Software Requirements Specification Version 1.0

Car Park

Fresher Academy

2018

1

**Table of Contents**

Table of Contents...................................................................................................................................2 **1. INTRODUCTION** ...........................................................................................................................3 1.1. Purpose ..................................................................................................................................3 ***1.2. In scope***..................................................................................................................................3 **2. OVERVIEW**....................................................................................................................................3 ***2.1. Actors*** .....................................................................................................................................3 2.2. System Use Case Diagram.....................................................................................................4 2.2.2. Car Park operation administration staff......................................................................5 **3. Functional Description**...............................................................................................................5 3.1 Functional Requirements .....................................................................................................5 3.1.2. Booking office management .......................................................................................10 3.1.3. Parking lot management.............................................................................................13 3.1.4. Car management..........................................................................................................17 3.1.5. Trip management........................................................................................................20 3.1.6. Ticket management.....................................................................................................24 3.2 Detailed Non-Functional Requirements............................................................................29 3.2.1 Logical Structure of the Data ......................................................................................29 3.2.2 Security.........................................................................................................................30 3.2.3 Performance.................................................................................................................30 3.2.4 Technial Requirement.................................................................................................30

2

**1. INTRODUCTION**

***1.1. Purpose***

The purpose of this document describes in detail the functions of "Car Park Management System". It shows the function and builds a complete system. It clearly explains the requirement of the system, gives an overview of screens and interacts with other systems or applications. And this document is for presentation with the customer to improve.

***1.2. In scope***

− The web-based system of car park management makes managing information fast, convenient and inexpensive, which provides top management services and solutions on the Internet. The car park management system is designed to meet the needs of employee management and operation administration at car park, as well as services for individuals business organizations in need.

− Main user:

+ Human resource management (HRM) staff.

+ Car Park management staff.

The website allows to manage car park employees, operation administration, parking registration, booking office, check in/ check out of cars, car trip and parking lot.

**2. OVERVIEW**

***2.1. Actors***

The diagram below describes all actors participating in "Car Park Management". Each actor has a different role in the system. Currently, there are the following actors: Human Resource Management staff, Car park operation administration staff.

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Human Resource Management staff | The staff has right to search, add, delete, edit employee information, statistic report |
| Car | park the staff has right to: |

3

|  |  |
| --- | --- |
| operation  administration staff | - Manage parking registration, including searching, adding, deleting, editing, and reporting statistics.  - Manage booking office, including searching, adding, deleting, editing ticket information and reporting statistics.  - Manage car park, including adding, deleting, editing car park information. |

***2.2. System Use Case Diagram***

The diagram below provides an overview of the groups of functions and actors involved in building the *Da Nang car park management website*:

***2.2.1. Human resource management (HRM) staff***

View employee list

Add employee

Edit employee

HRM Staff

Delete employee

4

***2.2.2. Car park operation administration staff***

Car management

Booking office

management

Parking lot

management

Car park operation

administration staff

Ticket management

Trip management

**3. Functional Description**

***3.1Functional Requirements***

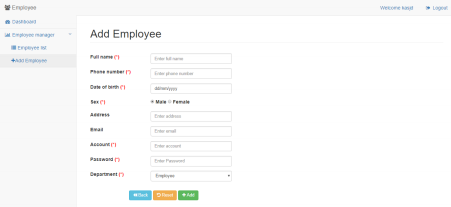
***3.1.1. Employee management***

|  |  |
| --- | --- |
| Use Case Name | Employee management |
| Use Case ID | UC02. |
| High Level Requirement Ref | Allows to manage information of employee such as creating, viewing details, displaying employee list |
| Actor | Admin. |
| Description | All of employee information is accessed from database and displayed on screen. The user can create, view details, display employee list. |
| Trigger | NA. |
| Pre- | User clicks Employee management on Menu |

5

|  |  |
| --- | --- |
| condition |  |
| Post  processing |  |

***3.1.1.1. Add employee***

*******Screen of adding employee*

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Add employee | | |
| **Description** | Allows users to add employee | | |
| **Screen Access** | Amin clicks **Employee management -> Employee List -> Add employee** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Full name | Text field – String (50) |  |  |
| Date of birth | Date Picker |  |  |
| Sex | RadioButton |  |  |
| Address | Text field – String (50) |  |  |
| Phone number | Text field – String (50) |  |  |
| Email | Text field – String (50) |  |  |

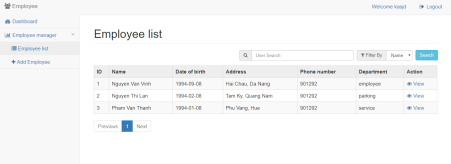
6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Account | | Text field – String (50) |  | |  | |
| Password | | Password – String(100) |  | |  | |
| Department | | ComboBox |  | |  | |
| Back | | Button |  | | The user clicks Back to return to the previous page | |
| Reset | | Button |  | | The user clicks Reset to delete all inputted data. | |
| Add | | Button |  | | The user clicks Add to add employee into Database | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Back | If user clicks Back, then system forwards to the previous screen. | | | Display the previous  screen. | | When database connection has error -> Display message “Database connection error” |
| Reset | If user clicks Reset, then system deletes inputted  information. | | | Delete all inputted  information | | When database connection has error -> Display message “Database connection error” |
| Add | When the user clicks Add, the system validates data. If data is valid, then save inputted data on the database, otherwise display error message. | | | Back to screen of Employee list and display the message "Add employee successfully". | | When inputting password with wrong format -> Display the message "Password must include uppercase, lowercase and number." |
| Validation | The password must have at least 6 characters,  including  uppercase, | | |  | |  |

7

|  |  |  |  |
| --- | --- | --- | --- |
|  | lowercase and number. |  |  |

***3.1.1.2. Display employee list***

*******Screen of employee list*

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Employee list | | |
| **Description** | Display employee list | | |
| **Screen Access** | Admin clicks **Employee management -> Employee list ->List** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Search Box |  |  |
| Search criteria | ComboBox |  |  |
| Search | Button |  |  |
| ID | Label - Integer |  |  |
| Name | Link - String (50) |  | Display name of employee. When the user clicks employee name, then forward to information page of this employee |
| Date of birth | Label - Date |  |  |
| Address | Label - String(50) |  |  |
| Phone number | Label - String(50) |  |  |

8

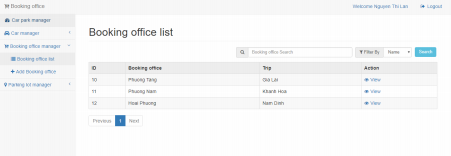
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Department | | Label - String(50) |  | |  | |
| Prev | | Link |  | | When user clicks this link, then display employee list at the previous page | |
| 1 | | Link |  | | When the user clicks this link, then display employee list at page 1 | |
| 2 | | Link |  | | When the user clicks this link, then display employee list at page 2 | |
| 3 | | Link |  | | When the user clicks this link, then display employee list at page 3 | |
| Next | | Link |  | | When user clicks this link, then display employee list at the next page | |
| View | | Link |  | | When user clicks this link, then display form information employee show the current information and perform action update or delete employee | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Search | When user clicks Search, the system will search employee according to inputted information. | | | If the user does not input any data to search, the system will display all employees on employee list table.  Otherwise, | | When database connection has error -> Display message  “Database  connection error”  When the system does not find out |

9

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | display search result on the employee list table. | any employee, then display message “No matches”  . |

***3.1.2. Booking office management***

***3.1.2.1. Booking office list***

*******Screen of booking office list*

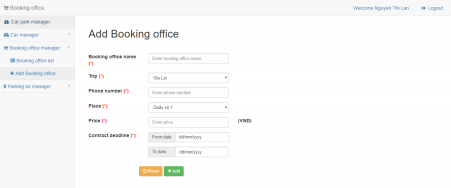
|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Booking office list | | |
| **Description** | Display booking office list | | |
| **Screen Access** | Admin clicks **Booking office -> List** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Search Box |  | The box to input search data |
| Search criteria | ComboBox |  | The box to select search criteria |
| Search | Button |  |  |
| Booking office | Link - String (50) |  |  |
| Trip | Label - String(50) |  |  |
|  | Label – String(20) |  |  |

10

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Prev | | Link |  | | When user clicks this link, then display booking office list at the previous page | |
| 1 | | Link |  | | When the user clicks this link, then display booking office list at page 1 | |
| 2 | | Link |  | | When the user clicks this link, then display booking office list at page 2 | |
| 3 | | Link |  | | When the user clicks this link, then display booking office list at page 3 | |
| Next | | Link |  | | When user clicks this link, then display booking office list at the next page | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Search | When the user clicks **Search,** the system will search booking office according to inputted data at Search box. | | | If the user does not input any data at Search box, the system will display all booking office on list table.  Otherwise,  display search result on the booking office list table. | | When database connection has error -> Display message  “Database  connection error”  When the system does not find out any booking office, then display message “No matches” |

11

***3.1.2.2. Add booking office***

*******Screen of adding booking office*

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Add booking office | | |
| **Description** | Allows staff to add booking office information | | |
| **Screen Access** | Click Booking office management -> Click “Booking office” -> Click “Add” | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Booking office name | Text field – String (50) |  |  |
| Phone number | Text field – String (50) |  |  |
| Price | Text field – |  |  |

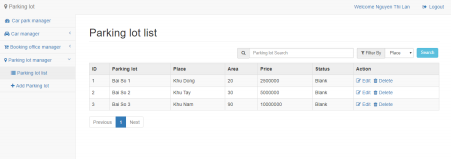
12

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | String (50) |  | | |  |
| Place | | ComboBox |  | | |  |
| Contract  deadline | | Date Picker |  | | |  |
| Reset | | Button |  | | | The user clicks Reset to delete all inputted data |
| Add | | Button |  | | | The user clicks Add to add booking office information and update database. |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | **Failure** | |
| Reset | When user click **Reset**, the system will delete all inputted data | | | Delete all inputted data |  | |
| Add | When the user clicks **Add**, the system will validate data. If data is valid, the system will update data into database, otherwise display error message. | | | Back to Detail screen and display  message “Add booking office successfully” | When inputting amount of money into price box with wrong format -> Display the message "Wrong format of price."  When inputting invalid contract deadline, then the system display message “Invalid contract deadline” | |

***3.1.3. Parking lot management***

***3.1.3.1. Parking lot list***

13

*Screen of Parking lot list page*

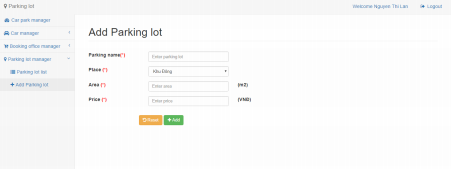
|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Parking lot list | | |
| **Description** | Display parking lot list | | |
| **Screen Access** | Admin clicks **Parking lot -> List** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Search Box |  |  |
| Search criteria | ComboBox |  |  |
| Search | Button |  |  |
| ID | Label  Interger |  |  |
| Parking lot | Link - String (50) |  |  |
| Place | Label - String(50) |  |  |
| Area | Label - String(20) |  |  |
| Status | Label - String(20) |  |  |
|  | Label – String(20) |  |  |
| Prev | Link |  | When user clicks this link, then display parking lot list at the previous page |
| 1 | Link |  | When the user clicks |

14

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | this link, then display parking lot list at page 1 | |
| 2 | | Link |  | | When the user clicks this link, then display parking lot list at page 2 | |
| 3 | | Link |  | | When the user clicks this link, then display parking lot list at page 3 | |
| Next | | Link |  | | When user clicks this link, then display parking lot list at the next page | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Search | When the user clicks **Search,** the system will searchnputted data at search parking lot according to ibox. | | | If the user does not input any data at Search box, the system will  automatically  display all parking lot on ticket list table. Otherwise,  display the search result on ticket list table. | | When database connection has error -> Display message  “Database  connection error”  When the system does not find out any parking lot, then display message “No matches” |

***3.1.3.2. Add parking lot***

15

*Screen of adding parking lot*

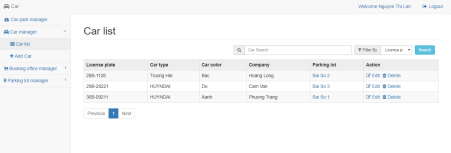
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen** | | Add parking lot | | | | |
| **Description** | | Allow staff to add parking lot information | | | | |
| **Screen Access** | | Click Parking lot management -> Click “Parking lot” -> Click “Add” | | | | |
| **Screen Content** | | | | | | |
| **Item** | | **Type** | **Data** | | | **Description** |
| Parking lot | | Text field – String (50) |  | | |  |
| Place | | ComboBox |  | | |  |
| Area | | Text field – String (50) |  | | |  |
| Price | | Text field – String (10) |  | | |  |
| Reset | | Button |  | | | The user clicks Reset to reset all data into default |
| Add | | Button |  | | | The user clicks Add to add information of parking lot and update into database |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | **Failure** | |

16

|  |  |  |  |
| --- | --- | --- | --- |
| Reset | When user click **Reset**, the system will delete all inputted data | Delete all inputted data |  |
| Add | When the user clicks **Add**, the system will validate data. If data is valid, the system will update data into database, otherwise display error message. | Back to Detail screen and display  message “Add booking office successfully” | When inputted “Area” is not number, the system will display message “Please input area with number format”  When inputted “Price” is not number, the system will display message “Please input price with number format” |

***3.1.4. Car management***

***3.1.4.1. Car list***

******17

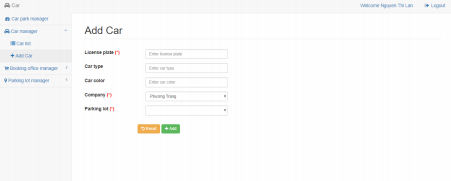
*Screen of car list*

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Car list | | |
| **Description** | Display Car list | | |
| **Screen Access** | Admin clicks **Car Management -> Car parking in lot -> List** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Search Box |  |  |
| Search criteria | ComboBox |  |  |
| Search | Button |  |  |
| License plate | Label - String (20) |  |  |
| Car type | Label - String(50) |  |  |
| Car color | Label - String(20) |  |  |
| Company | Label - String(50) |  |  |
| Parking lot | Label - String(50) |  |  |
|  | Label – String(20) |  |  |
| Prev | Link |  | When user clicks this link, then display car list at the previous page |
| 1 | Link |  | When the user clicks this link, then display car list at page 1 |
| 2 | Link |  | When the user clicks this link, then display car list at page 2 |
| 3 | Link |  | When the user clicks this link, then display car list at page 3 |
| Next | Link |  | When user clicks this link, then display car |

18

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | list at the next page | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Search | When the user clicks **Search,** the system will search car according to inputted data at search box. | | | If the user does not input any data at Search box, the system will  automatically  display all car on ticket list table. Otherwise,  display the search result on car list table. | | When database connection has error -> Display message  “Database  connection error” When the system does not find out any car, then display message “No matches” |

***3.1.4.2. Add car***

******

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Add car | | |
| **Description** | Allow Staff to add car information | | |
| **Screen Access** | Click Car management -> click “Car parking in lot” -> Click “Add” | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |

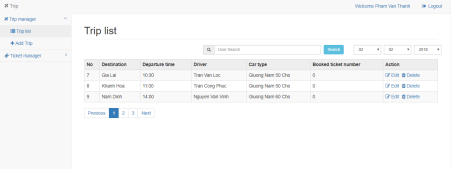
19

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| License plate | | Text field – String (50) |  | | |  |
| Car type | | Text field – String (50) |  | | |  |
| Car color | | Text field – String (50) |  | | |  |
| Company | | ComboBox |  | | |  |
| Parking lot | | ComboBox |  | | | Box to select parking lot (Only display parking lots with status “Emply” |
| Reset | | Button |  | | | The user clicks Reset to reset all data into default |
| Add | | Button |  | | | The user clicks Add to add information of cars and update into database, |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | **Failure** | |
| Reset | If user clicks Reset, then system deletes inputted information. | | | Delete all inputted  information. |  | |
| Add | When the user clicks **Add**, the system will validate data. If data is valid, the system will update new data into database,  otherwise display error message. | | | Back to detail screen and display  message “Add car  successfully” | When License plate box is empty -> The system will display message ‘’Please input License plate » | |

***3.1.5. Trip management***

***3.1.5.1. Trip list***

20



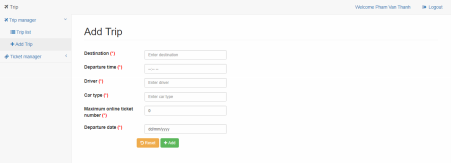
|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Trip list of company | | |
| **Description** | Display trip list of company | | |
| **Screen Access** | Admin clicks **Trip management -> List** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Search Box |  |  |
| Search | Button |  |  |
| No. | Label - String (20) |  |  |
| Destination | Label - String(50) |  |  |
| Departure time | Label - String(20) |  |  |
| Driver | Label - String(50) |  |  |
| Car type | Label - String(50) |  |  |
| Booked ticket number | Label – String(20) |  |  |
| Prev | Link |  | When user clicks this link, then display trip list at the previous page |

21

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | | Link |  | | When the user clicks this link, then display trip list at page 1 | |
| 2 | | Link |  | | When the user clicks this link, then display trip list at page 2 | |
| 3 | | Link |  | | When the user clicks this link, then display trip list at page 3 | |
| Next | | Link |  | | When user clicks this link, then display trip list at the next page | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Paging | When the user clicks any page at Paging, the system will query trip list at that page. Then the system will display trip list of the clicked page. | | | Display screen “Trip list” | |  |

22

***3.1.5.2. Add trip***

******

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Add information of trip | | |
| **Description** | Allows company to add trip information | | |
| **Screen Access** | Click Trip management -> Click “Add” | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Destination | Text field – String (50) |  |  |
| Departure time | Text field – String (50) |  |  |
| Driver | Text field – String (50) |  |  |
| Car type | Text field – String (50) |  |  |
| Maximum online ticket number | Text field – String (50) |  |  |
| Departure date | Date  Picker |  |  |
| Reset | Button |  | The user clicks Reset to reset data into default |

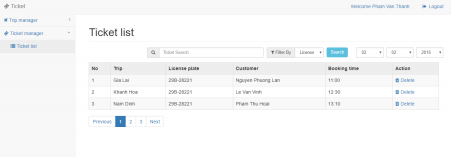
23

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Add | | Button |  | | | The user clicks Add to add information of trip into database |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | **Failure** | |
| Add | When the user clicks **Add**, the system will validate data. If data is valid, the system will save new data into database,  otherwise display error message. | | | Back to detail screen and display  message “Add trip  successfully” | When Maximum online ticket number is empty - > The system will display message ‘’Please select online ticket number’’  When inputting data over minimum, maximum of online ticket number -> The system will display message “Please input data between min and max”  When database connection has error -> Display message “Database connection error” | |

***3.1.6. Ticket management***

***3.1.6.1. Ticket list***

24



|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Ticket list | | |
| **Description** | Display Ticket list | | |
| **Screen Access** | The admin clicks **Ticket management -> List -> List** | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Search Box |  | Box to input search data |
| Search criteria | ComboBox |  | Box to select search criteria |
| Search | Button |  |  |
| No. | Label - Integer |  |  |
| Trip | Label - String (50) |  |  |
| License plate | Label - String (50) |  |  |
| Customer | Label - String(50) |  |  |
| Booking time | Label - Time |  |  |
| View | Link |  | When the user clicks **View** at any row, then display detail information page of ticket at that row |
| Prev | Link |  | When user clicks this link, then display ticket list at the |

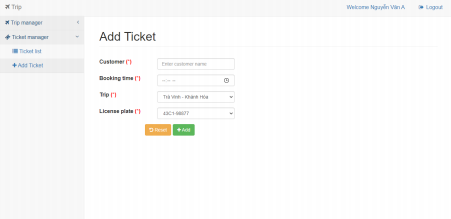
25

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | previous page | |
| 1 | | Link |  | | When the user clicks this link, then display ticket list at page 1 | |
| 2 | | Link |  | | When the user clicks this link, then display ticket list at page 2 | |
| 3 | | Link |  | | When the user clicks this link, then display ticket list at page 3 | |
| Next | | Link |  | | When user clicks this link, then display ticket list at the next page | |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | | **Failure** |
| Search | When the user clicks **Search,** the system will search ticket according to inputted data at search box. | | | If the user does not input any data at Search box, the system will  automatically  display all tickets on ticket list table. Otherwise,  display the search result on ticket list table. | | When database connection has error -> Display message  “Database  connection error”  When the system does not find out any ticket, then display message “No matches” |
| View | When the user clicks **View** at any row, the system will query information of the ticket at that row. Then the system will forward to screen View detail of ticket. | | | Display screen “View detail of ticket.” | | When database connection has error -> Display message  “Database  connection error” |

26

|  |  |  |  |
| --- | --- | --- | --- |
| Paging | When the user clicks any page at Paging, the system will query 10 records of tickets at that page. Then the system will display ticket list of the clicked page. | Display screen “Ticket  management” |  |

***3.1.6.2. Add ticket***

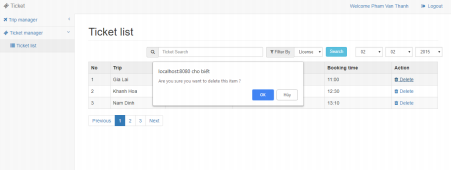
******

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Add information of ticket | | |
| **Description** | Allows company to add ticket information | | |
| **Screen Access** | Click Ticket management -> Click “Add Ticket” | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Customer | Text field – String (50) |  |  |
| Booking time | Time  Picker |  |  |
| Trip | Combo  box |  |  |
| License plate | Combo  box |  |  |

27

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Reset | | Button |  | | | The user clicks Reset to reset data into default |
| Add | | Button |  | | | The user clicks Add to add information of ticket into database |
| **Screen Actions** | | | | | | |
| **Action Name** | **Description** | | | **Success** | **Failure** | |
| Add | When the user clicks **Add**, the system will validate data. If data is valid, the system will save new data into database,  otherwise display error message. | | | Back to detail screen and display  message “Add ticket  successfully” | When database connection has error -> Display message “Database connection error” | |

***3.1.6.3. Delete ticket***

******

|  |  |
| --- | --- |
| **Screen** | Delete ticket |
| **Description** | Allows company to delete ticket |
| **Screen**  **Access** | Click Ticket management -> Click “List” -> Click “Delete” |

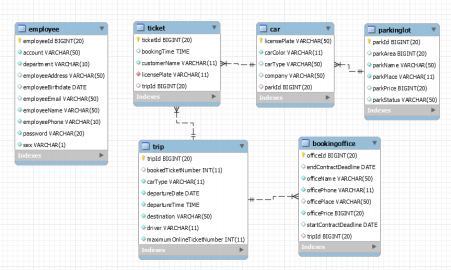
28

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen Content** | | | | | |
| **Item** | **Type** | **Data** | | **Description** | |
| Title | Label  String  (20) |  | | Display title of popup | |
| Content | Label – String  (50) |  | | Display content of popup, warning users to confirm before deleting. | |
| Yes | Button |  | | Click Yes if deleting ticket | |
| No | Button |  | | Click No if not deleting ticket | |
| **Screen Actions** | | | | | |
| **Action**  **Name** | **Description** | | **Success** | | **Failure** |
| No | If the user does not want to delete the ticket, then click No | | Close Popup, display the current page. | | When database connection has error -> Display message “Database connection error”  The screen still displays the current page |
| Yes | If the user confirms to delete the ticket, then click Yes | | The screen displays  message  “Delete  successfully”  and back to Ticket list screen. | | When database connection has error -> Display message “Database connection error”  The screen still displays the current page |

***3.2 Detailed Non-Functional Requirements***

***3.2.1 Logical Structure of the Data***

29

The logical structure of the data to be stored on the system is given below: 

***3.2.2 Security***

TBD

***3.2.3 Performance***

TBD

***3.2.4 Technial Requirement***

Framwork : Spring framwork version 4.x

IDE, tools : Eclipse, Maven 3.x

Server tool : Tomcat 8.x

Java version : Java 1.8 or late

Data base : Oracle 11g

30