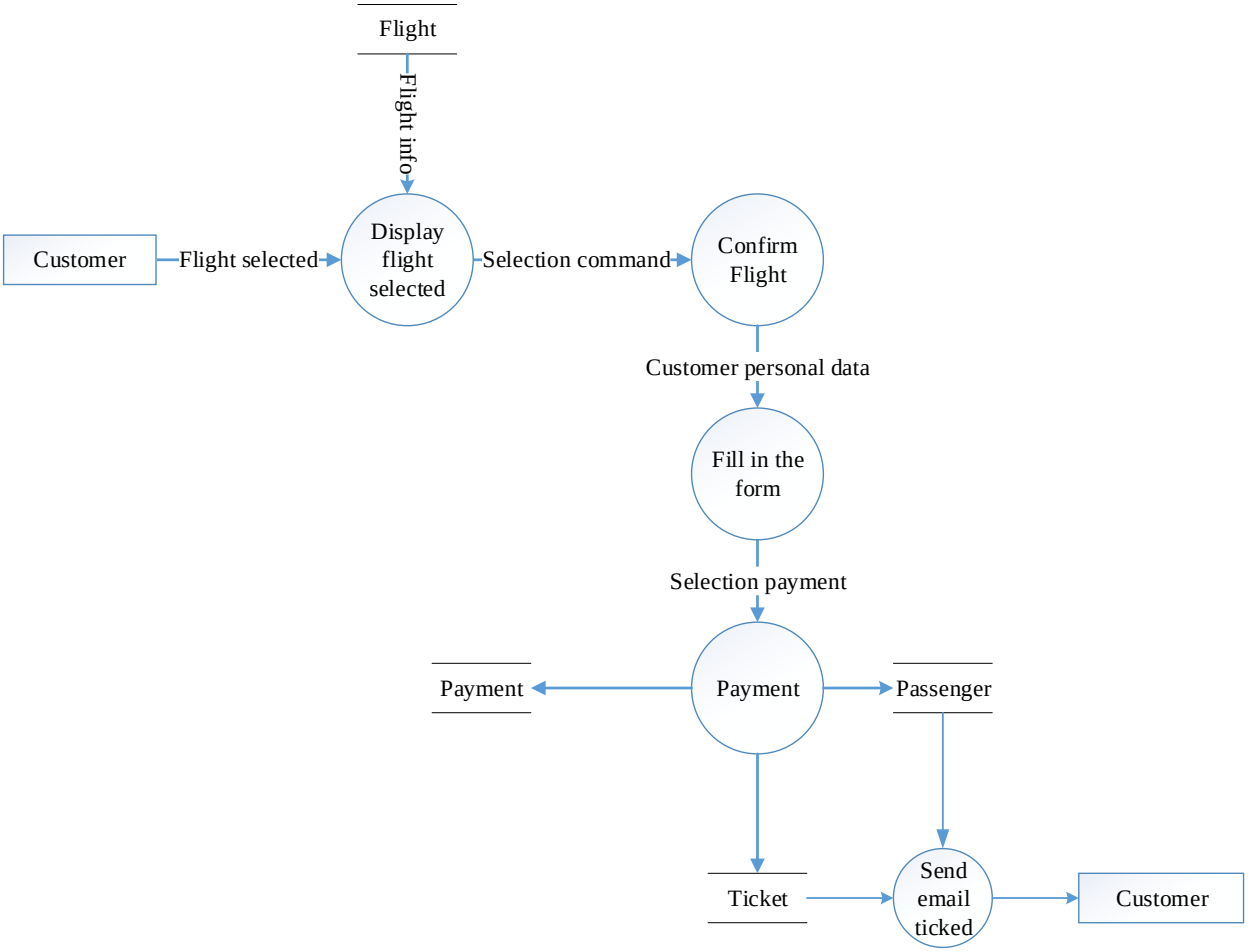



### 5.2.3 Search Flight

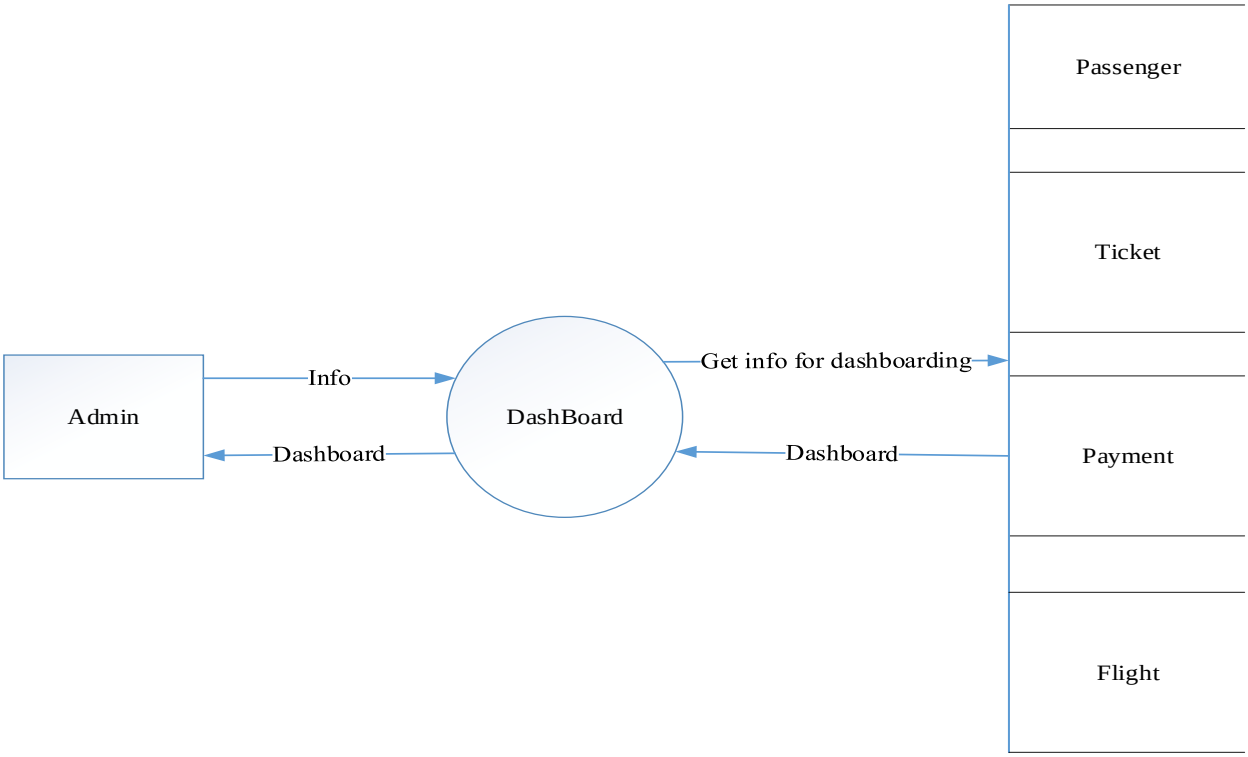
### 5.2.4 Trace Booking



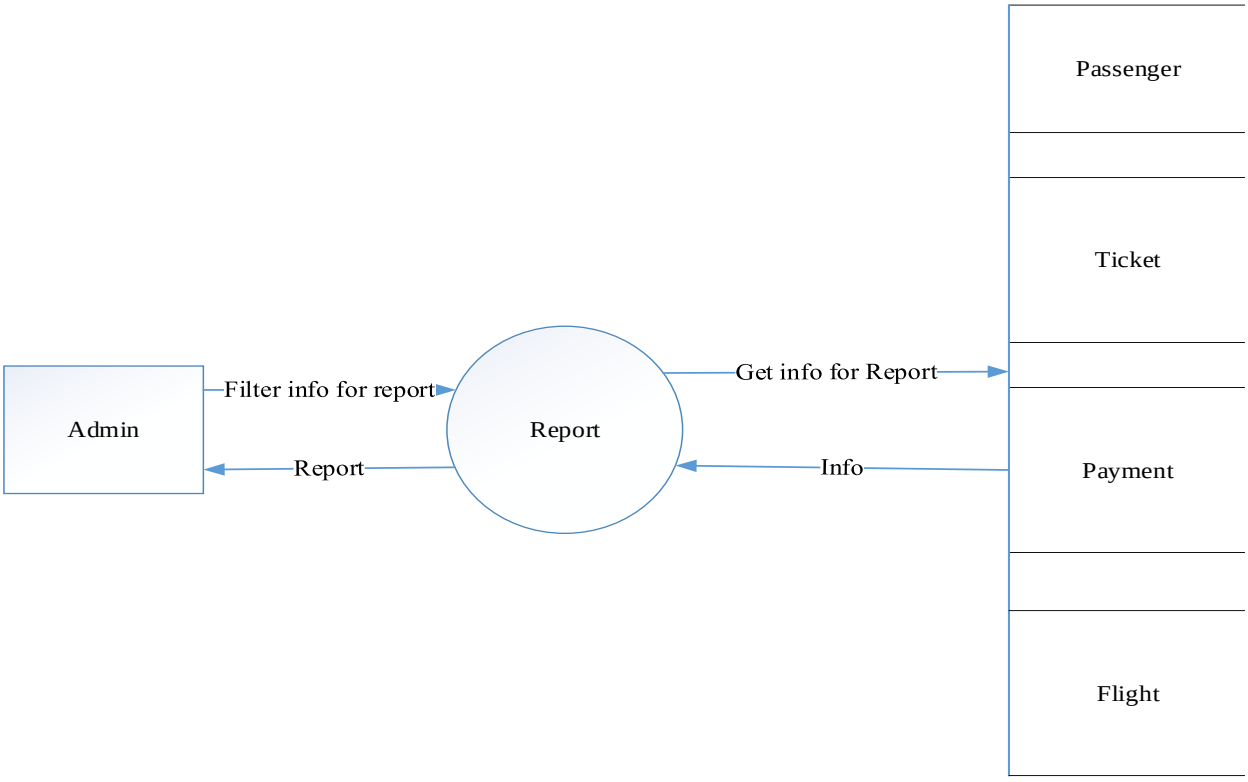
### 5.2.5 Booking


Flight
Flight info
Customer
Flight selected
Display flight selected
Selection command
Confirm Flight
Customer personal data
Fill in the form
Selection payment
Payment
Payment
Passenger
Ticket
Send email ticked
Customer

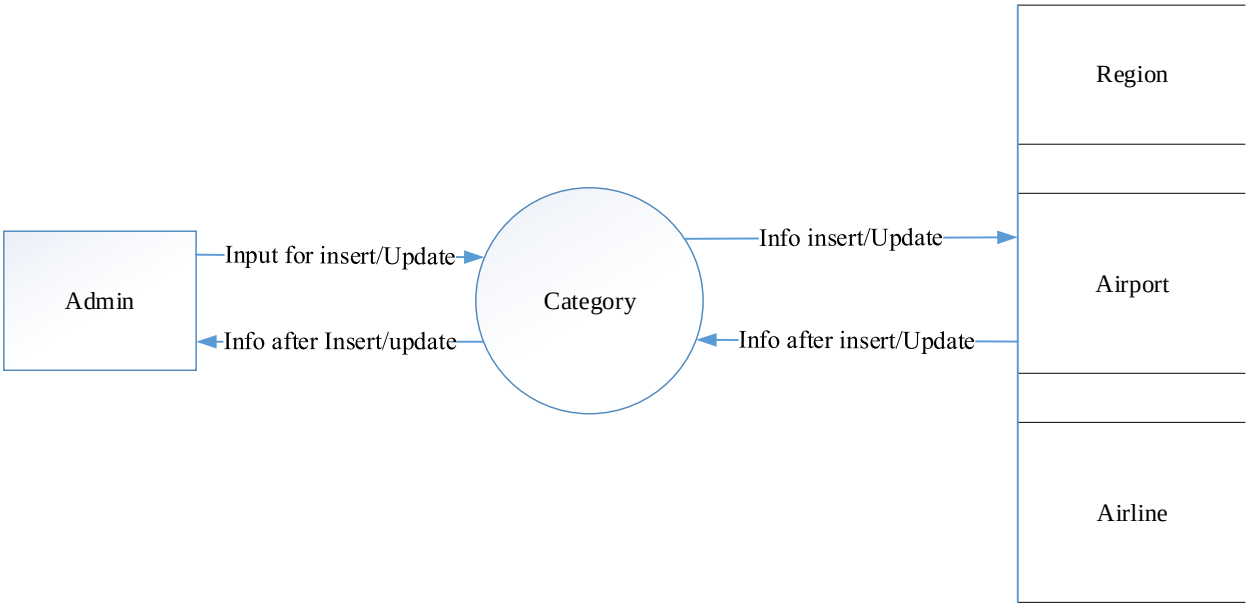
### 5.2.6 Dashboard




### 5.2.7 Report

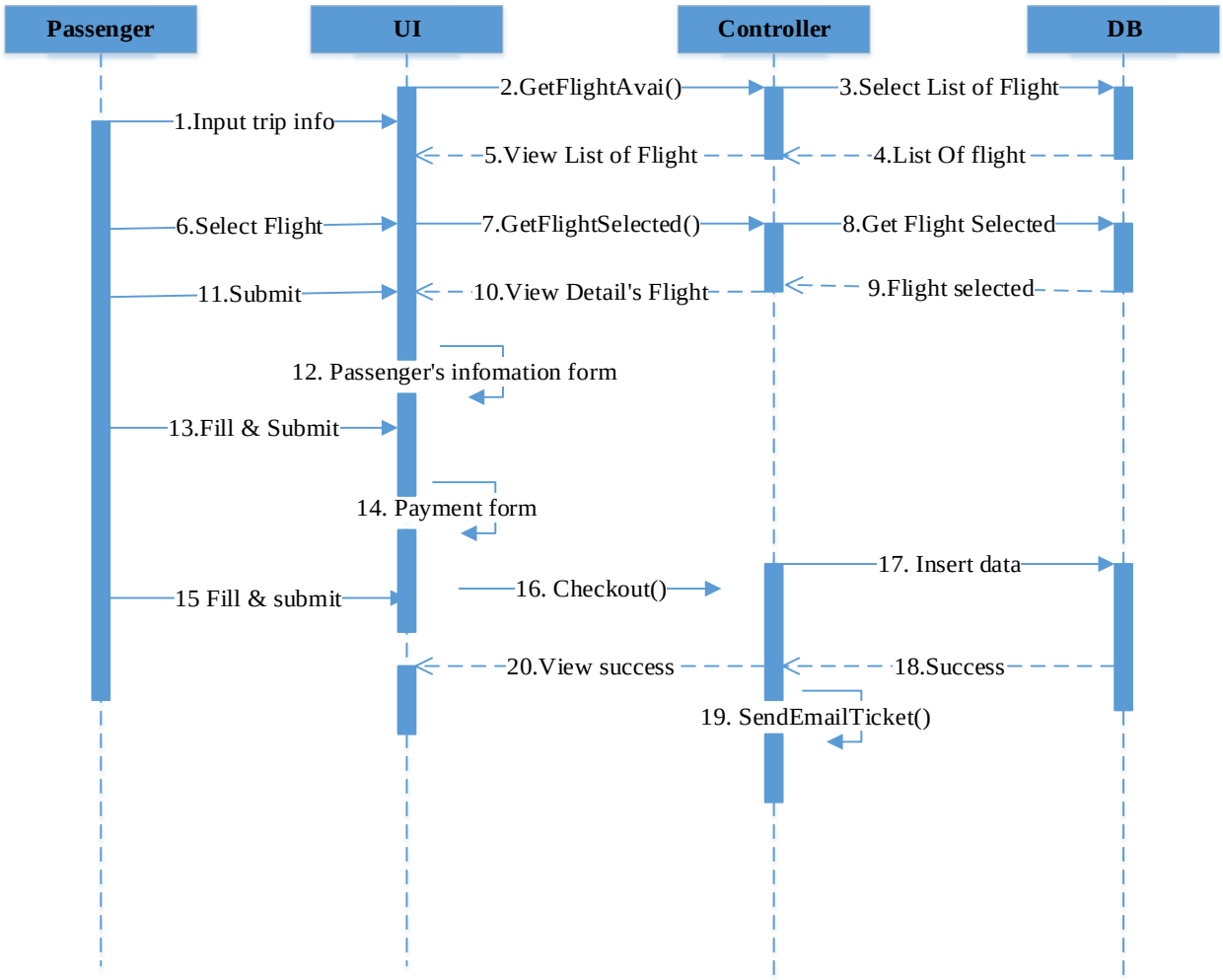


### 5.2.7 Category



# 6. Sequence

## 6.1 Booking

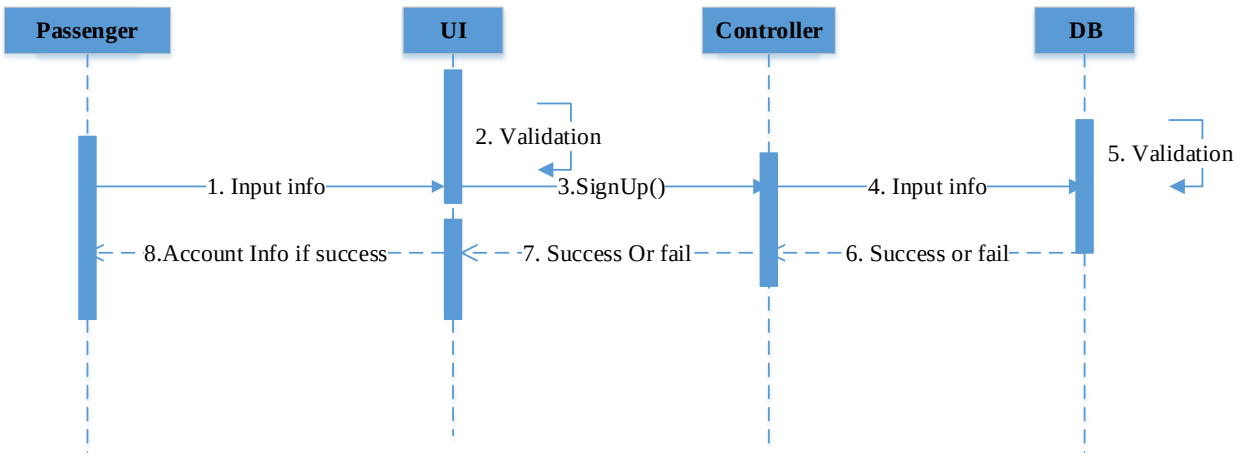



| **Author** | | |
|---|---|---|
| **Use case name** | Booking | |
| **Actor** | Passenger | |
| **Description** | Passenger wanna booking flight | |
| **Requirements** | Passenger provide information about trip | |
| **Pre-conditions** | | |
| **Post -Condition** | Success: have a message confirm and get Ticket. | |
| **Basic flow** | **Actor action** | **System Responses** |

| | | |
|---|---|---|
| | 1. Input the trip info into booking page, then click Searh.<br>6. Passenger choise flight from UI then click Select button.<br>11. Passenger submit the flight on UI<br>13. Passenger input information to form and submit<br>15. Passenger input payment info and submit | 2. System call getFlightAvai() to controller to get data from DB with info that passenger input.<br>3. Controller get match info from DB.<br>4. DB return all match info to controller.<br>5. Controller return view to UI.<br>7. System call getFlightSelected() to Controller to get flight detail that passenger selected.<br>8. Controller get flight detail that passenger selected from DB.<br>9. DB return flight detail that passenger selected to controller.<br>10. Controller return view of detail that passenger selected.<br>12. Display form that passenger input info. |

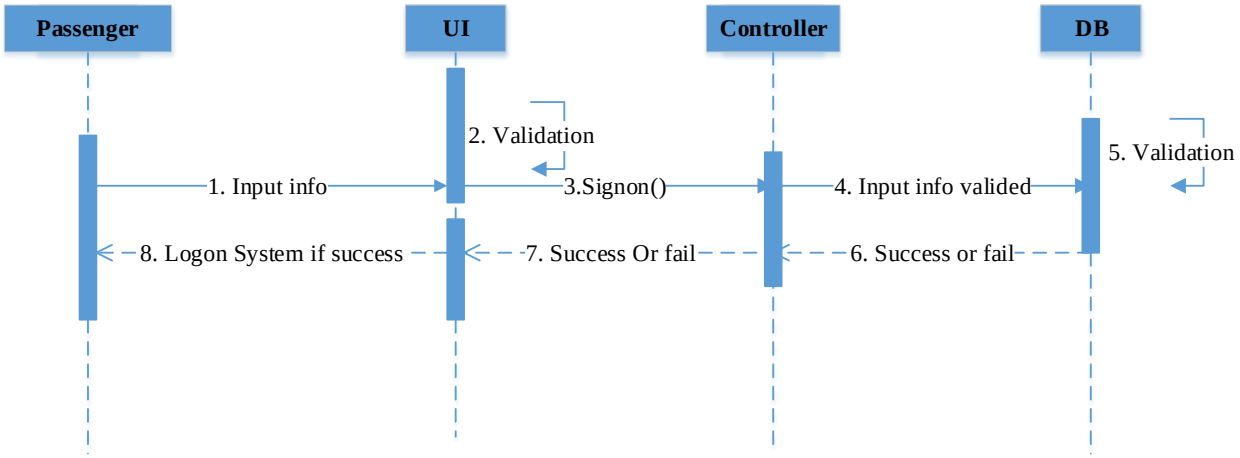
| | | |
|---|---|---|
| | | 14. UI validate and display payment form.<br>16. UI call CheckOut() medthod to Controller.<br>17. Controller insert data to DB.<br>18. DB return message to controller.<br>19. Controller call SendEmailTicket().<br>20. Controller display message success to UI. |
| **Alternative flow** | | |
| **Exception** | All field is blank or day depart greater than day return. | There are message to passenger and no result return |

### 6.2 Sign Up



| **Author** | | |
|---|---|---|
| **Use case name** | SignUp | |
| **Actor** | Passenger | |
| **Description** | Passenger wanna to registry to website | |
| **Requirements** | Passenger provide information to sign up | |
| **Pre-conditions** | | |
| **Post -Condition** | Success: Passeger become member of website | |
| **Basic flow** | **Actor action** | **System Responses** |
| | 1. Passger go to Sign up page then Input infomation.<br>8. If success passenger have become member of website with account. | 2. UI Valadate<br>3. System call SignUp() to controller<br>4. Controller get match info from DB.<br>5. Validate info in DB<br>6 . DB return result.<br>7. Controller return view Sign up success or fail |
| **Alternative flow** | | |
| **Exception** | 1. All field is blank.<br>2. Email is blank or exitst.<br>3. AccountID length is <3 or >20<br>4. Password lenth is < 6 or > 14.<br>5. Account ID is exists in DB. | There are message to passenger if input is not valid.<br>If input is valid but Account ID is exists in DB, message return form DB. |

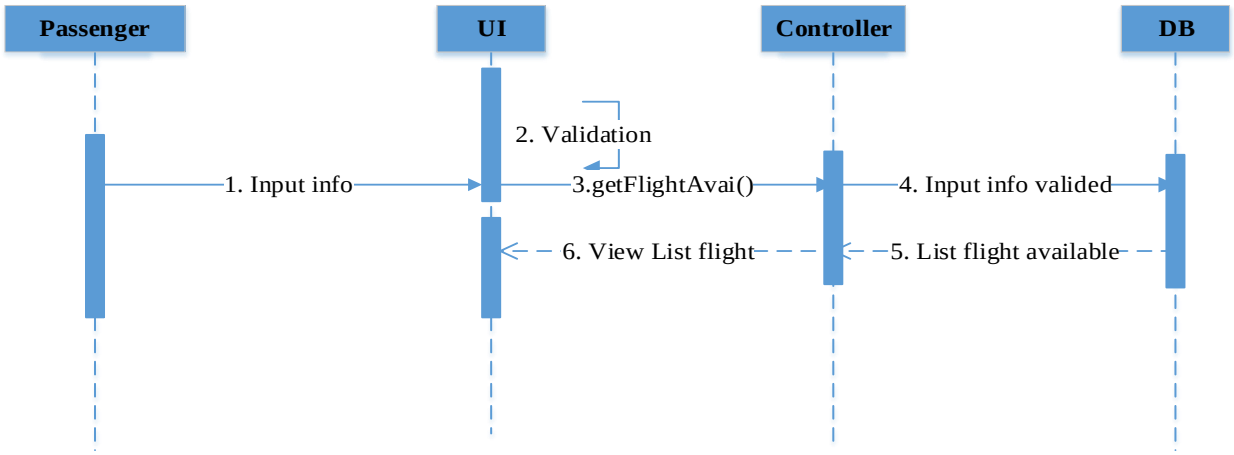
### 6.3 Logon




| | | |
|---|---|---|
| **Author** | | |
| **Use case name** | Logon | |
| **Actor** | Passenger | |
| **Description** | Passenger wanna to logon to website | |
| **Requirements** | Passenger provide accountID and password. | |
| **Pre-conditions** | Passenger registed. | |
| **Post -Condition** | Success: Passeger logon to website | |
| **Basic flow** | **Actor action** | **System Responses** |
| | 1. Input Account info and submit.<br>8. If success passenger logon to website | 2. UI Valadate<br>3. System call Sigon() to controller<br>4. Controller get match info from DB.<br>5. Validate info in DB<br>6 . DB return result.<br>7. Controller return view login success or fail |

| | | |
|---|---|---|
| **Alternative flow** | | |
| **Exception** | 1. All field is blank.<br>2. Account ID is not exists in DB. | There are message to passenger if input is not valid.<br>If input is valid but not match in DB, message return form DB. |

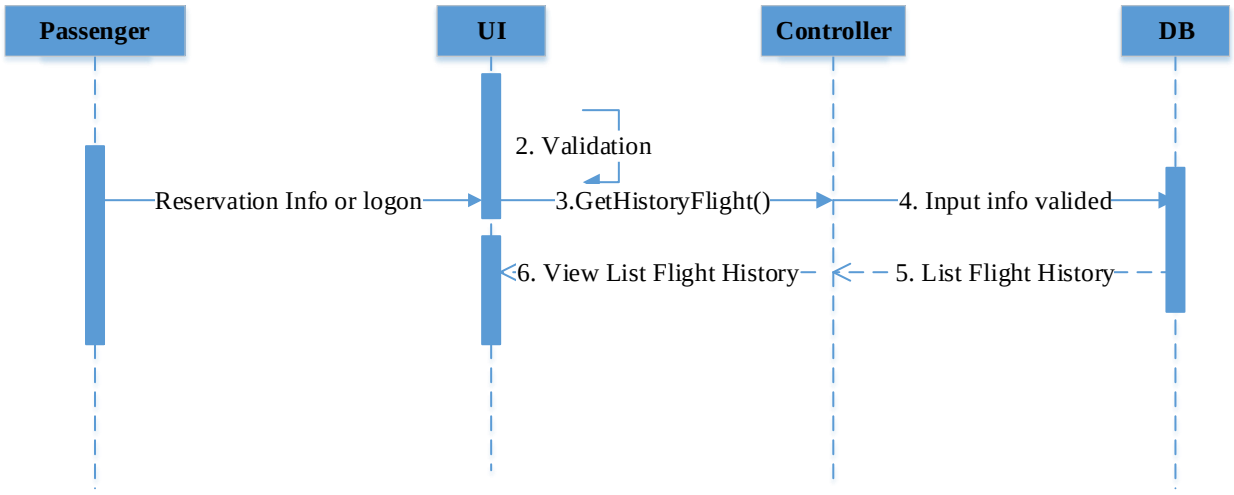
### 6.4 Search Flight




| | | |
|---|---|---|
| **Author** | | |
| **Use case name** | Booking | |
| **Actor** | Passenger | |
| **Description** | Passenger wanna booking flight | |
| **Requirements** | Passenger provide information to search | |
| **Pre-conditions** | | |
| **Post -Condition** | Success: All Flight match filter info return. | |
| **Basic flow** | **Actor action** | **System Responses** |
| | 2. Input the trip info into booking page, then click Searh. | 2. UI Valadate |

| | | |
|---|---|---|
| | | 3. System call getFlightAvai() to controller to get data from DB with info that passenger input.<br>4. Controller get match info from DB.<br>5. DB return all match info to controller.<br>6 .Controller return view to UI. |
| **Alternative flow** | | |
| **Exception** | All field is blank or day depart greater than day return. | There are message to passenger and no result return |

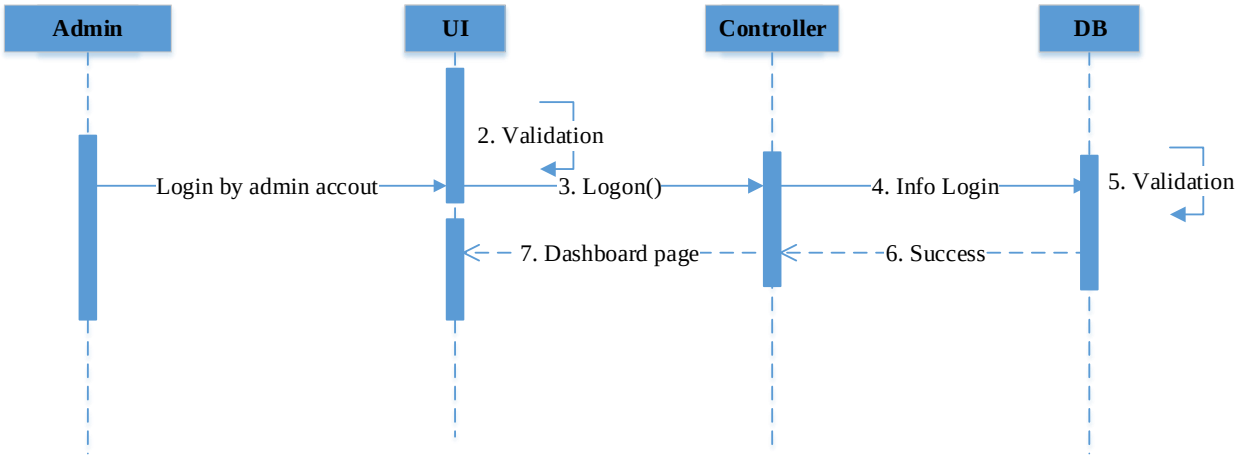
### 6.5 Trace Booking




| | |
|---|---|
| **Author** | |
| **Use case name** | Trace Booking |

| | | |
|---|---|---|
| **Actor** | Passenger | |
| **Description** | Passenger wanna view history of booking | |
| **Requirements** | Passenger provide Reservation info to search or logon to website | |
| **Pre-conditions** | Passeger booking success at least once. | |
| **Post -Condition** | Success: All Flight History match info return. | |
| **Basic flow** | **Actor action** | **System Responses** |
| | 1. Passger go to trace booking page input reservation info or logon to website | 2. UI Valadate<br>3. System call GetHistoryFlight() to controller to get data from DB with info that passenger input.<br>4. Controller get match info from DB.<br>5. DB return all match info to controller.<br>6 .Controller return view to UI. |
| **Alternative flow** | | |
| **Exception** | Reservation code and email blank. | There are message to passenger and no result return |

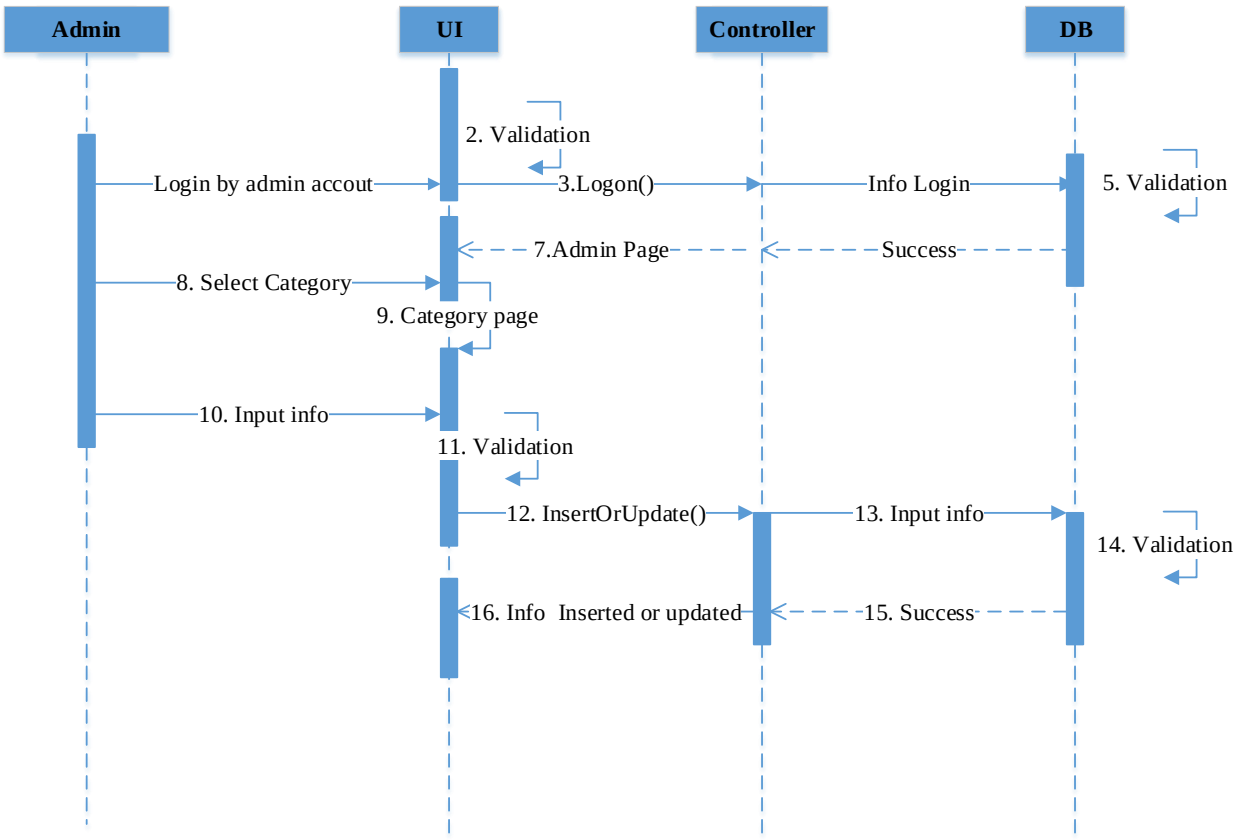
### 6.6 DashBoard




| **Author** | | |
|---|---|---|
| **Use case name** | **DashBoard** | |
| **Actor** | Admin | |
| **Description** | Admin view Dashboard about general web active infomation | |
| **Requirements** | Login by Admin account | |
| **Pre-conditions** | | |
| **Post -Condition** | Success: Dashboard is display. | |
| **Basic flow** | **Actor action** | **System Responses** |
| | 1. Login by admin account. | 2. UI Valadate<br>3. System call Logon() to controller to get data from DB with info that passenger input.<br>4. Controller get logon info from DB.<br>5. Validate logon info<br>6. DB return match logon info to controller.<br>7 .Controller return view Dashboard to UI. |

| | | |
|---|---|---|
| **Alternative flow** | | |
| **Exception** | Account is not admin | There are message to passenger and no result return |

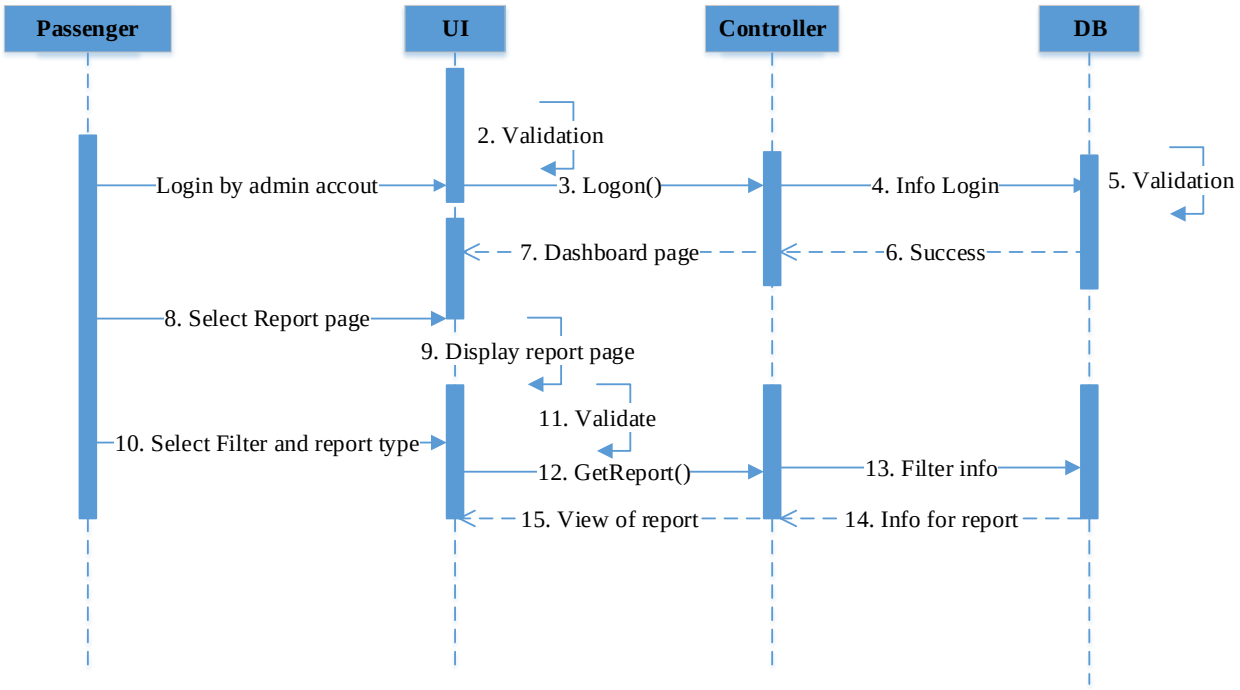
### 6.7 Category




| | |
|---|---|
| **Author** | |
| **Use case name** | **Category** |
| **Actor** | Admin |
| **Description** | Admin can to manage cateogry of website |
| **Requirements** | Login by Admin account |
| **Pre-conditions** | |
| **Post -Condition** | Success: Category is insert or update |

| | **Actor action** | **System Responses** |
|---|---|---|
| **Basic flow** | 1. Login by admin account.<br>8. Admin select Category.<br>10. Admin input information to insert or update then submit. | 2. UI Valadate<br>3. System call Logon() to controller to get data from DB with info that passenger input.<br>4. Controller get logon info from DB.<br>5. Validate logon info<br>6. DB return match logon info to controller.<br>7 .Controller return admin page to UI.<br>9. Category page display.<br>11. Validate input.<br>12. Call InsertOrUpdate() to controller.<br>13. Controller send info to db.<br>14. Validate info in DB.<br>15. Return message success.<br>16. Insert or update info success display in UI. |
| **Alternative flow** | | |
| **Exception** | Account is not admin, Input value is not valid | There are message to passenger and no result return |

## 6.8 Report




| **Author** | | |
|---|---|---|
| **Use case name** | **Category** | |
| **Actor** | Admin | |
| **Description** | Admin can to manage cateogry of website | |
| **Requirements** | Login by Admin account | |
| **Pre-conditions** | | |
| **Post -Condition** | Success: Report information is display. | |
| **Basic flow** | **Actor action** | **System Responses** |
| | 1. Login by admin account.<br>8. Admin select Report page.<br>10. Admin input filter and select report type | 2. UI Valadate<br>3. System call Logon() to controller to get data from DB with info that passenger input.<br>4. Controller get logon info from DB. |

| | | |
|---|---|---|
| | | 5. Validate logon info<br>6. DB return match logon info to controller.<br>7 .Controller return admin page to UI.<br>9. Report page display.<br>11. Validate input.<br>12. Call GetReport() to controller.<br>13. Controller send info to db.<br>14. DB return info for report.<br>15. Report info display in UI |
| **Alternative flow** | | |
| **Exception** | Account is not admin,<br>Input value is not valid | There are message to passenger and no result return |

**Review 2 - Task Sheet**

| **Project Ref. No.:**<br>**Project title: Flight Booking** | | | **Activity Plan Prepared by: Trí** | | **Date of Preparation of Activity Plan** | | |
|---|---|---|---|---|---|---|---|
| **Sr.No.** | **Task** | | **Start Date** | **End Date** | **Actual day** | **Member's Name** | **status** |
| 1 | Use-Case | Sign up | 19/02/2019 | 19/02/2019 | 1 | Phát | Complete |
| 2 | | Login | 19/02/2019 | 19/02/2019 | 1 | Phát | Complete |
| 3 | | Trace Booking | 20/02/2019 | 20/02/2019 | 1 | Phát | Complete |
| 4 | | Category | 21/02/2019 | 21/02/2019 | 1 | Phát | Complete |
| 5 | | Search flight | 19/02/2019 | 19/02/2019 | 1 | Trí | Complete |
| 6 | | Report | 20/02/2019 | 20/02/2019 | 1 | Trí | Complete |
| 7 | | Booking | 20/02/2019 | 20/02/2019 | 1 | Lâm | Complete |
| 8 | | Dashboard | 21/02/2019 | 22/02/2019 | 1 | Lâm | Complete |
| 9 | DFD | Sign up | 19/02/2019 | 19/02/2019 | 1 | Phát | Complete |
| 10 | | Login | 19/02/2019 | 19/02/2019 | 1 | Phát | Complete |
| 11 | | Trace Booking | 20/02/2019 | 20/02/2019 | 1 | Phát | Complete |
| 12 | | Category | 21/02/2019 | 21/02/2019 | 1 | Phát | Complete |
| 13 | | Search flight | 19/02/2019 | 19/02/2019 | 1 | Trí | Complete |
| 14 | | Report | 20/02/2019 | 20/02/2019 | 1 | Trí | Complete |
| 15 | | Booking | 20/02/2019 | 20/02/2019 | 1 | Lâm | Complete |
| 16 | | Dashboard | 21/02/2019 | 22/02/2019 | 1 | Lâm | Complete |
| 17 | Sequence | Sign up | 19/02/2019 | 19/02/2019 | 1 | Phát | Complete |
| 18 | | Login | 19/02/2019 | 19/02/2019 | 1 | Phát | Complete |
| 19 | | Trace Booking | 20/02/2019 | 20/02/2019 | 1 | Phát | Complete |
| 20 | | Category | 21/02/2019 | 21/02/2019 | 1 | Phát | Complete |
| 21 | | Search flight | 19/02/2019 | 19/02/2019 | 1 | Trí | Complete |
| 22 | | Report | 20/02/2019 | 20/02/2019 | 1 | Trí | Complete |
| 23 | | Booking | 20/02/2019 | 20/02/2019 | 1 | Lâm | Complete |
| 24 | | Dashboard | 21/02/2019 | 22/02/2019 | 1 | Lâm | Complete |

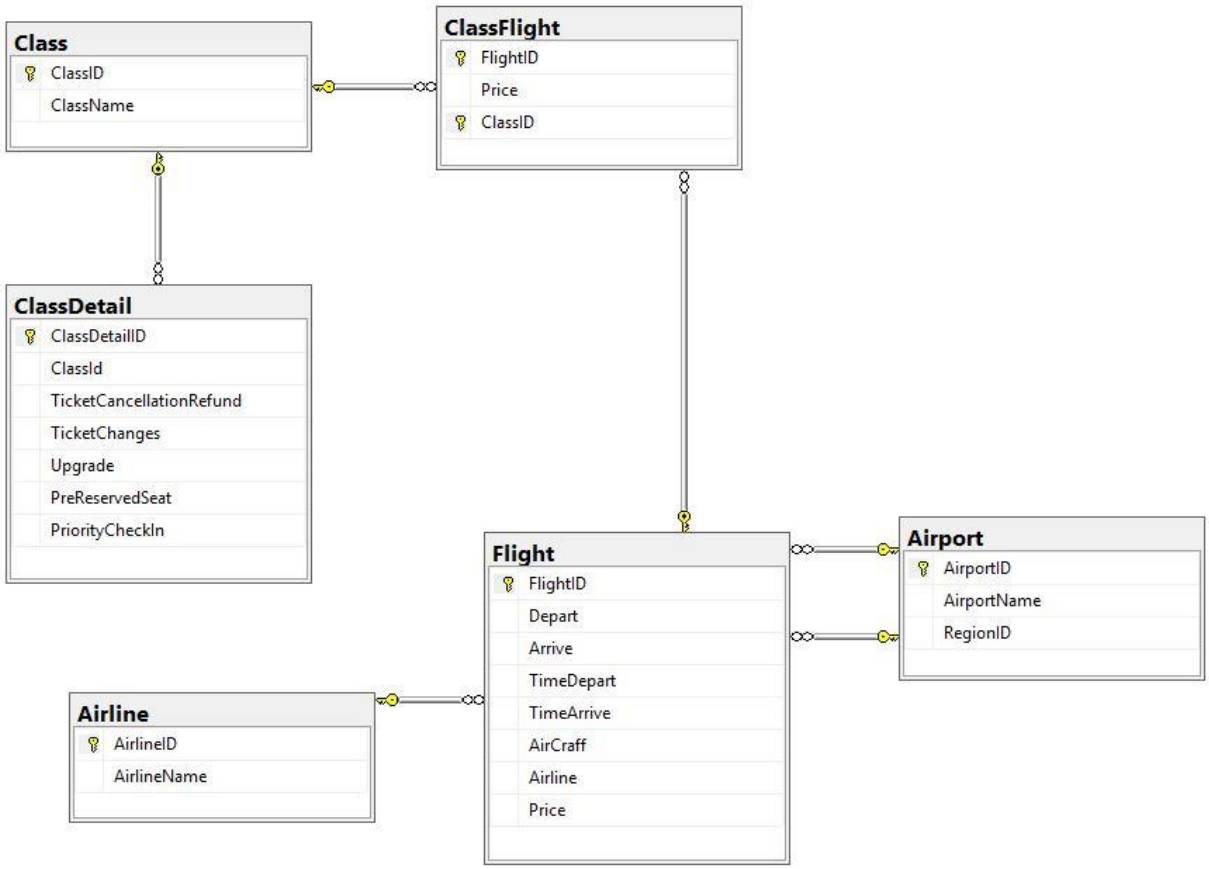
**Group Leader** **Faculty**

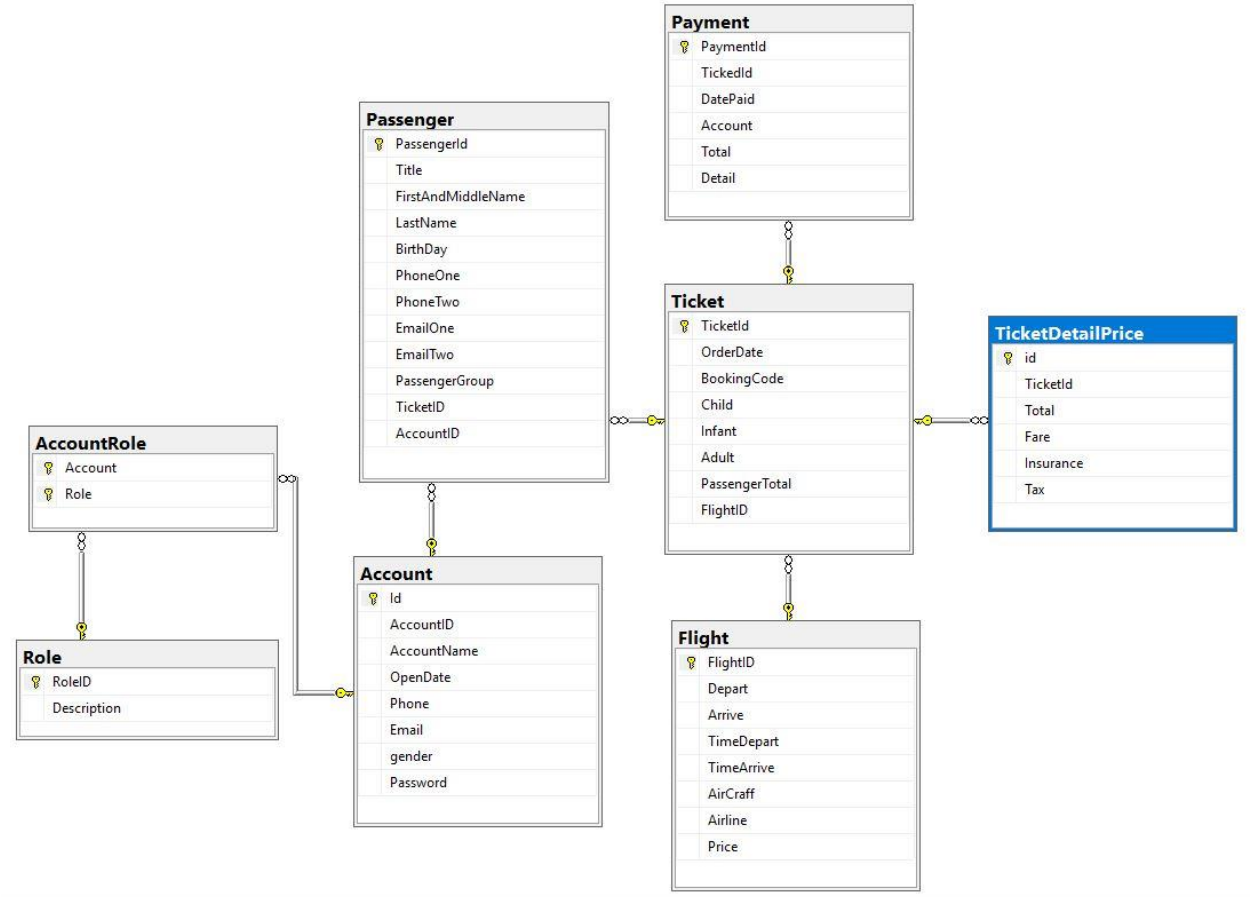
Nguyen Huu Tri Nguyen Hoang Thao

# REVIEW 3

## 7. Entity Relationship (E-R) Diagram

### 7.1 E-R Diagram






## 7.2 Table

### 7.2.1 Region

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| RegionID | varchar(15) | Primary Key | The identification of Region |
| RegionName | Nvarchar(50) | Not null | |

### 7.2.2 Airport

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| AirportID | varchar(15) | PK | The identification of Airport |
| AirportName | Nvarchar(50) | Not null | |
| RegionID | varchar(15) | Not null, FK | |

### 7.2.3 Airline

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| AirlineID | varchar(15) | PK | The identification of Airline |
| AirlineName | Nvarchar(50) | Not null | |

### 7.2.4 Flight

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| FlightID | varchar(15) | PK | The identification of Flight |
| Depart | varchar(15) | Not null,FK | |
| Arrive | varchar(15) | Not null, FK | |
| TimeDepart | Datetime | Not null | |
| TimeArrive | Datetime | Not null | |
| AirCraff | Varchar(50) | Not null | |
| Airline | int | Not null, FK | |
| Price | float | Not null | |

### 7.2.5 Class

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| ClassID | int | Primary Key | The identification of Region |
| ClassName | Varchar(50) | Not null | |

### 7.2.6 ClassDetail

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| ClassDetailID | int | Primary Key | The identification |
| ClassId | Varchar(50) | Not null, FK | |
| TicketCancellationRefund | bit | Not null | |
| TicketChanges | bit | Not null | |
| Upgrade | bit | Not null | |
| PreReservedSeat | bit | Not null | |
| PriorityCheckIn | bit | Not null | |

### 7.2.7 ClassFlight

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| FlightID | int | PK,FK | The identification |
| ClassID | Varchar(50) | PK,FK | |
| Price | float | Not null | |

### 7.2.8 Ticket

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| TicketId | int | PK,FK | The identification |
| OrderDate | datetime | Not null | |
| BookingCode | float | Not null | |
| Child | tinyint | Not null | |
| Infant | tinyint | Not null | |
| Adult | tinyint | Not null | |
| PassengerTotal | tinyint | Not null | |
| FlightID | int | Not null,FK | |

### 7.2.9 Passenger

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| PassengerId | int | PK,FK | The identification |
| Title | Char(3) | Not null | |
| FirstAndMiddleName | varchar | Not null | |
| LastName | varchar | Not null | |
| BirthDay | varchar | Not null | |
| PhoneOne | varchar | Not null | |
| PhoneTwo | varchar | | |
| EmailOne | varchar | Not null | |
| EmailTwo | varchar | | |
| PassengerGroup | varchar | Not null | |
| TicketID | Int | Not null,FK | |
| AccountID | Int | FK | |

### 7.2.10 Payment

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| PaymentId | int | PK | The identification |
| TickedId | int | Not null | |
| DatePaid | datetime | Not null | |
| Account | int | Not null | |
| Total | real | Not null | |
| Detail | nvarchar(500) | Not null | |
| PaymentId | int | Not null,FK | |
| TickedId | int | Not null,FK | |
| DatePaid | datetime | Not null | |
| Account | varchar | Not null | |
| TicketID | Int | Not null,FK | |
| AccountID | Int | Not Null,FK | |

### 7.2.11 TicketDetailPrice

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| id | int | PK | The identification |
| TicketId | int | Not null, FK | |
| Total | float | Not null | |
| Fare | float | Not null | |
| Insurance | float | Not null | |
| Tax | float | Not null | |

### 7.2.12 Account

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| Id | int | PK | The identification |
| AccountID | varchar(50) | Unique | |
| AccountName | nvarchar(100) | Not null | |
| OpenDate | datetime | Not null | |
| Phone | varchar(50) | Not null | |
| Email | varchar(50) | Unique | |
| gender | bit | | |
| Password | Varchar(200) | Not null | |

### 7.2.13 Role

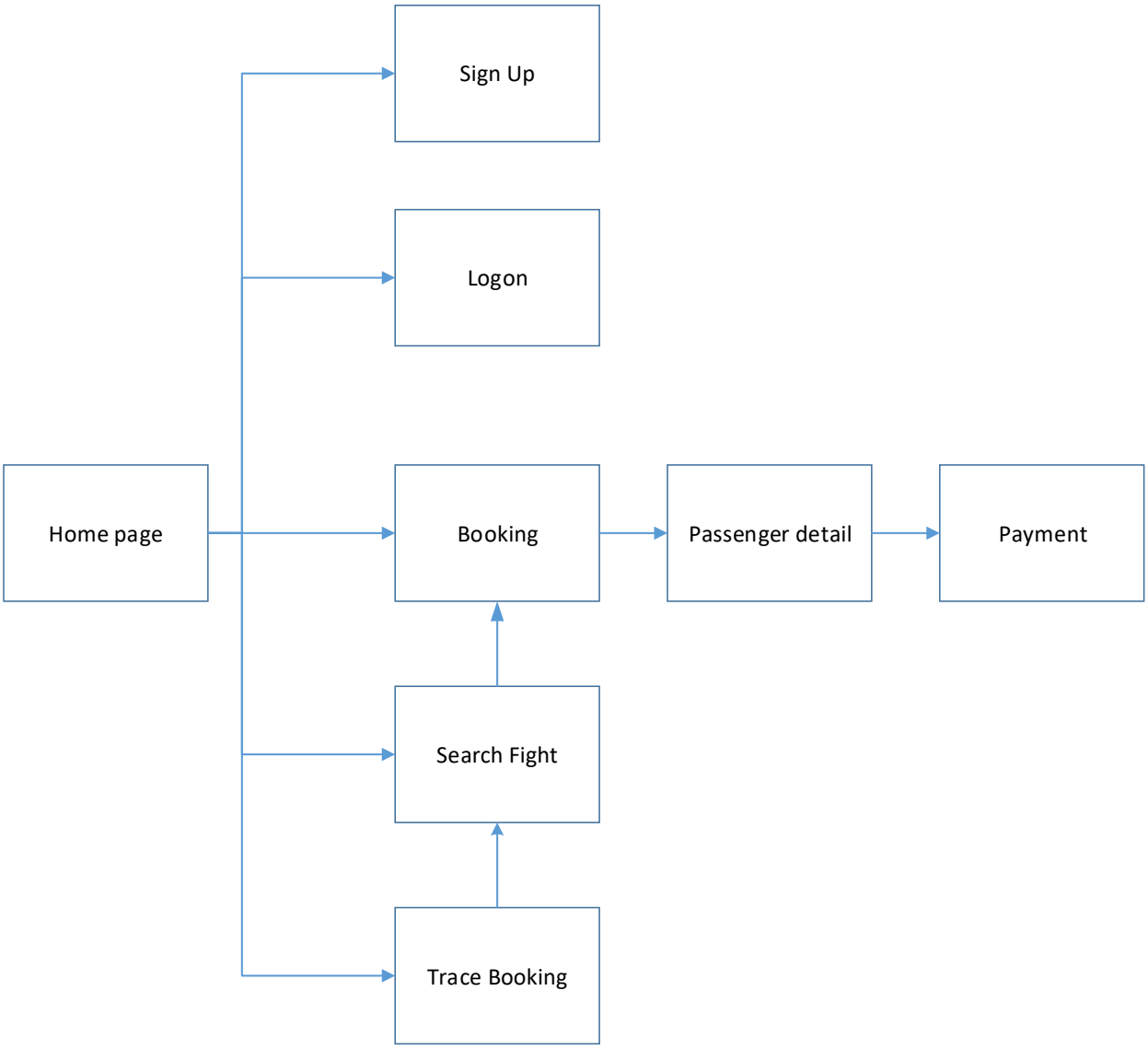
| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| RoleID | varchar(15) | PK | The identification |
| Description | nvarchar(50) | | |

### 7.2.14 AccountRole

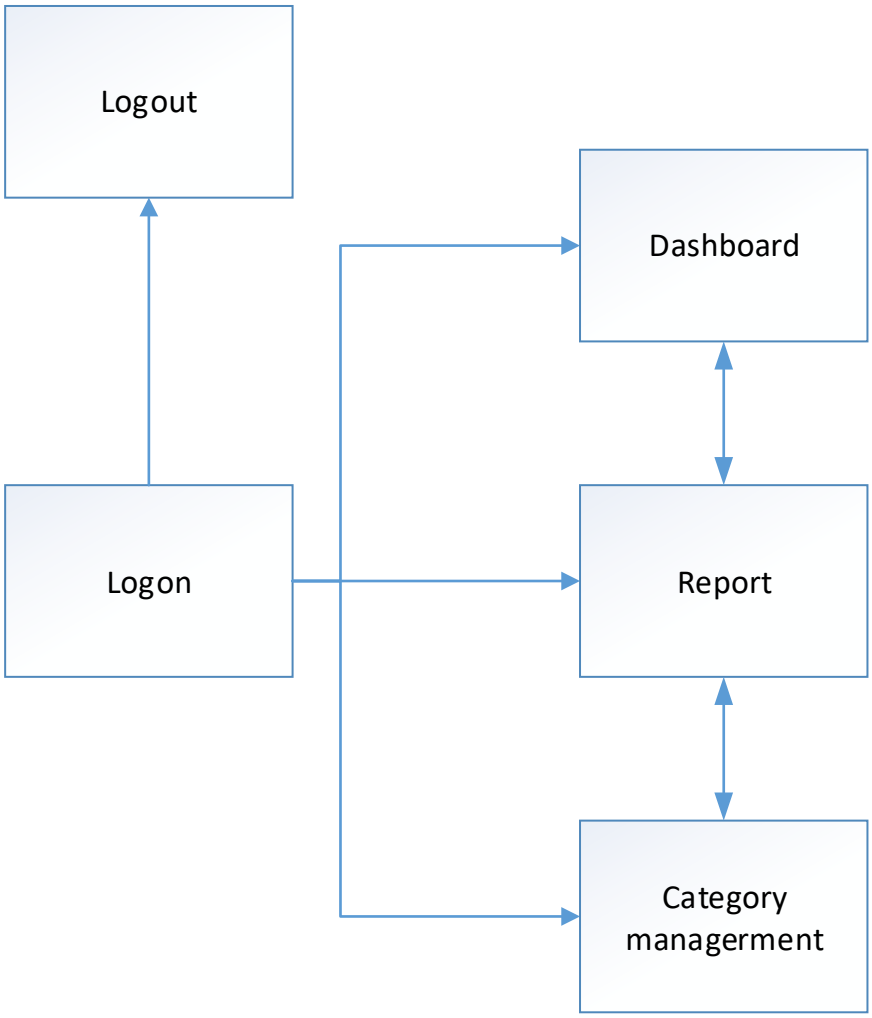
| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| Account | int | PK,FK | The identification |
| Role | varchar(15) | PK,FK | |

## 8. Site map

### 8.1 Passenger

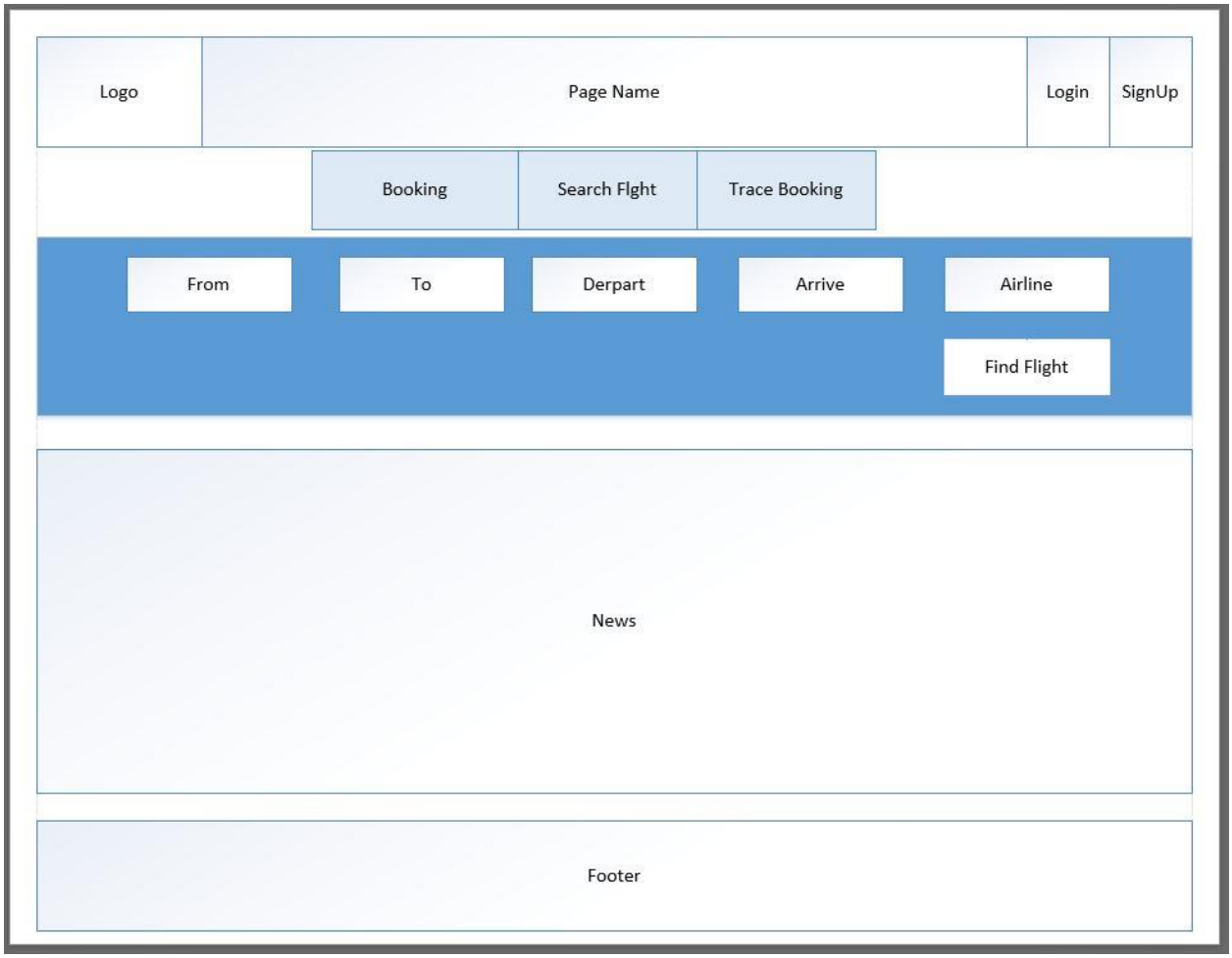


### 8.2 Admin

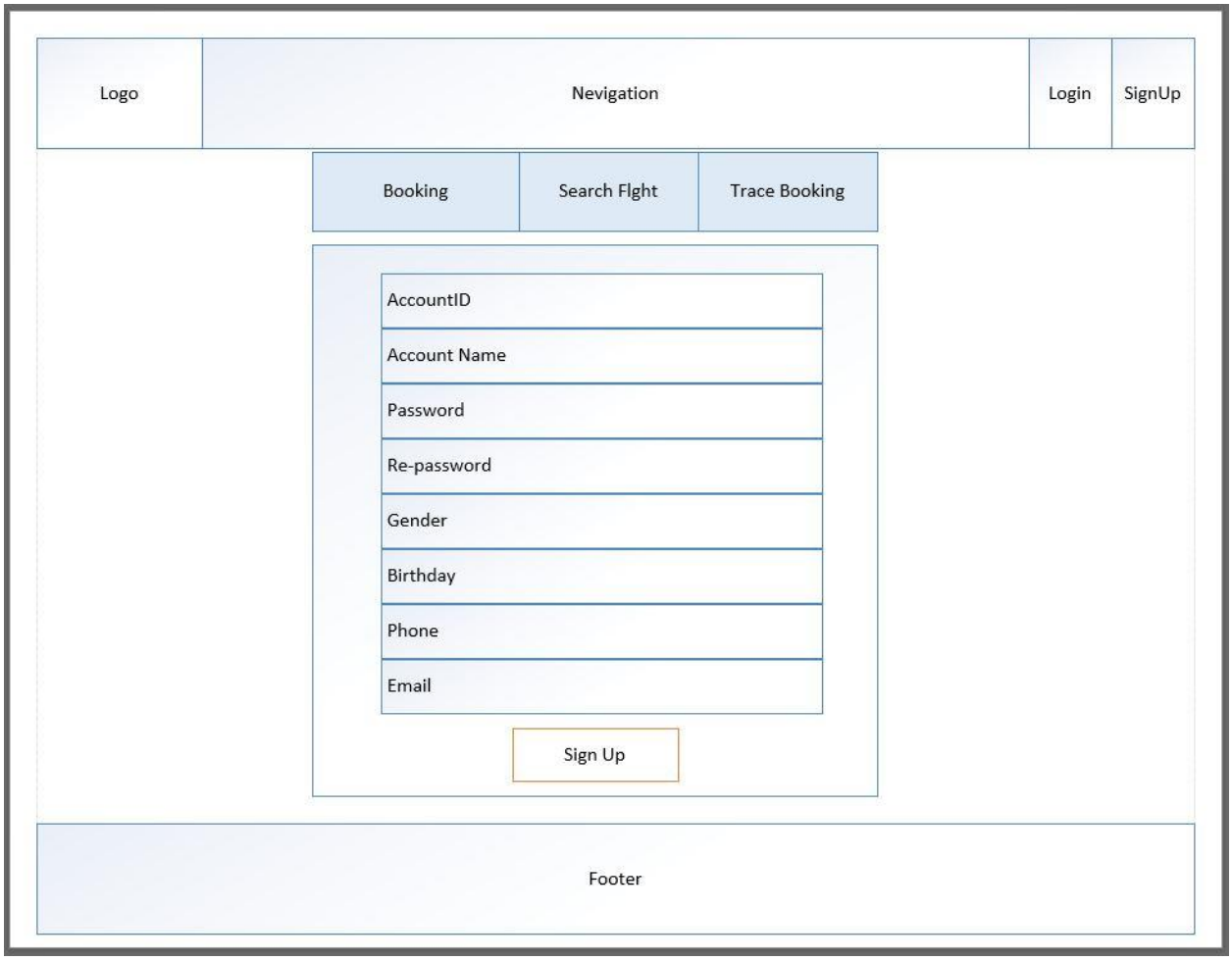


## 9. UI

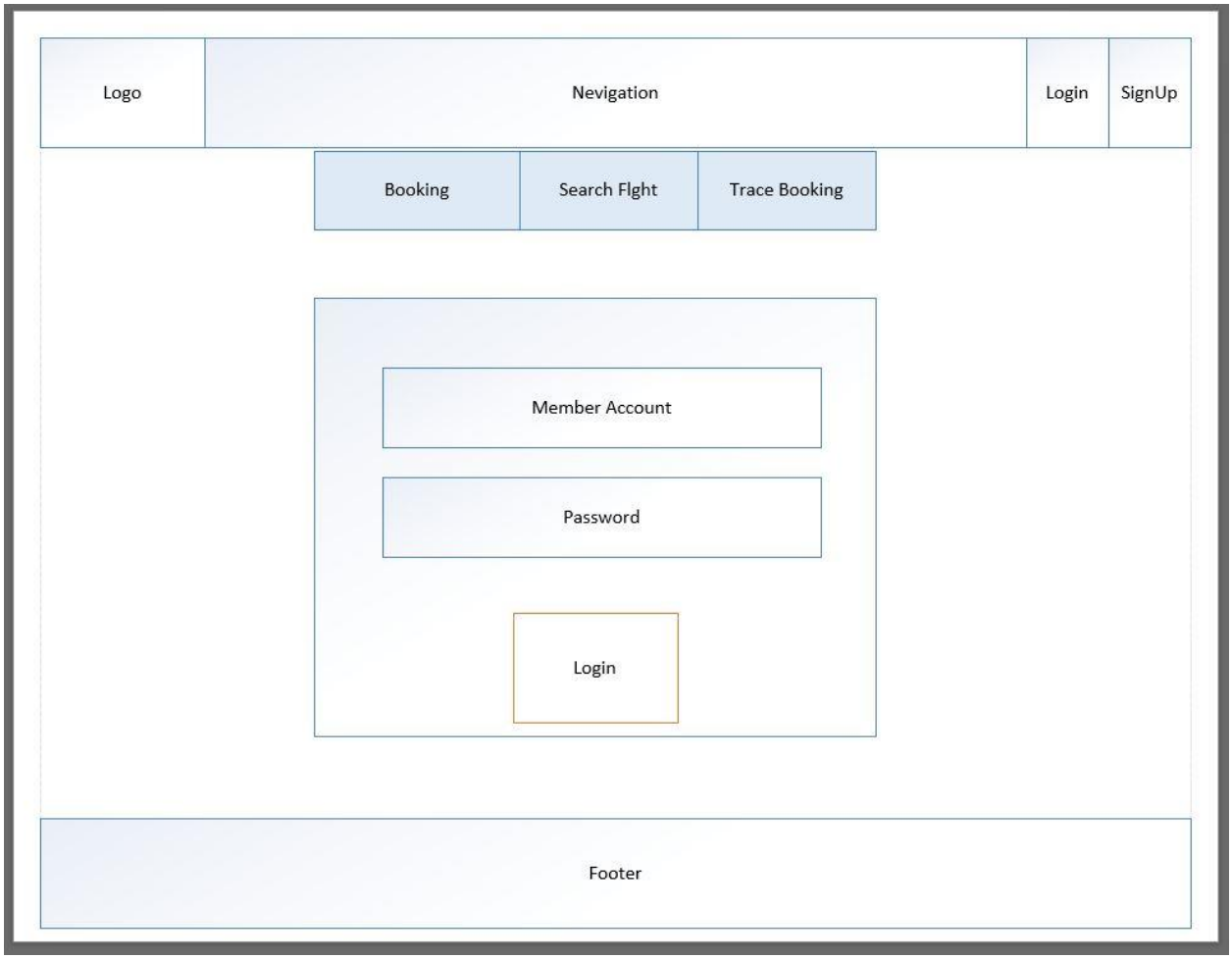
### 9.1 Home Page



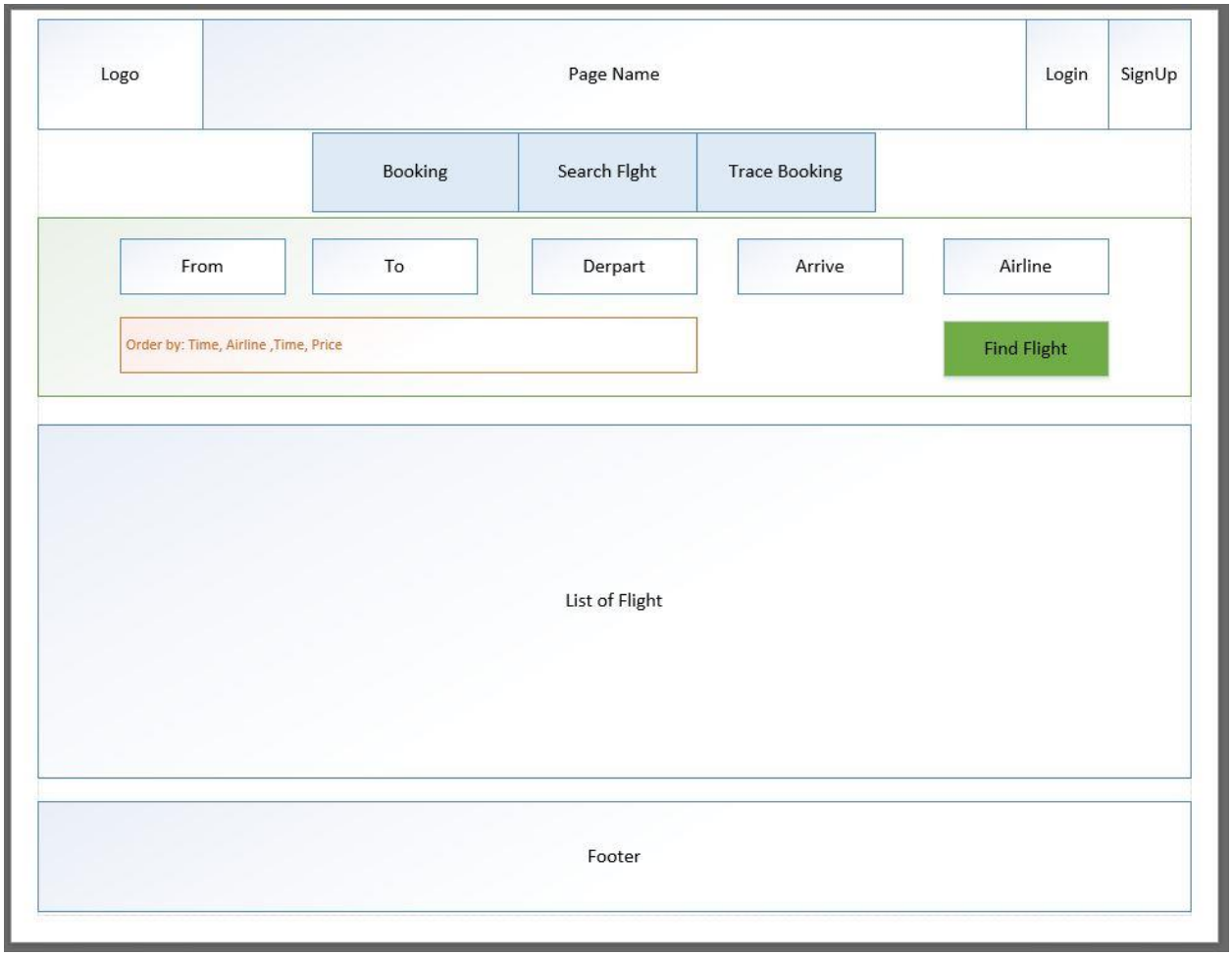
## 9.2 Sign Up


Logo
Nevigation
Login
SignUp
Booking
Search Flght
Trace Booking
AccountID
Account Name
Password
Re-password
Gender
Birthday
Phone
Email
Sign Up
Footer

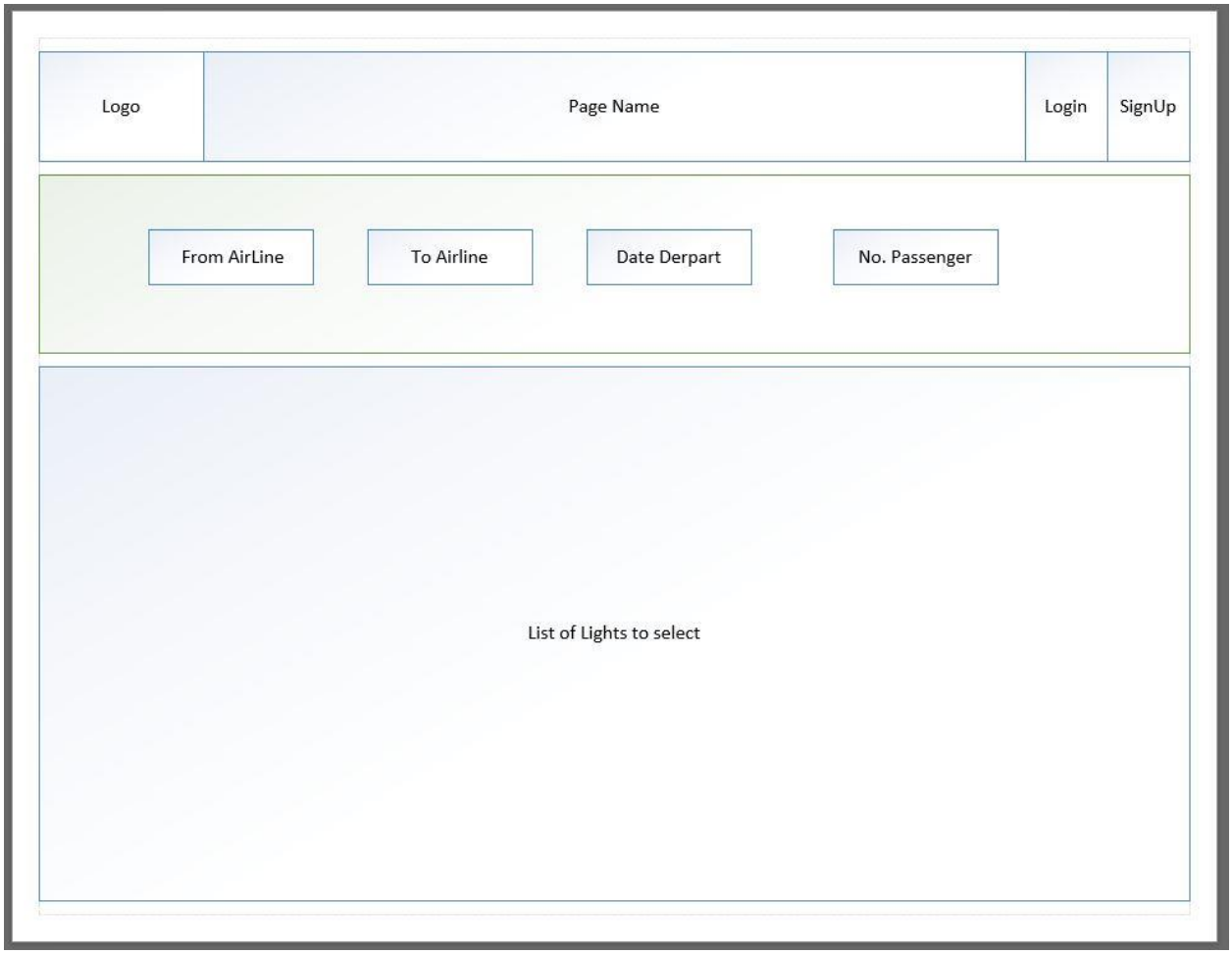
## 9.3 Logon


Logo
Nevigation
Login
SignUp
Booking
Search Flght
Trace Booking
Member Account
Password
Login
Footer

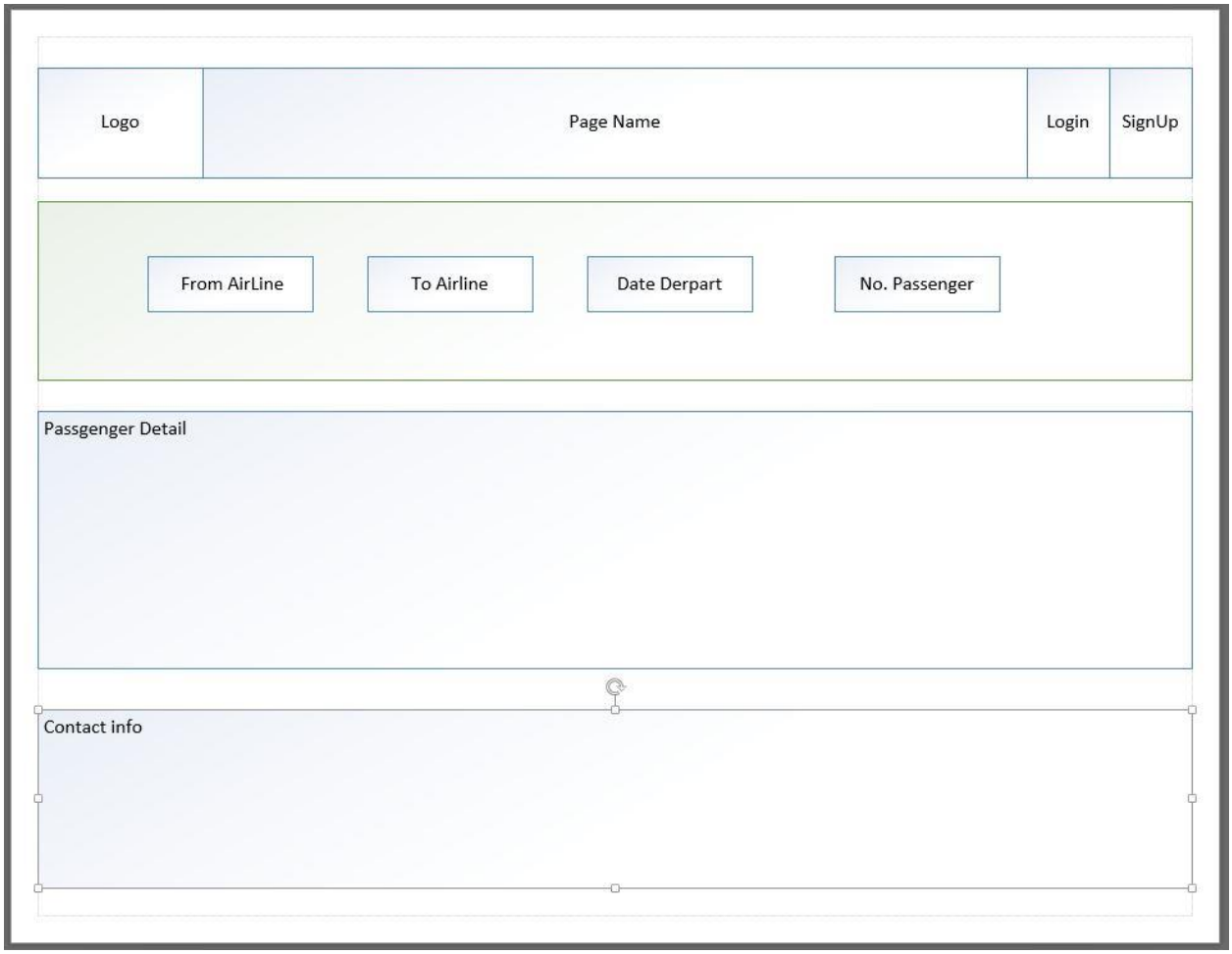
## 9.4 Search Flight



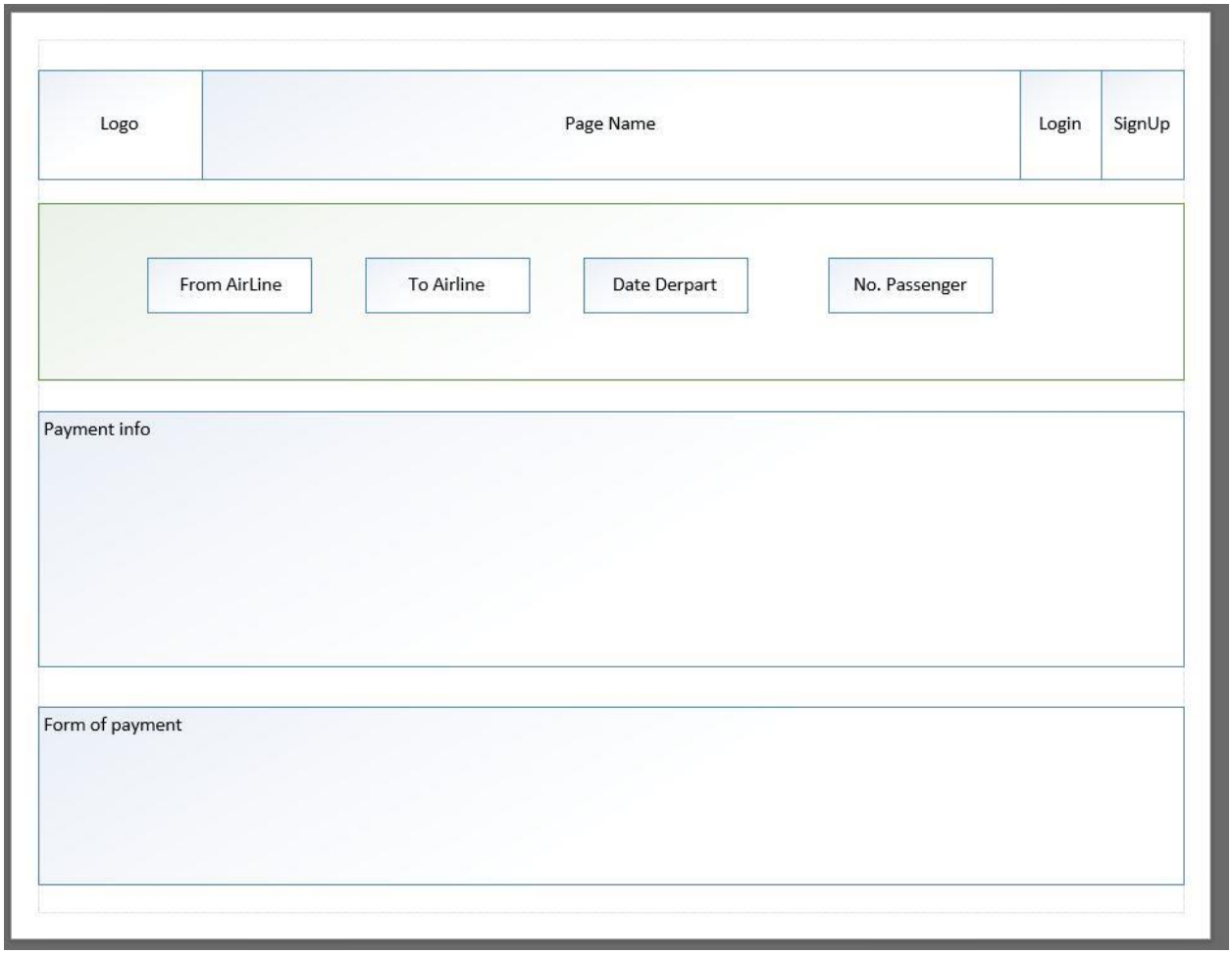
## 9.5 Booking


Logo
Page Name
Login
SignUp
From AirLine
To Airline
Date Derpart
No. Passenger
List of Lights to select

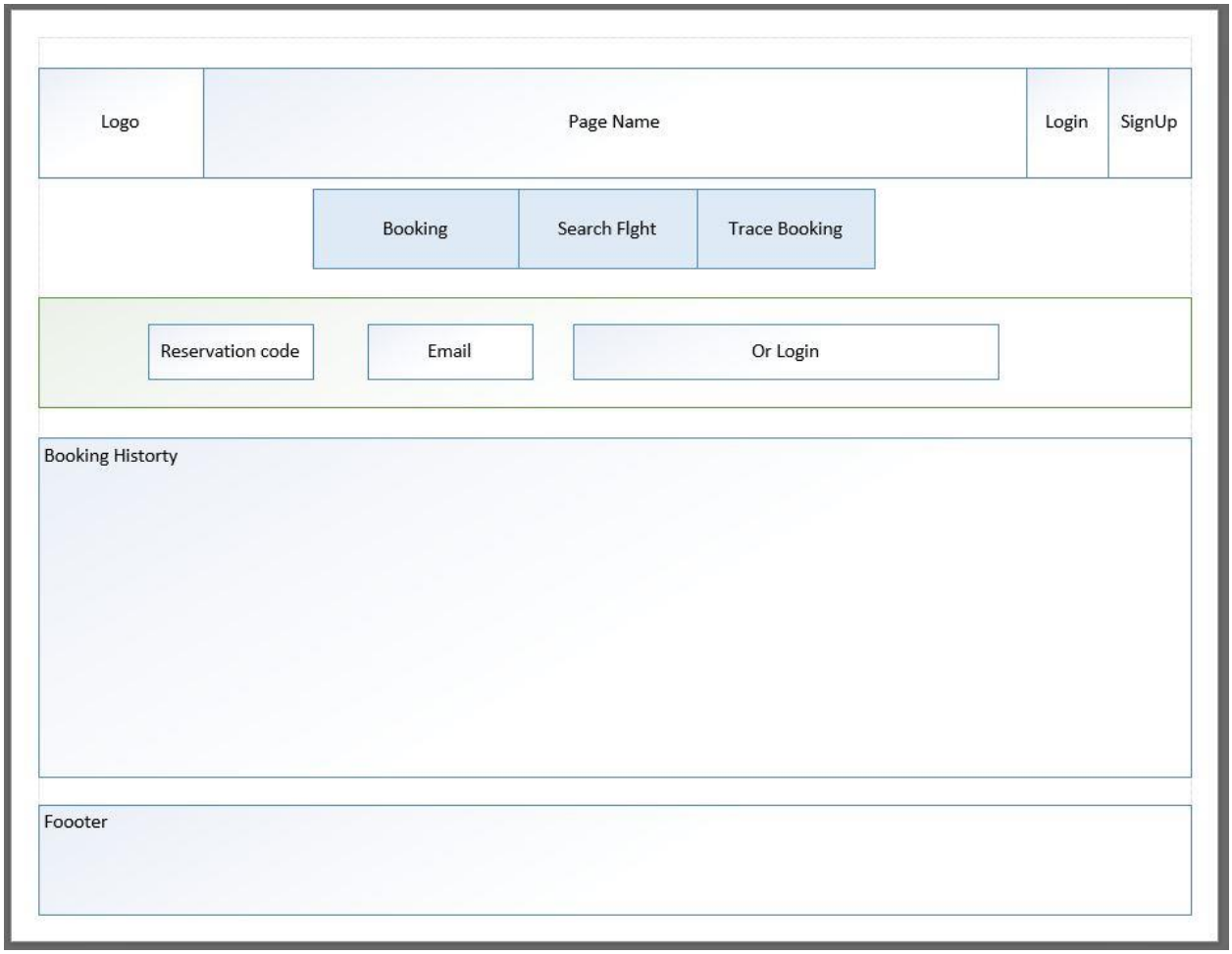
## 9.6 Passenger Info



## 9.7 Payment

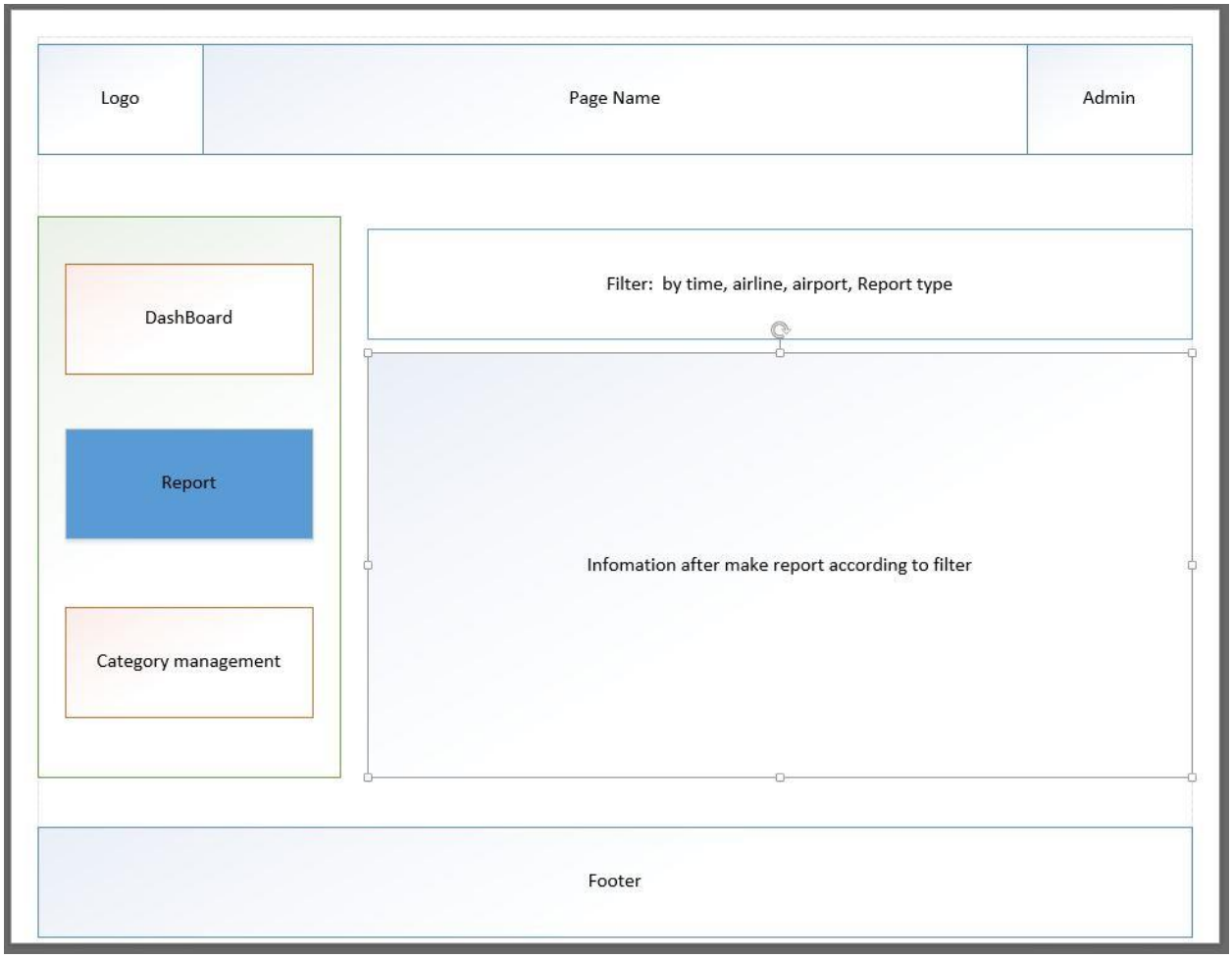

Logo
Page Name
Login
SignUp
From AirLine
To Airline
Date Derpart
No. Passenger
Payment info
Form of payment

## 9.8 Trace Booking

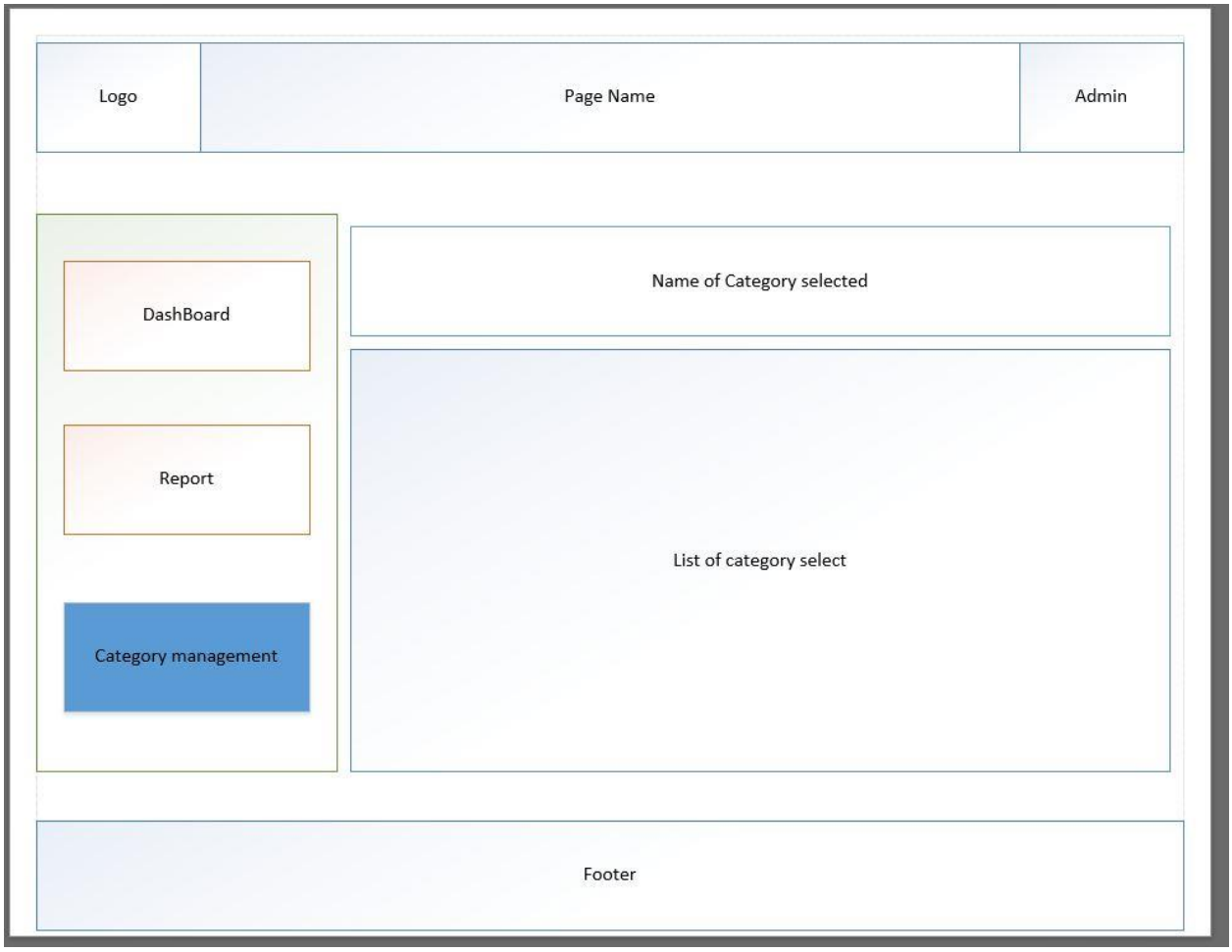


## 9.9 Dashboard

## 9.10 Report


Logo
Page Name
Admin
DashBoard
Report
Category management
Filter: by time, airline, airport, Report type
Infomation after make report according to filter
Footer

## 9.11 Category



**Review 3 - Task Sheet**

| Project Ref. No.:<br>Project title: Flight Booking | | Activity Plan Prepared by: Trí | | Date of Preparation of Activity Plan | | |
|---|---|---|---|---|---|---|
| **Sr.No.** | **Task** | **Start Date** | **End Date** | **Actual day** | **Member's Name** | **status** |
| 1 | Draw Entity relationship diagram | 23/02/2019 | 24/02/2019 | 2 | Trí | Complete |
| 2 | Describing table. | 24/02/2019 | 24/02/2019 | 1 | Trí | Complete |
| 3 | Frontend - Home page implementation | 25/02/2019 | 25/02/2019 | 1 | Lâm | Complete |
| 4 | Frontend – Sign Up page implementation | 25/02/2019 | 25/02/2019 | 1 | Phát | Complete |
| 5 | Frontend - Logon page implementation | 25/02/2019 | 25/02/2019 | 1 | Phát | Complete |
| 6 | Frontend – Trace Booking page implementation | 26/02/2019 | 26/02/2019 | 1 | Phát | Complete |
| 7 | Frontend – Search Flight page implementation | 25/02/2019 | 25/02/2019 | 1 | Trí | Complete |
| 8 | Frontend – Search Flight Result page implementation | 26/02/2019 | 26/02/2019 | 1 | Trí | Complete |
| 9 | Frontend – Booking page implementation | 27/02/2019 | 27/02/2019 | 1 | Lâm | Complete |
| 10 | Frontend – Passenger imformation page implementation | 27/02/2019 | 27/02/2019 | 1 | Lâm | Complete |
| 11 | Frontend – Payment imformation page implementation | 27/02/2019 | 27/02/2019 | 1 | Lâm | Complete |
| 12 | Backend – Insert/Update Region | 27/02/2019 | 27/02/2019 | 1 | Phát | Complete |
| 13 | Backend – Insert/Update Airport | 27/02/2019 | 27/02/2019 | 1 | Phát | Complete |
| 14 | Backend – Insert/Update Airline | 27/02/2019 | 27/02/2019 | 1 | Phát | Complete |
| 15 | Backend – Get information for Dashboard | 28/02/2019 | 28/02/2019 | 1 | Lâm | Complete |
| 16 | Backend – Get information for Statistical Reports | 01/03/2019 | 01/03/2019 | 1 | Trí | Complete |
| 17 | Report – Ticket form | 27/02/2019 | 27/02/2019 | 1 | Lâm | Complete |
| 18 | Report - Trace booking by time | 29/02/2019 | 29/02/2019 | 1 | Phát | Complete |
| 19 | Report – List flight by Airport, Airline, Time | 29/02/2019 | 01/03/2019 | 2 | Trí | Complete |

| **Group Leader** | **Faculty** |
| --- | --- |
| Nguyen Huu Tri | Nguyen Hoang Thao |