

EASY TICKETS

2011

Hung Vuong Aptech Education

07/12/2011

**Supervisor:** Nguyen Xuan Doan

**Curriculum:** ACCP i7.1

**Semester :** 4 - HDSE

**Batch Code:** 0906E

**Student :** Pham Hong Tam

Tran Phan Quoc Hai

Nguyen Huu Sang

Nguyen Duc Viet



This is to certify that

**Mr. Pham Hong Tam**

has successfully designed and

developed

**EasyTickets**

Submitted by:

**PHAM HONG TAM**

Date of Issue: 07/12/2011

Authorized Signature: \_\_\_\_\_\_\_\_\_\_\_



This is to certify that

**Mr. Tran Phan Quoc Hai**

has successfully designed and

developed

**EasyTickets**

Submitted by:

**TRAN PHAN QUOC HAI**

Date of Issue: 07/12/2011

Authorized Signature: \_\_\_\_\_\_\_\_\_\_\_



This is to certify that

**Mr. Nguyen Huu Sang**

has successfully designed and

developed

**EasyTickets**

Submitted by:

**NGUYEN HUU SANG**

Date of Issue: 07/12/2011

Authorized Signature: \_\_\_\_\_\_\_\_\_\_\_



This is to certify that

**Mr. Nguyen Duc Viet**

has successfully designed and

developed

**EasyTickets**

Submitted by:

**NGUYEN DUC VIET**

Date of Issue: 07/12/2011

Authorized Signature: \_\_\_\_\_\_\_\_\_\_\_

**1. PROJECT DEFINITION:**

[**TABLE OF CONTENT**]

**Content Page**

1. Introduction 6
2. Problem Definition 7
3. Customer Requirement Specification (CRS)
4. Project Plan
5. System Architecture
6. Nonfunctional requirement specification
7. Use Case
   1. Customer
   2. Administrator
8. Flowchart
9. Sequence Diagram
10. Entity Relation Ship Diagram (ERD)
11. Data Flow Diagram (DFD)
12. Class Diagram
13. Table Design
14. GUI Standard Document
15. Site Map
16. Interface Design Program
17. Task Sheet
18. Unit Testing Check List
19. Final Check



**INTRODUCTION**

In today’s busy world, people find it hard to find the time for entertainment. It is really pretty time consuming to find out the various entertainment shows/movies and any other such shows in and around the city. And on the top of it is more irritating to go physically and book the ticket.

E-commerce can be thought as the best solution for this. Due to which the people can buy the tickets right from home. No more buying in black. Moreover one can find the information just with a click and get the booking done.

**Showbiz** is the company is principally into ticket sales, marketing and distribution of event tickets. The services include providing information on multiple event categories, providing ticketing facilities for several leading multiplexes, arenas and stadiums. They are thinking of the concept of developing a web portal, where a customer can book movie tickets in theatres that are made available in the site. They wish to launch the website by the name [**www.easytickets.com**](http://www.easytickets.com) where they can provide convenient, secure, and fair access to the best possible tickets offered by the clients. Showbiz aims at making events happening across the city more accessible-just a click away. Booking tickets can now an effortless, fun and exciting experience for every showbiz customer.



1. Overview:

* Building a web portal, where a customer can book movie tickets in theatres that are made available in the site.
* The website by the name [**www.easytickets.com**](http://www.easytickets.com) where it can provide convenient, secure, and fair access to the best possible tickets offered by the clients.
* In addition there are other kinds of tickets as Sport Event, Music, Drama.

1. People shall use the system:

* Customer, who visits the site and books the ticket.
* Administrator, who manages the site.

1. Book the ticket:

* Check the availability of tickets on mentioned date and time in specified city.
* View detail of ticket (venue, day/time, artist…).
* Book the tickets.
* The user selects a particular movie and show in a selected city, if there is availability of tickets for that show, then confirmation for ticket booking must be done. In case of non availability of tickets, the portal must display next available date & time.
* There is no cancellation of tickets once they are booked, at any cost. The provision for money transaction is provided in the same portal. This requires possession of a bank‘s credit or debit card with the customer.
* Online booking can be done only 15 days in advance.

1. Difference:

* Customer can *contact* with Company about problems.
* Website can provide information of company(Aboutus) and FAQ.

**2. CUSTOMER REQUIREMENT SPECIFICATION/OVERVIEW**

**Name of the client:** Showbiz Company

**System Study:**

* 1. ***List of inputs to the system***
* Information of customer.
* Contact information of customer.
* The ticket information was booked by customer.
  1. ***List of outputs expected from the system***
* Ticket information.
* List of booking.
* FAQ information.
* Email of administrator.

***2.3. Overview of processes involved in the system***

1. Customer will register account, and login website
2. After login, Customer book ticket, system will check it. If tickets are confirmed, system will store and notify customer.
3. In case of non availability of tickets, must display next available date/time.
4. Customer contact with company, website will send mail from administrator for customer
5. Customer view FAQ, about us.
   1. ***Data Constraints/Triggers/Validation tables.***
6. Account Information:

* Ebtablished date must to be greater than birthday of customer.
* Phone must to be integer number
* Person Card Number must to be integer number

1. Ticket Information:

* Discount is less than price
* Ticket total must be greater than or equal to 0.
* Price is greater than 0
* Ebtablished data must to be less than or equal to view date.

1. Booking

* Discount < Price Total
* If user books the ticket, program will update ticket number
* Delivery status is equal to Yes or No (Integer value is 1 or 0)
* Ticket total must be greater than 0
* Price total must be greater than 0
* Online booking can be done only 15 days in advance.
  1. ***Hardware and software required for implementing the project***

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| * A minimum computer system that will help you access all the tools in the courses is a Pentium 166 or better * 128 Megabytes of RAM or better | * Java Virtual Machine/ J2EE server. * Notepad/Java editor (Netbeen 6.8). * j2sdk1.4.1\_02 (or later). * EJB Dev Kit . * Java enabled web server. * JSP / Servlets Dev. Kit . * Subversion. |

**3. PROJECT PLAN**

***3.1. Project Details***

* Customer: **Showbiz Company**
* Date of the Project Plan: 07/09/2011
* The Project Purpose:
* The purpose of the project is to develop an intervention program using information technology as a means of providing information about ticket booking with a program offering home visits. The evaluation of the project will yield important for developing of Showbiz Company**.**
* The people can buy the tickets right from home. No more buying in black. Moreover one can find the information just with a click and get the booking done.
* The Project Scope:
* Define problem.
* Define deliverables.
* Define functionality and data.
* Define technical structure.
* Design and coding on module.
* Test and Review.
* Our Understanding Of The Client Organization:

**Showbiz**‘ is the company is principally into ticket sales, marketing and distribution of event tickets.

* Team Member:

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibility** |
| Pham Hong Tam | Leader | Manage group. |
| Nguyen Huu Sang | Developer | Coding and design document. |
| Tran Phan Quoc Hai | Developer | Coding |
| Nguyen Duc Viet | Tester | Test |

***3.2. Documents from Client***

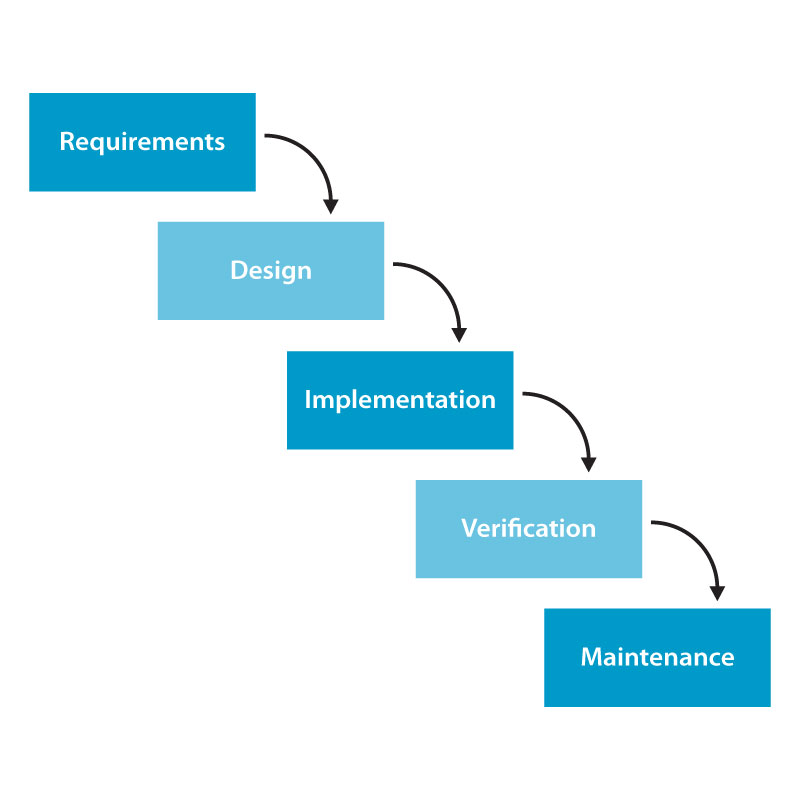
* Requirement (EasyTickets.doc file)
* Feadback form (eProject\_Feeback\_Form.exl file)
* Project status (eProject\_Status\_Report(V1).exl file)
* Project specification (ProjectSpecification.doc file)

***3.3. Deliverables***

* Installation Manual
* User Manual
* Document
* Application

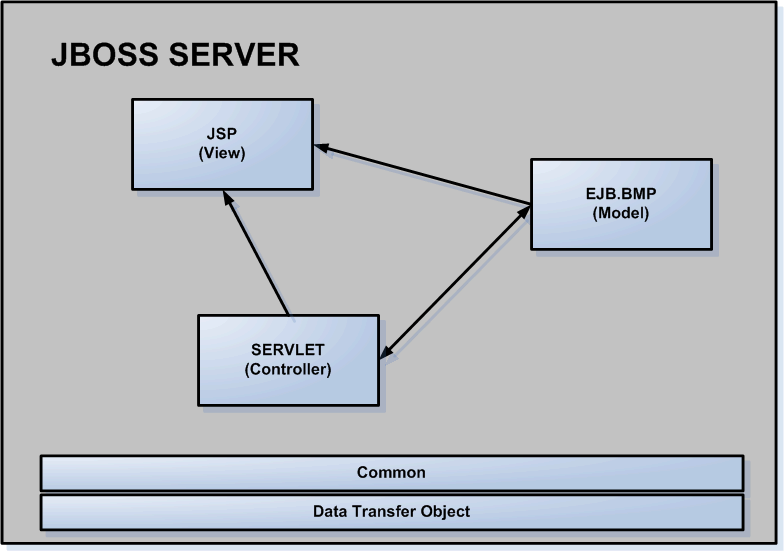
***3.4. Project Dependences***

* Support from teacher.
* Good working environment.
* Members with good capacity.

***3.5. Phase Of The Project***

***3.6. Quality Plan***

* Review Activities (review meeting participants, and frequency).
* Testing Activities (Unit Test Plans, Integration Testing and Functionality Testing).
* Group will do something such as test links, test function every meeting.
* Backup and recovery strategies (in case of disk crash and network failures).
* After weekly, group will backup source code, database.

**4. SYSTEM ARCHITECTURE:**

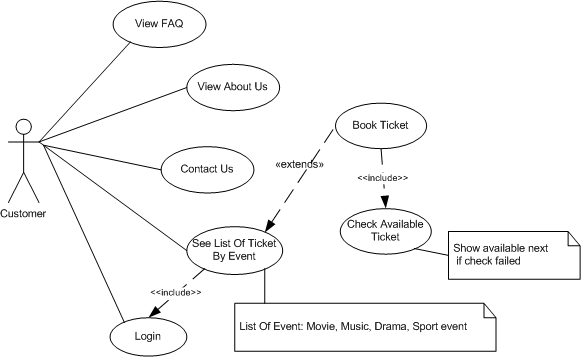
Benefits:

* Easier to build, maintain and extend
* Provide single point of control (Servlet) for security & logging
* Encasulate incoming data into a form usable by the backend
* Can reusable code

**5. NON FUNCTIONAL REQUIREMENT SPECIFICATION:**

* Evolution: Website uses some techniques such as JSP & Servlet, EJB... and it is always ready for upgrade, expanse the function and it also allow the administrator change the parameter and the configuration of the system easily.
* Friendliness: The website’ layout is clearly, friendly and easy to use with many kinds of user: user has the knowledge about information technology as well as user doesn’t have the knowledge about information technology.
* Safeness:
* The information in the system is only edited by authorized users.
* The personal information such as profile, advertisement and the property information is only edited by the member who is the owner of that information. Only the administrator or the authorized users have the suitable responsibility and authority can control, manage and update the information of system.

**6.USECASE**

******

***6.1. Customer***

***6.1.1.View FAQ***

|  |  |
| --- | --- |
| Use Case Name | View FAQ |
| Actors | Customer |
| Description | Actor want to see common question and answers. |
| Pre-conditions | N/A |
| Post-conditions | Success: Show FAQ information.  Fail: Not data. |
| Basic Flow | 1. Actor goes to Home page of website, and clicks the FAQ image.  2. System redirects to FAQ page and shows information of FAQ.  3. This page has a View link, actor clicks View link to see detail of FAQ. |
| Alternative Flow | N/A |
| Exception | N/A |

***6.1.2.View About Us***

|  |  |
| --- | --- |
| Use Case Name | View About Us |
| Actors | Customer |
| Description | Actor can see company information. |
| Pre-conditions | N/A |
| Post-conditions | N/A |
| Basic Flow | N/A |
| Alternative Flow | N/A |
| Exception | N/A |

***6.1.3. Contact Us***

|  |  |  |
| --- | --- | --- |
| Use Case Name | Contact Us | |
| Actors | Customer | |
| Description | Actor can contact with administrator. | |
| Pre-conditions | N/A | |
| Post-conditions | Success: Actor waits mail from administrator.  Fail: Contact again. | |
| Basic Flow | 1. Actor goes to Home page of  website, then click Contact image.  2. System redirects to Contact page that has form with some control as:  - ‘Title’ text field  - ‘Content’ text field  - ‘Email’ text field  3. Actor enters title, content and  email, then click the Submit  button.  4. System checks the information.  5. System saves data into database. | |
| Alternative Flow | Actor clicks Reset button to reset the value of all fields to defaut value. | |
| Exception | Actor Actions:  Actor doesn’t enter sufficient information. | System Responses:  - System redirects to contact page again.  -System shows a message:  “Title can’t empty”, “Content can’t empty”, “Email can’t empty”. |

***6.1.4. Login***

|  |  |  |
| --- | --- | --- |
| Use Case Name | Login | |
| Actors | Customer | |
| Description | Actor logins to view ticket information and booking. | |
| Pre-conditions | Actor must register member. | |
| Post-conditions | Success: Actor logged in to website.  Fail: Login again. | |
| Basic Flow | Actor Actions:  1. Actor goes to Home page of website, then click Login link.  2. System redirects to Login page that has form with some control as:  - ‘Username’ text field.  - ‘Password’ text field.  - ‘Login’ button.  - ‘Cancel button.  3. Actor enters Username and Password, then click the Submit button.  4. System checks the information.  5. System redirects to Home page. | |
| Alternative Flow | Actor Actions:  1. Actor clicks Register link.  2. If login is successful, system redirects Home page if role of account is customer.  3. If role of account is admin, system will redirects to Admin page | |
| Exception | Actor Actions:  1. Actor enters invalid username and password. | System Responses:  - System redirects to Login page  again.  - System shows a message: “Invalid Account”. |

***6.1.5. See List Of Ticket By Event***

|  |  |  |
| --- | --- | --- |
| Use Case Name | See List Of Ticket By Event. | |
| Actors | Customer | |
| Description | Actor want to see a list of tickets by event. | |
| Pre-conditions | Actor must login to see ticket information. | |
| Post-conditions | Success: Actor can view ticket information and booking.  Fail: Actor registers and logins again. | |
| Basic Flow | Actor Actions:  1. Actor goes to Home page of  website, then click <<Name>> Event link. | System Responses:  2. System redirects to Show Ticket page that has form with control ‘Book’ button, More link, Close link (Close link is shown after user have clicked More link). |
| Alternative Flow | Actor clicks into More link to see detail of ticket.  Actor clicks into Close link to close detail of ticket.  Actor books ticket | |
| Exception | Actor doesn’t login who click Name of event Link, system warns user must login. | |

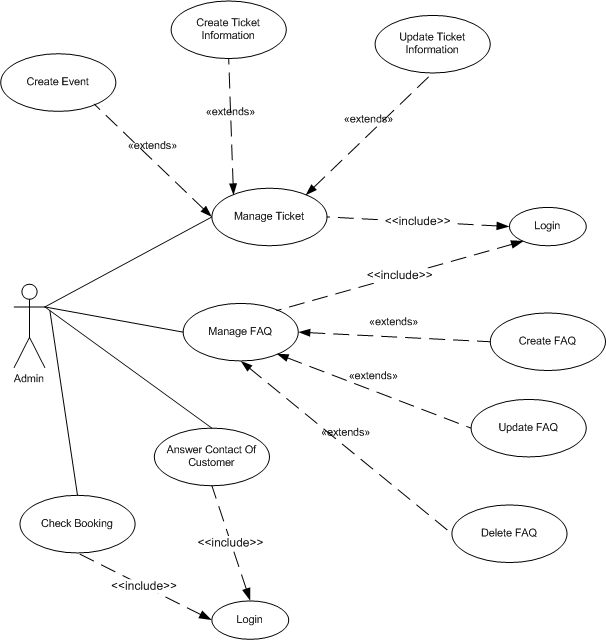
***6.1.6. Book Ticket***

|  |  |  |
| --- | --- | --- |
| Use Case Name | Book Ticket | |
| Actors | Customer | |
| Description | Actor want to book ticket. | |
| Pre-conditions | Check available ticket and customer must login. | |
| Post-conditions | Success: Actor can go to company to recevie ticket.  Fail: View next available ticket. | |
| Basic Flow | 1. Actor goes to Home page of  website, then click Name Of Event link on menu.  2. System redirects to Book Ticket that has form with some control as:  -‘Ticket Total’ text field.  -‘Payment Type’ combobox.  -‘Card Number’ text field.  - ‘Check price’ button.  - ‘Submit‘ button.  3. Actor enters information on form then click the Submit  button.  4. System checks the information.  5. System redirects to show ticket page. | |
| Alternative Flow | 1.If booking finishs, Actor can click “Back to Ticket page” link, system redirects to ticket page.  2. Actor click Check price button to show price and discount who must pay. | |
| Exception | Actor Actions:  1. Actor inputs invalid amount.  2. Not available ticket. | System Responses:  - System redirects to booking page again.  - System shows a message: “Not available ticket”, “must be number”. |

***6.1.7. Check Available Ticket***

|  |  |
| --- | --- |
| Use Case Name | Check Available Ticket |
| Actors | Customer |
| Description | Check available ticket |
| Pre-conditions | N/A |
| Post-conditions | Success: Customer can booking.  Fail: Show next available ticket. |
| Basic Flow | 1. Actor goes to Home page of  website, then clicks <<Name>> Event link, clicks Book button.  2. System checks available ticket,  3. System shows next available ticket or redirects to booking page. |
| Alternative Flow | If ticket isn’t available, system will redirect Show ticket page again. |
| Exception | N/A |

***6.2. Admin***



***6.2.1. Check Booking***

|  |  |
| --- | --- |
| Use Case Name | Check Booking |
| Actors | Admin |
| Description | Actor updates status of booking. |
| Pre-Conditions | Actor has logged in with admin role. |
| Post-conditions | Status of booking was updated |
| Basic Flow | 1. Actor goes to Home page of  website, then click login link.  2. System redirect to admin page. This page has a Booking table with a some title as:  +Ticket.  + Booking.  + FAQ.  + Contact.  3. After login, click into Booking  4. System shows booking information and a some control as:  - ‘Delivery Status’ combobox.  - ‘Item’ checkbox.  - ‘Submit’ button.  5. Actor chooses ‘New’ status on Delivery Status combobox.  6. System shows booking which user send to.  7. Actor checks items and clicks Submit button.  6. System updates database and response about this current page. |
| Alternative Flow | Actor can change Delivery Status combobox is ‘Finish’ to see ticket which is accepted by actor. |
| Exception | N/A |

***6.2.2. Manage Ticket***

***6.2.2.1. Create Ticket Information***

|  |  |
| --- | --- |
| Use Case Name | Create Ticket Information |
| Actors | Admin |
| Description | Actor wants to create a new ticket. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Actor creates a new ticket in this web site.  Fail: Refill information. |
| Basic Flow | 1. Actor goes to Home page of website, then click login link.  2. Actor logins system with admin role.  3. After login, system redirects to admin page.  4. This page has a table with Booking, Ticket, FAQ, Contact title.  5. Actor clicks into Ticket tite of table with control as: ‘Type’ Combobox. Actor chooses Event type.  6. System shows list of event, On each row of event table has Create Ticket link, actor clicks into it. System redirect to new form.  This new form has a some controls as:  - ‘Submit’ button.  - ‘Ticket Total’ button.  - ‘Price’ button.  - ‘Discount’ text field.  - ‘Promotion’ text area.  - ‘View Date’ text field.  - ‘Hour’ combobox.  - ‘Minute’ combobox  - ‘Reset’ button  7. Actor enters information above.  8. System check information as ticket total, view date, time, discount.  9. System saves in database. |
| Alternative Flow | 1. Ticket Total must be number. 2. View Date, time must be number. 3. View Date is greater than current date. 4. Actor clicks View Date texfield, system shows table of date to actor can choose correct date. 5. Actor clicks Reset button to reset default value of form. |
| Exception | 1. Actor enters incorrect date.  2. Actor enters Ticket Total text field doesn’t number, system will warns. |

***6.2.2.2. Update Ticket Information***

|  |  |
| --- | --- |
| Use Case Name | Update Ticket Information |
| Actors | Admin |
| Description | Actor updates the ticket about price, ticket number, discount, date, promotion. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Ticket information was updated.  Fail: Refill information. |
| Basic Flow | 1. Actor goes to Home page of website, then clicks login link.  2. Actor logins system with admin role.  3. After login, system redirects to admin page. This form has a table with Ticket, Booking, Contact, FAQ title.  4. Actor clicks into Ticket tite of table with a ‘Type’ Combobox, then actor chooses Ticket type.  5. System shows list of ticket and choose a item by clicks Update link. System redirects to new form with a some controls as:  - ‘Submit’ button.  - ‘Ticket Total’ button.  - ‘Price’ button.  - ‘Discount’ text field.  - ‘Promotion’ text area.  - ‘View Date’ text field.  - ‘Hour’ combobox.  - ‘Minute’ combobox  - ‘Reset’ button  6. Actor enters information above.  7. System check information as ticket total, view date, time, discount.  8. System updates database. |
| Alternative Flow | 1. Ticket Total must be number.  2. View Date, time must be number.  3. View Date is greater than current date.  4. Actor clicks View Date texfield, system shows table of date to actor can choose correct date.  5. Actor clicks Reset button to reset default value of form. |
| Exception | 1. Actor enters incorrect date.  2. Actor enters Ticket Total text field doesn’t number, system will warns. |

***6.2.2.3. Create Event***

|  |  |
| --- | --- |
| Use Case Name | Create Event |
| Actors | Admin |
| Description | Actor wants to create a event for ticket. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Admin can create ticket with this event.  Fail: Refill information. |
| Basic Flow | 1. Actor logins system with admin role. 2. After login, system redirects to Admin page has a some title as: Ticket, Booking, Contact, FAQ title on table. 3. Actor clicks Ticket title. 4. This form has a ‘Type’ combobox, actor changes combobox is Event type. Create Event button is shown, actor clicks it. System shows new form. 5. This new form has a some control as:   ‘Title’ field  ‘Content’ field  ‘Artist’ field  ‘Event Type’ combobox  ‘Venue’ combobox  ‘City’ combobox.  ‘Submit’ button  ‘Reset’ button  ‘Image’ field   1. Actor fills information and submit. 2. System checks information and save in database. |
| Alternative Flow | 1. Actor clicks Reset button to reset default value of form.  2. Actor chooses image with JPEG, JPG, PNG, GIF file. |
| Exception | 1. If Actor doesn’t fill information, system will warn. 2. If image doesn’t correct with format, system will warn. |

***6.2.3. Manage FAQ***

***6.2.3.1. Create FAQ***

|  |  |
| --- | --- |
| Use Case Name | Create FAQ |
| Actors | Admin |
| Description | Actor will answer or post questions available to. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Admin create a new FAQ in this web site  Fail: Refill information. |
| Basic Flow | Actor logins system with admin role.  System redirects to admin page with table contains titles as:  Ticket, Booking, FAQ, Contact title.  Actor chooses FAQ, system will show list of FAQ, and “Create FAQ” button, actor clicks it, system will show new form with some control as:  + ‘Title’ field  + ‘Content’ field  + ‘Submit’ button  + ‘Reset button’  Actor fills information.  System checks information and saves in database. |
| Alternative Flow | Actor clicks Reset button to reset default value of form. |
| Exception | If actor doesn’t fill title and content, system will warn. |

***6.2.3.2. Update FAQ***

|  |  |
| --- | --- |
| Use Case Name | Update FAQ |
| Actors | Admin |
| Description | Admin will update the questions and answers. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Admin update FAQ in this web site  Fail: Refill information. |
| Basic Flow | Actor logins system with admin role.  System redirects to admin page with table contains titles as:  Ticket, Booking, FAQ, Contact title.  Actor chooses FAQ, system will show list of FAQ, and “Create FAQ” button. On row of FAQ list, actor clicks Update link of FAQ item, system will show new form with some control as:  + ‘Title’ field  + ‘Content’ field  + ‘Submit’ button  + ‘Reset button’  Actor fills information.  System checks information and updates in database. |
| Alternative Flow | Actor clicks Reset button to reset default value of form. |
| Exception | If actor doesn’t fill title and content, system will warn. |

***6.2.3.3. Delete FAQ***

|  |  |
| --- | --- |
| Use Case Name | Delete FAQ |
| Actors | Admin |
| Description | Actor will delete the questions and answers. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Admin delete FAQ in this web site  Fail: Refill information. |
| Basic Flow | 1. Actor logins system with admin role.  2. System redirects to admin page with table contains titles as:  Ticket, Booking, FAQ, Contact title.  3. Actor chooses FAQ, system will show list of FAQ, and “Create FAQ” button. On each row of FAQ list, actor clicks Delete link of FAQ item, this form will show a “Do you sure want to delete ” message, actor click OK to accept.  4. System deletes FAQ. |
| Alternative Flow | 1. If system delete successful, system will remove on display.  2. If “Do you sure want to delete ” message is shown, actor clicks cancel to close this message. |
| Exception | N/A |

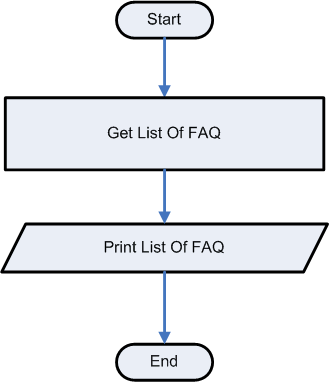
***6.2.4. Answer Contact Of Customer***

|  |  |
| --- | --- |
| Use Case Name | Answer Contact Of Customer. |
| Actors | Admin |
| Description | Actor will answer questions from customers. |
| Pre-conditions | Actor has logged in. |
| Post-conditions | Success: Email is sent.  Fail: Refill information |
| Basic Flow | 1. Actor logins system with admin role.  2. System redirects to admin page with table contains titles as:  Ticket, Booking, FAQ, Contact title.  3. Actor chooses Contact, system will show list of contact, and “Answer” link on row of contact list. System will show a new form with some control as: ‘Title’ text field, ‘Answer’ text field, ‘Send’ button, ‘Reset’ button.  4. Actor enters Answer field and clicks Sent button  4. System send mail to customer and save answer in database. |
| Alternative Flow | Actor click Reset button to reset default value on form |
| Exception | N/A |

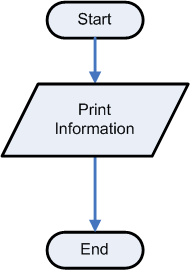
**7. FLOWCHART DIAGRAM**

***7.1. Customer***

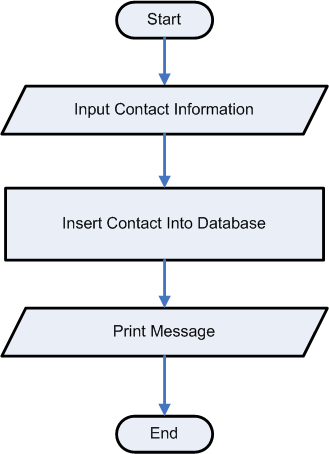
***7.1.1. View FAQ***

****

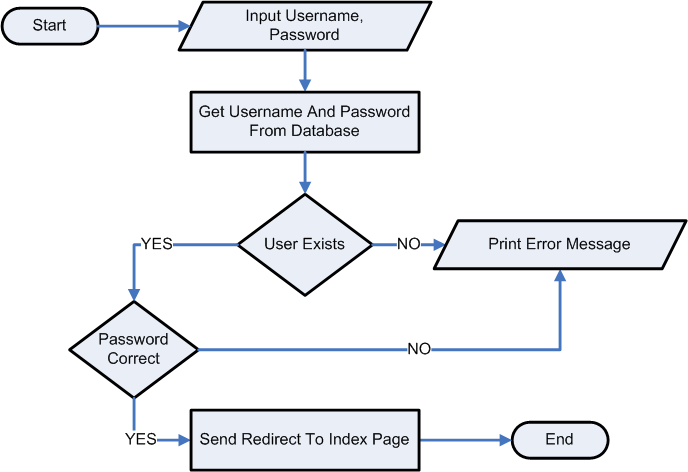
***7.1.2. View About Us***



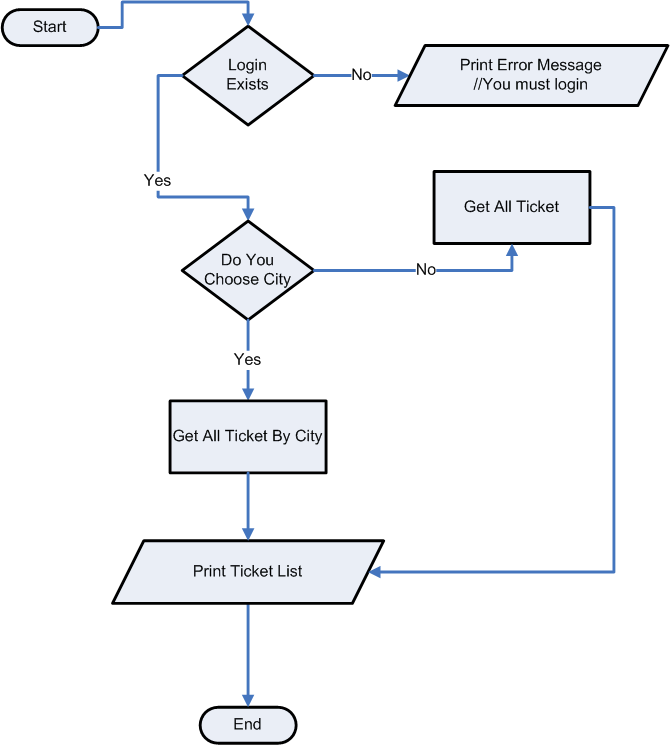
***7.1.3. Contact Us***

****

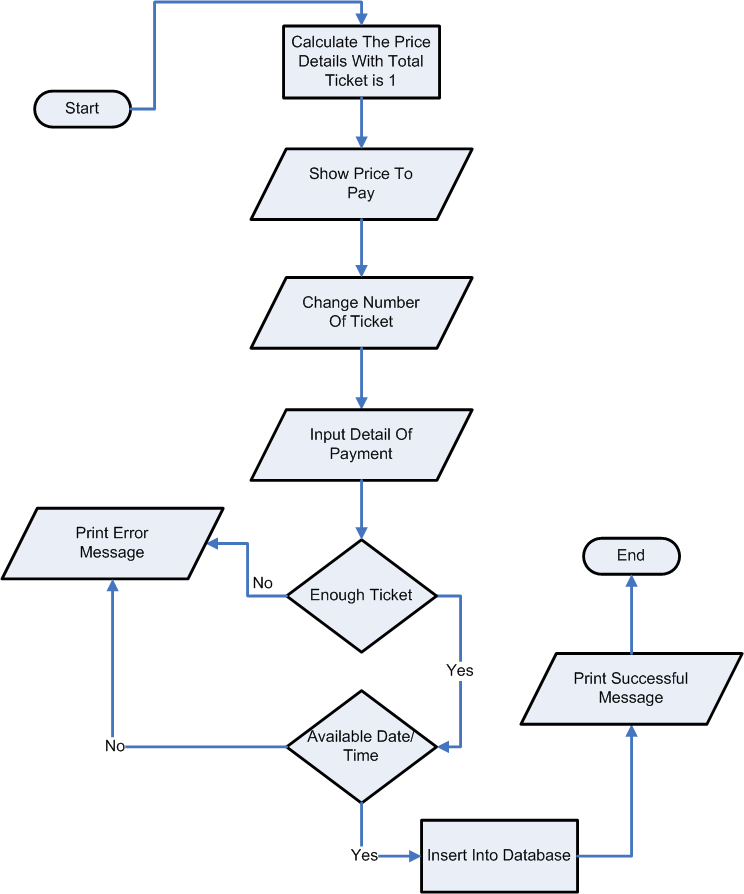
***7.1.4. Login***

******

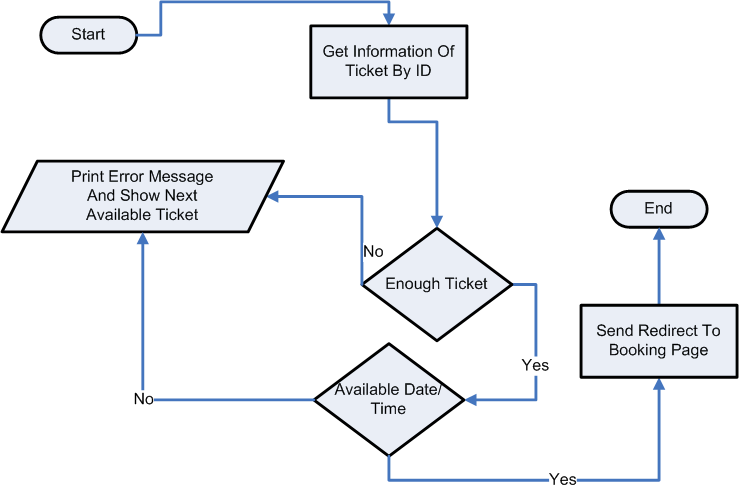
***7.1.5. See List Of Ticket By Event***

******

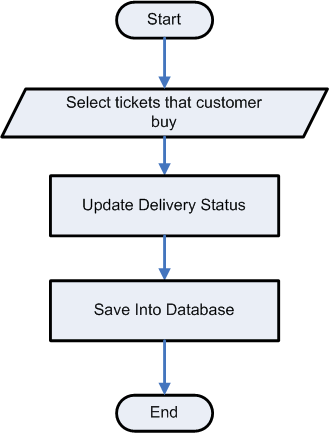
***7.1.6. Book Ticket***

******

***7.1.7. Check Available Ticket***

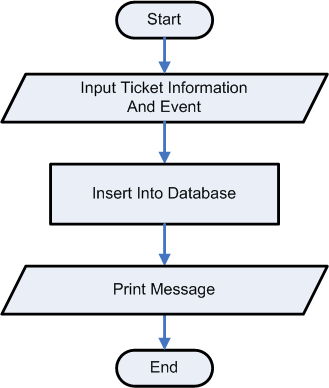
******

***7.2. Admin***

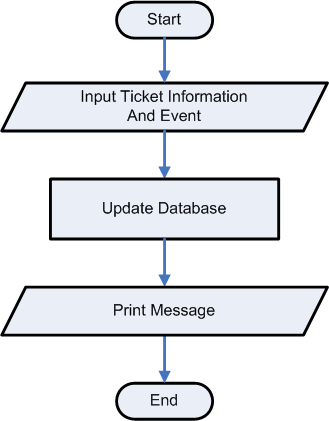
** *7.2.1. Check Booking***

***7.2.2. Manage Ticket***

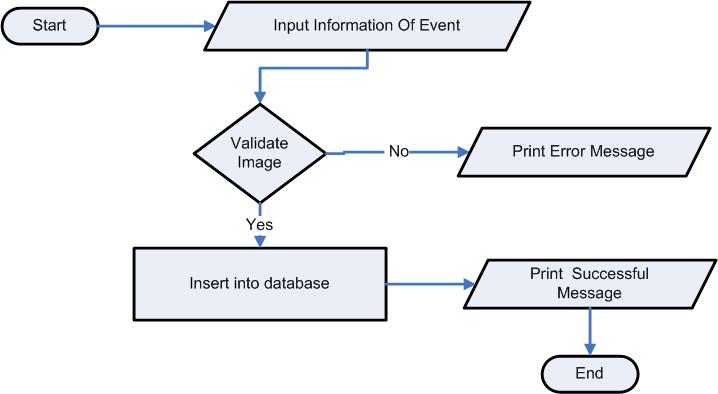
***7.2.2.1. Create Ticket Information***

******

***7.2.2.2. Update Ticket Information***

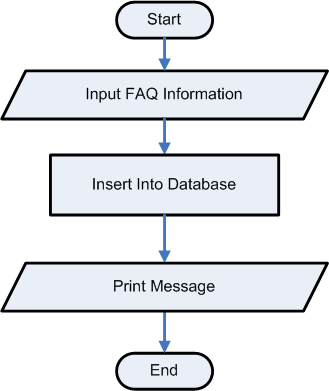
******

***7.2.2.3. Create Event***

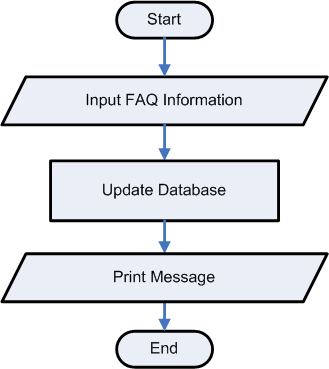
******

***7.2.3. Manage FAQ***

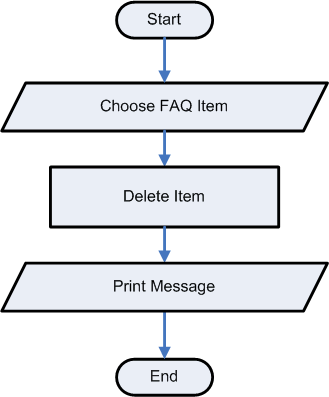
***7.2.3.1. Create FAQ Information***

****

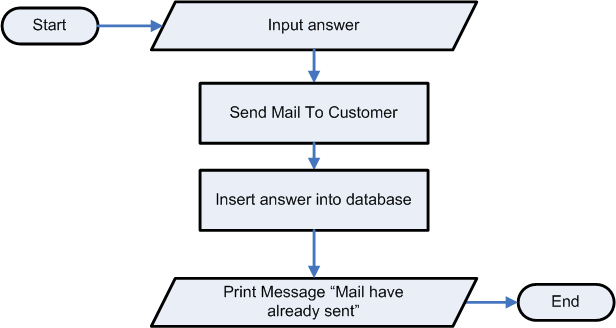
***7.2.3.2. Update FAQ Information***

******

***7.2.3.3. Delete FAQ Information***

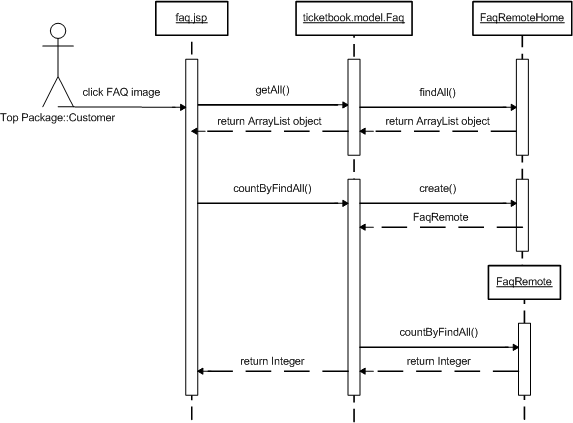
******

***7.2.4. Answer Contact Of Customer***

******

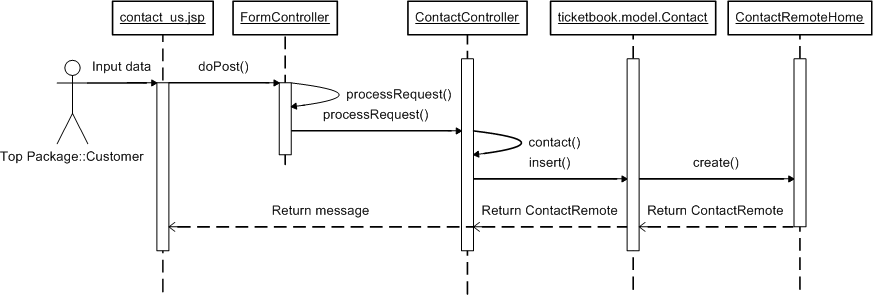
**8. SEQUENCE DIAGRAM**

***8.1 Customer***

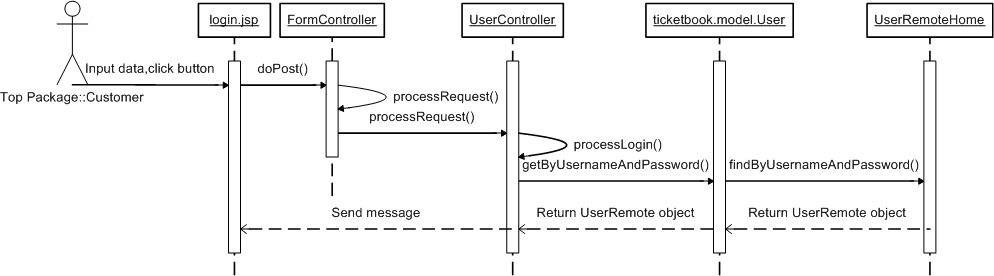
** *8.1.1. View FAQ***

***8.1.2. View About Us (Not Available)***

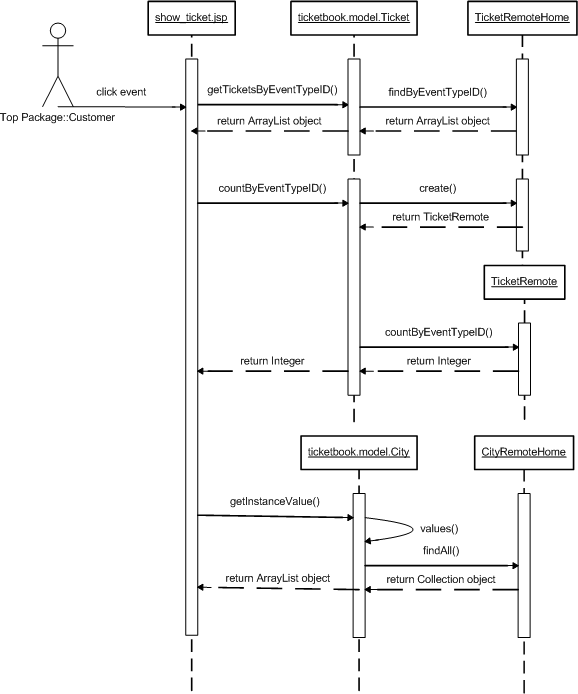
***8.1.3. Contact Us***

****

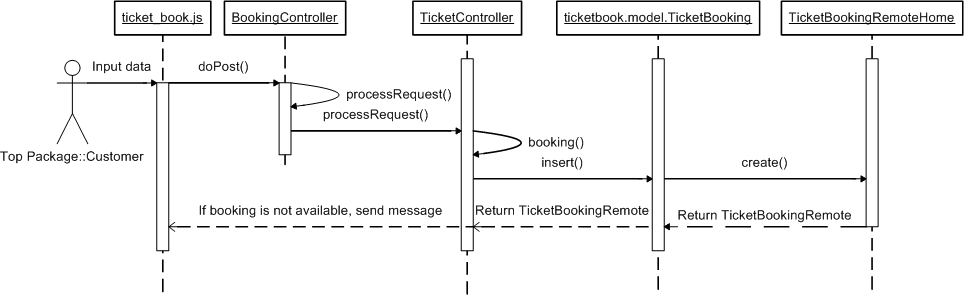
***8.1.4. Login***

******

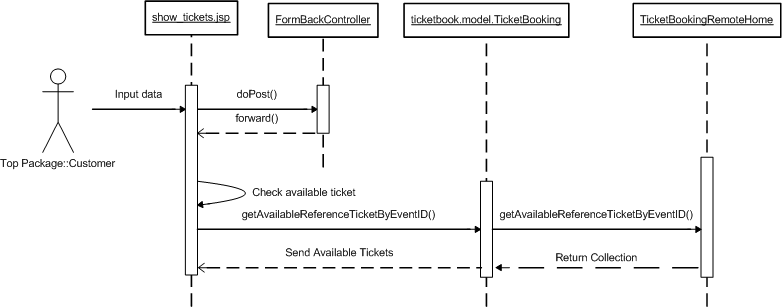
***8.1.5. See List Of Ticket By Event***

******

***8.1.6. Book Ticket***

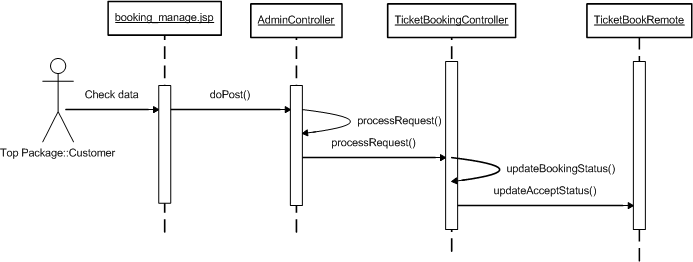
******

***8.1.7. Check Available Ticket***

******

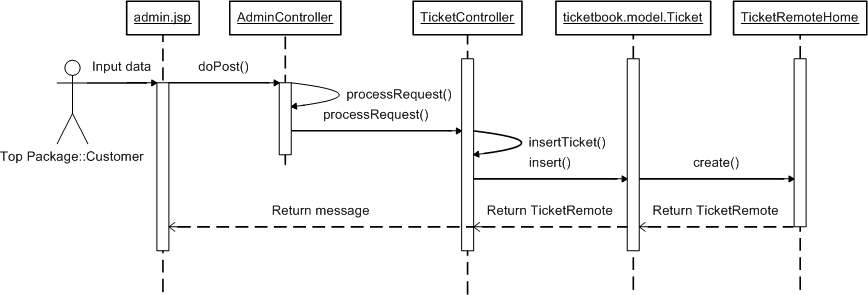
***8.2. Admin***

***8.2.1. Check Booking***

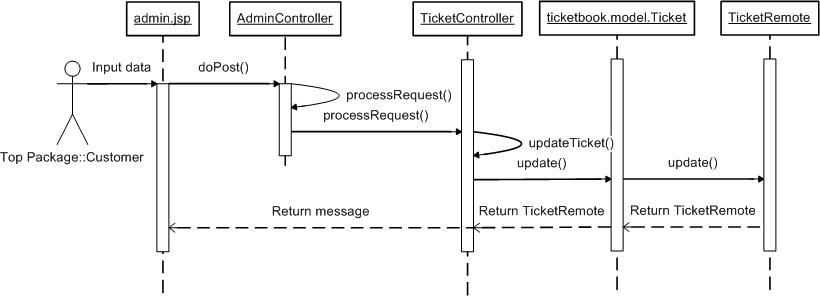
******

***8.2.2. Manage Ticket***

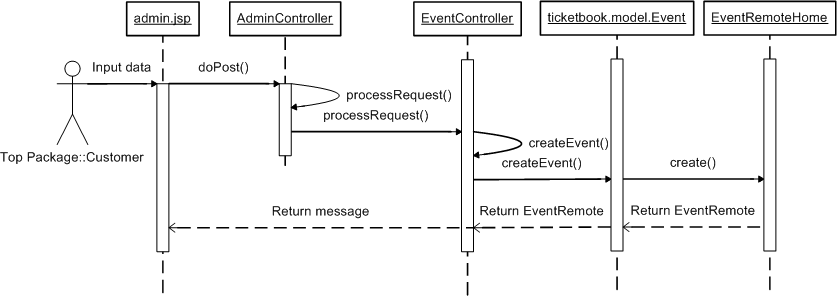
***8.2.2.1. Create Ticket Information***

******

***8.2.2.2. Update Ticket Information***

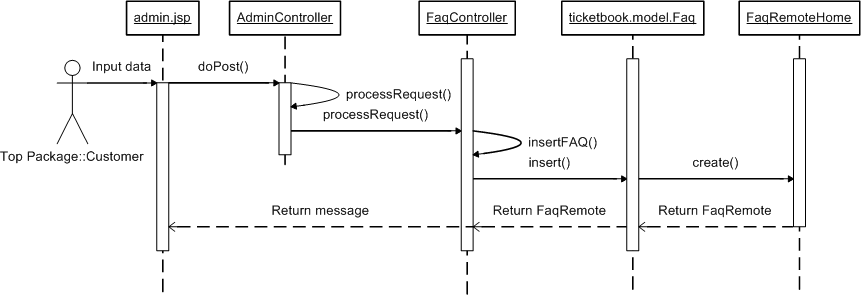
******

***8.2.2.3. Create Event***

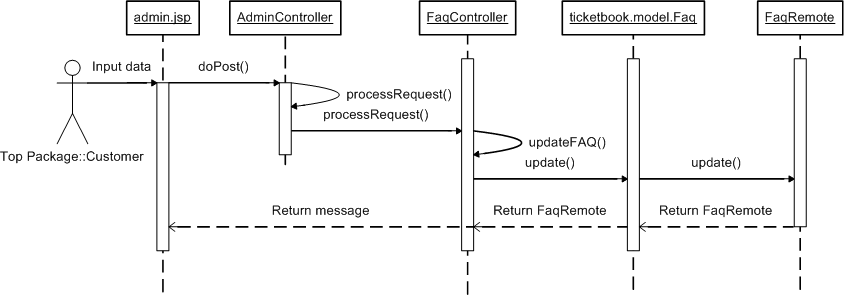
******

***8.2.3. Manage FAQ***

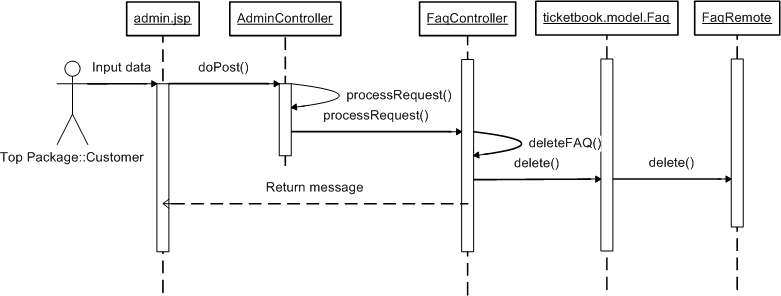
***8.2.3.1. Create FAQ Information***

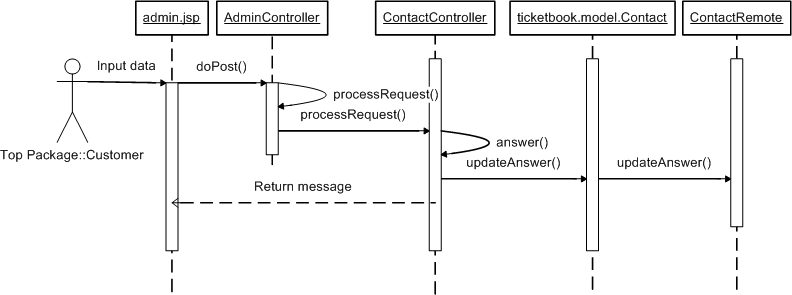
******

***8.2.3.2. Update FAQ Information***

******

***8.2.3.3. Delete Ticket Information***

******

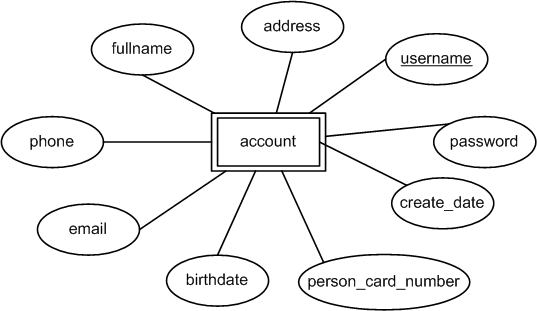
** *8.2.4. Answer Contact Of Customer***

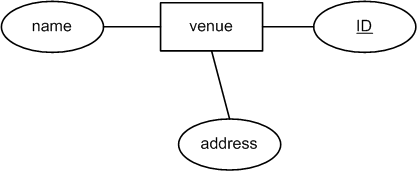
**9. E-R DIAGRAM**

***9.1. Entities***

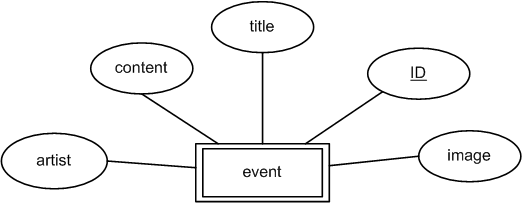
**role**

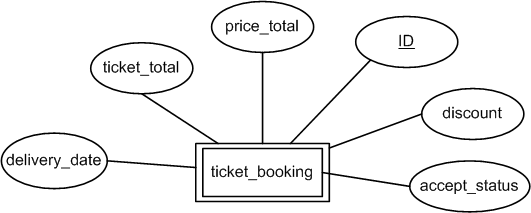
**city**

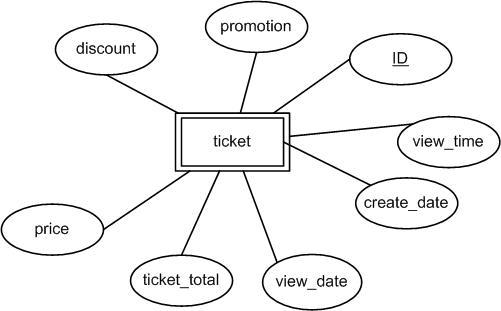
****

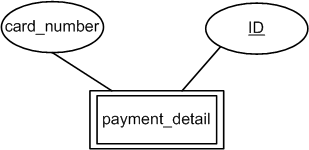
****

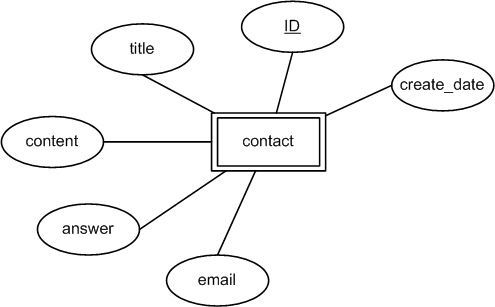
**event_type**

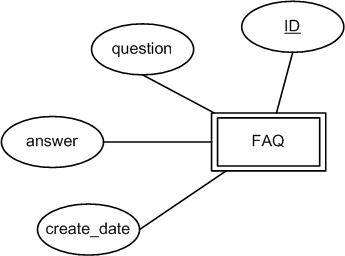
****

****

****

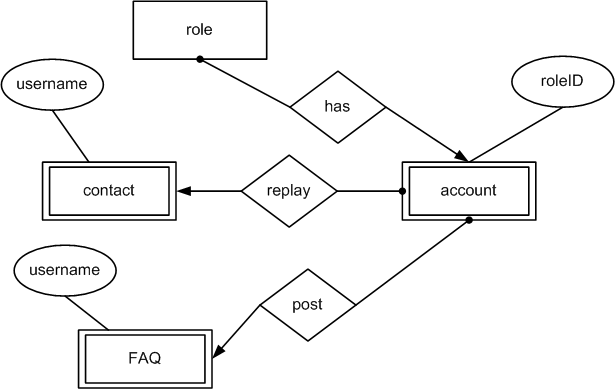
****

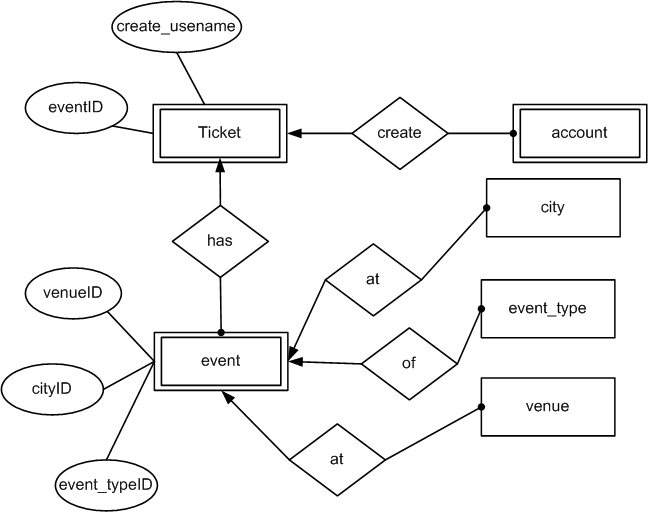
****

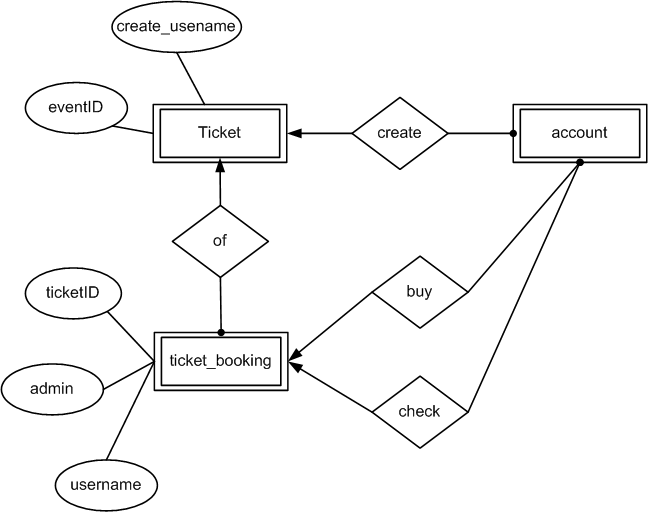
****

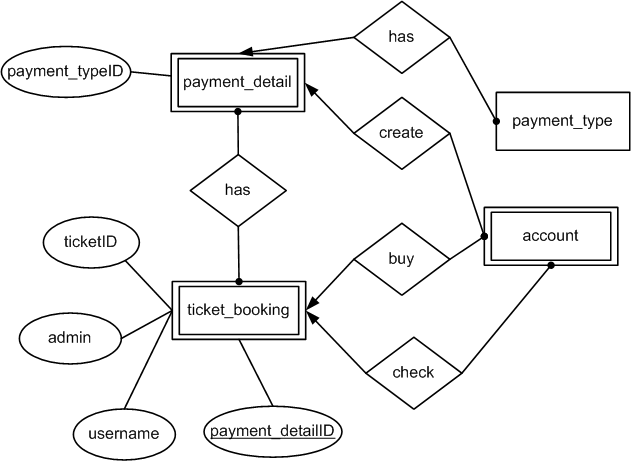
**payment_type**

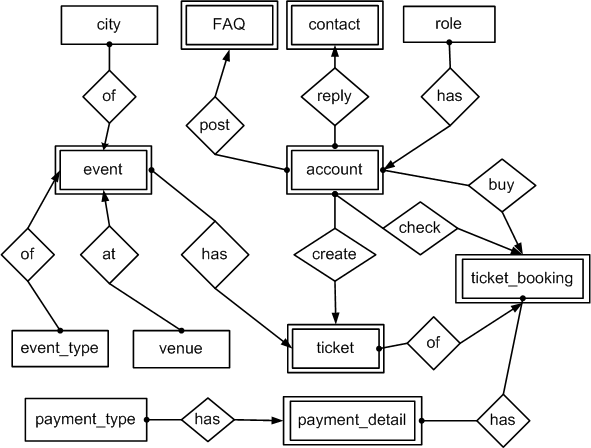
***9.2. Relationship***

** *9.2.1. Relationship Of Contact, FAQ, and Account***

** *9.2.2. Relationship Of Ticket***

** *9.2.3. Relationship Of Booking***

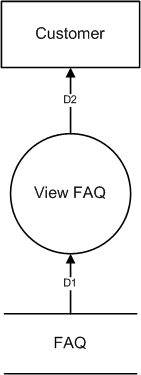
***9.2.4. Relationship Of Payment***

***9.2.5. Overview***

**10. DATA FLOW DIAGRAM (DFD)**

***10.1. Customer***

***10.1.1 View FAQ***



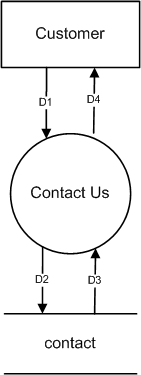
**Note:**

D1: List of FAQ.

D2: D1

***10.1.2 About Us (Not Available)***

***10.1.3 Contact Us***



**Note**

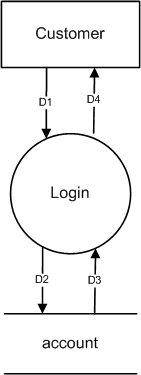
D1: Title, content, email.

D2: D1.

D3: True or False.

D4: Success or error message.

***10.1.4. Login***



**Note**

D1: Username, password.

D2: D1

D3: True or False.

D4: Success or error message.

***10.1.5. See List Of Ticket By Event***

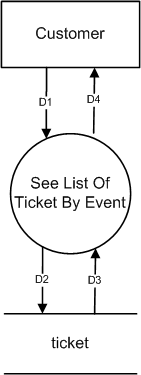
**Note**

D1: Event type, City.

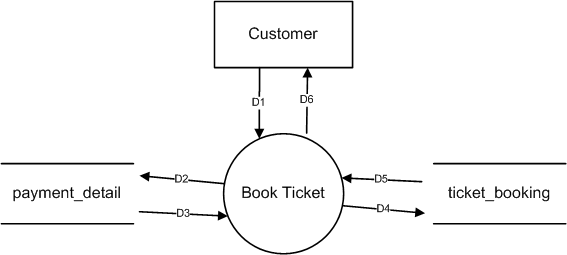
D2: D1

D3: List of ticket by event type (..and city if user chooses it).

D4: D3

****

***10.1.6. Book Ticket***



**Note**

D1: Ticket total, card number, payment type.

D2: Card number, payment type.

D3: Payment.

D4: Payment, price total, discount, username of customer.

D5: True or False.

D6: Success or error message.

***10.1.7. Check Available Ticket***

**Note**

D1: Ticket total, view date.

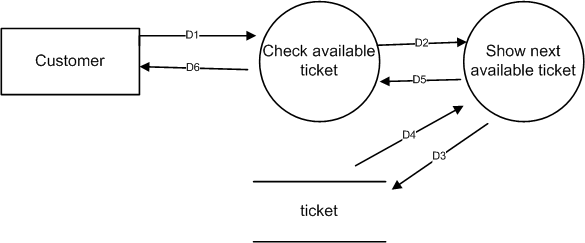
D2: Event.

D3: D2

D4: List of ticket by event.

D5: D4

D6: D5



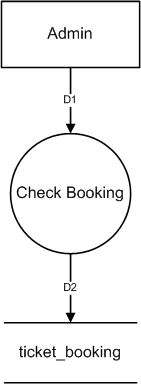
***10.2. Admin***

***10.2.1. Check booking***

**Note**

D1: Ticket.

D2: Ticket, accept status, username of admin.



***10.2.2. Manage Ticket***

***10.2.2.1. Create Ticket***

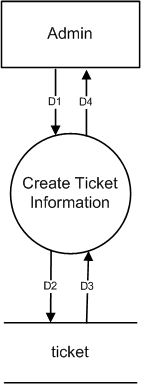
**Note**

D1: Price, promotion, discount,ticket total, view date, view time, event.

D2: Price, promotion, discount,ticket total, view date, view time, event, username of admin.

D3: True or False.

D4: Successful or Error message.

******

***10.2.2.2 Update Ticket***

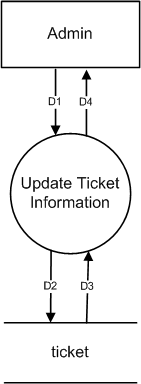
**Note**

D1: Price, promotion, discount,ticket total, view date, view time, event.

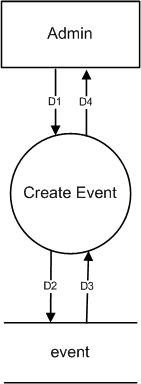
D2: D1

D3: True or False.

D4: Successful or Error message.

******

***10.2.2.3 Create Event***

******

**Note**

D1: Title, content, artist, image, city, venue, event type.

D2: D1

D3: True or False.

D4: Successful or Error message.

***10.2.3. Manage FAQ***

***10.2.3.1 Create FAQ***

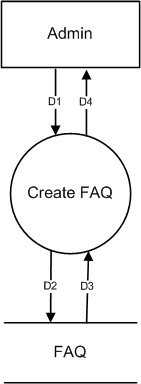
**Note**

D1: Question, answer.

D2: Question, answer, username of admin.

D3: True or False.

D4: Successful or Error message.

******

***10.2.3.2 Update FAQ***

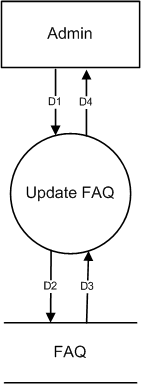
**Note**

D1: Question, answer.

D2: D1

D3: True or False.

D4: Successful or Error message.

******

***10.2.3.3 Delete FAQ***

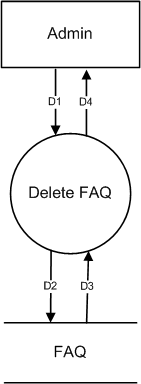
**Note**

D1: FAQ.

D2: D1

D3: True or False.

D4: Successful or Error message.

******

***10.2.3. Answer Contact Of Customer***

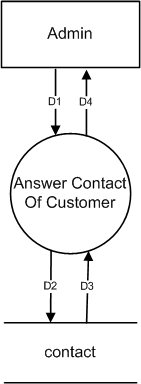
**Note**

D1: Answer.

D2: Answer, username.

D3: True or False.

D4: Successful or Error message.

****

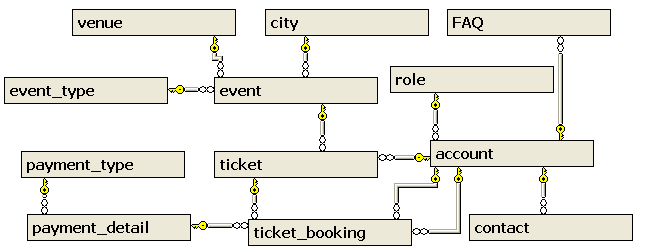
**11. CLASS DIARAM**

**12. TABLE DESIGN**

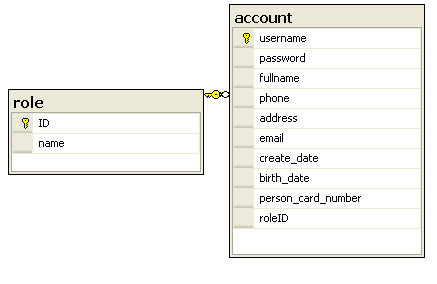
***12.1. Table List***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **Total** | **Primary Key** | **Foreign Key** |
| Account | User profile | 10 | Username | Role.roleID |
| City | City | 2 | ID | - |
| Contact | Contact list of Customer | 7 | ID | Account.username |
| FAQ | FAQ | 5 | ID | FAQ.username |
| Payment\_detail | Credit card or debit card | 3 | ID | Payment\_type.ID |
| Payment\_type | Credit or debit type | 2 | ID | - |
| Role | Admin, Customer | 2 | ID | - |
| Ticket | TicketID | 10 | ID | 1. Account.username  2. Event.ID |
| Ticket\_booking | - | 10 | ID | 1. Ticket.ID  2. Account.ID  3. Payment\_detail.ID |
| Event | EventID | 8 | ID | 1. Event\_type. ID  2. Venue.ID  3. City.ID |
| Event\_type | Default value as “Movie”, ”Sport”, ”Music”, “Drama”… | 2 | ID | - |
| Venue | Venue | 3 | ID | - |

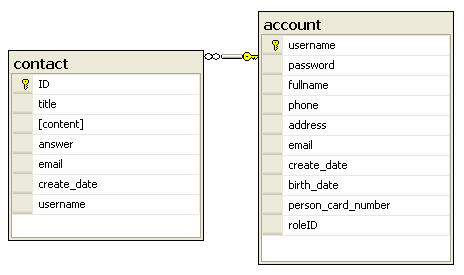
***12.2. Digram***

 ***12.2.1. Overview***

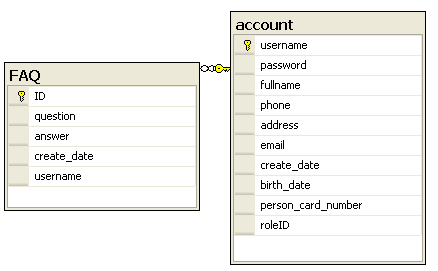
***12.2.2. Digram shows the relationship of Account***

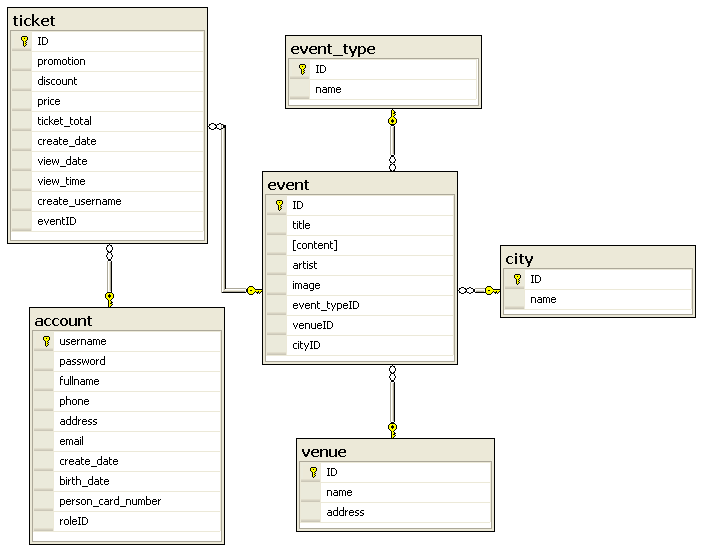
**

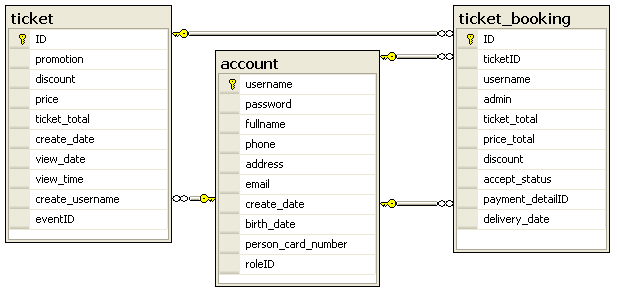
***12.2.3. Digram shows the relationship of Contact***



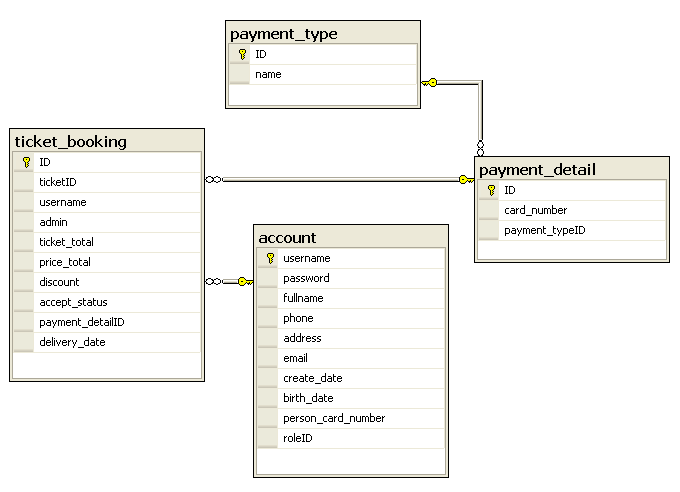
***12.2.4. Digram shows the relationship of FAQ***



** *12.2.5. Digram shows the relationship of Ticket***

***12.2.6. Digram shows the relationship of Booking***

***12.2.7.******Digram******shows******the relationship of Payment***



***12.3. Table Specification***

TABLE: Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| Username | Varchar(50) | No | - | Primary Key |
| Password | Varchar(50) | No | - | - |
| FullName | Nvarchar(50) | No | - | - |
| Phone | Varchar (15) | Yes | - | - |
| Address | Varchar(100) | No | - | - |
| Email | Varchar(100) | Yes | - | - |
| Create\_date | DateTime | Yes | - | - |
| Brith\_date | DameTime | Yes | - | - |
| Person\_card\_numer | Varchar(20) | No | - | - |
| RodeID | Int | Yes | - | ForeignKey |

TABLE: City

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key. |
| Name | Nvarchar(50) | No | - | - |

TABLE: Contact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key. |
| Title | Nvarchar(200) | No | - | - |
| Content | Nvarchar(500) | No | - | - |
| Answer | Nvarchar(500) | Yes | - | - |
| Email | Varchar(100) | No | - | - |
| Create\_Date | DateTime | Yes | - | - |
| UserName | Varchar(50) | Yes | - | Foreignkey |

TABLE: Event

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Title | Nvarchar(200) | No | - | - |
| Content | Nvarchar(100) | No | - | - |
| Artist | Nvarchar (200) | Yes | - | - |
| Image | Varchar(100) | Yes | - | - |
| Event\_TypeID | Int | Yes | - | ForeignKey |
| Value\_ID | Int | Yes | - | Foreignkey |
| City\_ID | Int | Yes | - | Foreignkey |

TABLE: Event\_Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Name | Nvarchar(100) | No | - | - |

TABLE: FAQ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Question | Nvarchar(200) | No | - | - |
| Answer | Nvarchar(500) | No | - | - |
| Create\_Date | DateTime | Yes | - | - |
| UserName | Varchar(50) | Yes | - | ForeignKey |

TABLE: Payment\_Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Card\_Number | Varchar(20) | No | - | - |
| Payment\_TypeID | Int | Yes | - | ForeignKey |

TABLE: Payment\_Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Name | Nvarchar(100) | No | - | - |

TABLE: Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No. | - | Primary Key |
| Name | Nvarchar(50) | No. | - | - |

TABLE: Ticket

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Promotion | Varchar(300) | Yes | - | - |
| Discount | Money | Yes | - | - |
| Price | Money | No | - | - |
| Ticket\_Total | Int | No | - | - |
| Create\_Date | DateTime | No | - | - |
| View\_Date | DateTime | No | - | - |
| View\_Time | Varchar(5) | No | - | - |
| Create\_UserName | Varchar(50) | Yes | - | ForeignKey |
| EventID | Int | Yes | - | ForeignKey |

TABLE: Ticket \_Booking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Ticket\_ID | Int | Yes | - | ForeignKey |
| UserName | Varchar(50) | Yes | - | ForeignKey |
| Admin | Varchar(50) | Yes | - | ForeingKey |
| Ticket\_Total | Int | No | - | - |
| Price\_Total | Money | Yes | - | - |
| Discount | Money | Yes | - | - |
| Accept\_Status | Char(1) | No | ‘1’ or ‘0’ | - |
| Payment\_DetailID | Int | Yes | - | - ForeignKey  - Unique Key |
| Delivery\_Date | Datetime | Yes | - | - |

TABLE:Venue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **ValueRange** | **Key** |
| ID | Int | No | - | Primary Key |
| Name | Nvarchar(100) | No | - | - |
| Address | Nvarchar(100) | No | - | - |

**13. GUI STANDARD DOCUMENT**

|  |  |
| --- | --- |
| **Property** | **Value** |
| Layout | Header, menu, banner, content, footer. |
| Background color for body. | #e8ecef |
| Background color for theme. | White+#176c9c |
| Form Theme and Color Scheme. | None |
| **Header** | |
| Font | Tahoma;white;11 pixels; |
| Alignment | Right |
| Background Color | #282b34 |
| **Menu** | |
| Font | Tahoma;white;12 pixels;bold |
| Alignment | Left |
| Background Color | #282b34 |
| **Content** | |
| Title-Font-Size | 16 pixel |
| Title-Font-Color | #176c9c |
| Title-Font-Style | Tahoma |
| Content-Font-Size | 11 pixel |
| Content-Font-Color | #323232 |
| Content-Font-Style | Tahoma |
| Title-Alignment | Left |
| Content-Alignment | Justify |
| Background Color | White |
| **Footer** | |
| Font | Center; #808080;10px; |
| Background-color | #282b34 |
| **Form** | |
| Background color of controls on the form. | Default |
| Foreground color of controls on the form. | Default |
| **Link** | |
| Link-Font-Color | Contentlayout: Silver  Menu layout: White |
| Link-Font-Style | Tahoma |
| Link-Font-Move | Menu layout: Gray |
| Link-Font-Visited | Menu layout: Gray |
| Text-decoration | None |

**14. SITEMAP**

**Login**

Hello

Every

Body

…

**Home**

Hello

Every

Body

…

**About Us**

Hello

Every

Body

…

**Contact Us**

Hello

Every

Body

…

**FAQ**

Hello

Every

Body

…

**Register**

Hello

Every

Body

…

**Admin**

Hello

Every

Body

…

**Show Tickets**

Hello

Every

Body

…

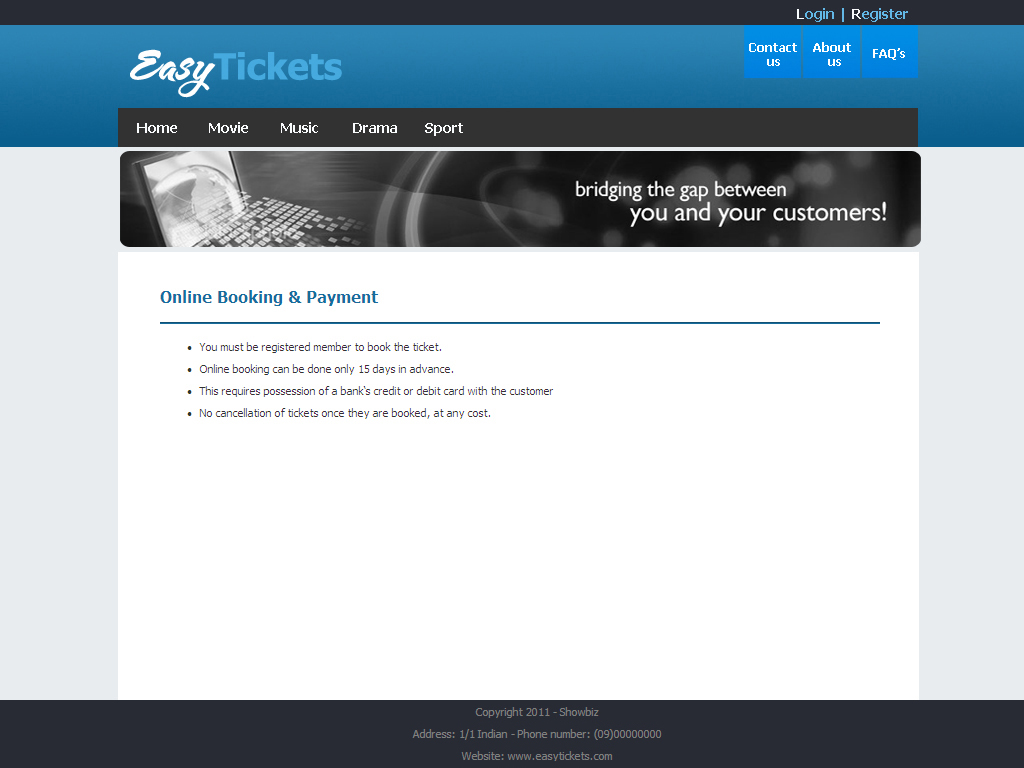
**Booking**

Hello

Every

Body

…

******

**15. INTERFACE DESIGN PROGRAM**

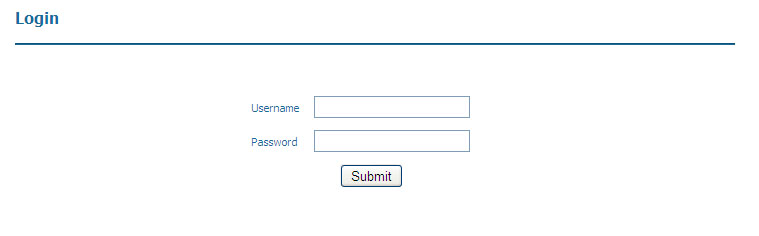
***15.1 Home (Template)***

******

**15. INTERFACE DESIGN PROGRAM**

***15.2. About Us Page***

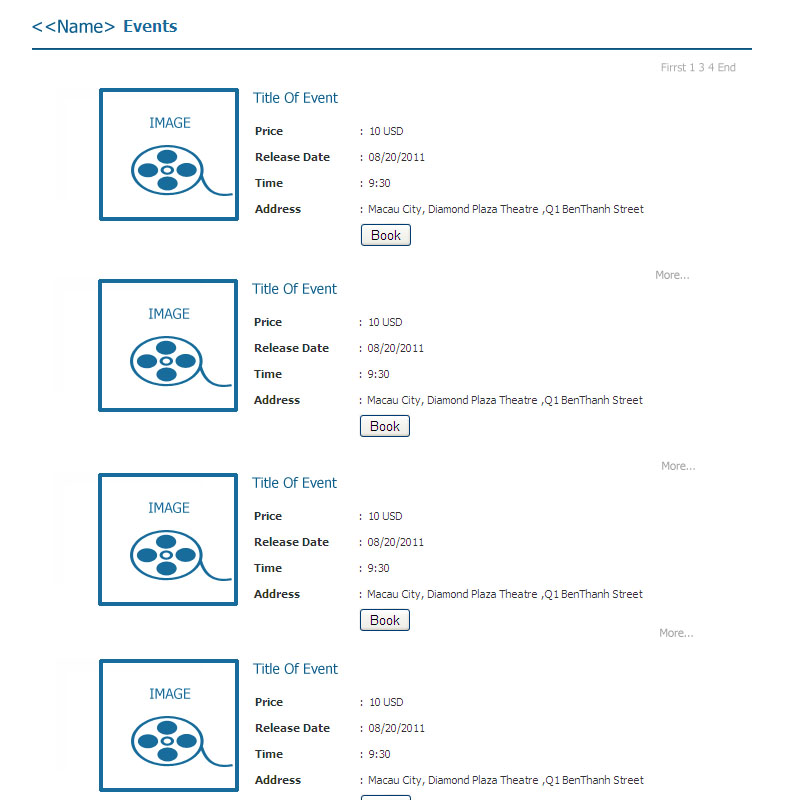
***15.3. Login Form***



Screen description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Login | | | |
| **Description** | User must login to view ticket and book ticket. | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| Username (1) | Username | Textbox | - | - |
| Password (2) | Password | Textbox | - | - |
| Submit (3) | - | Button | - | Redirect to home page, you can view information about ticket and book ticket |

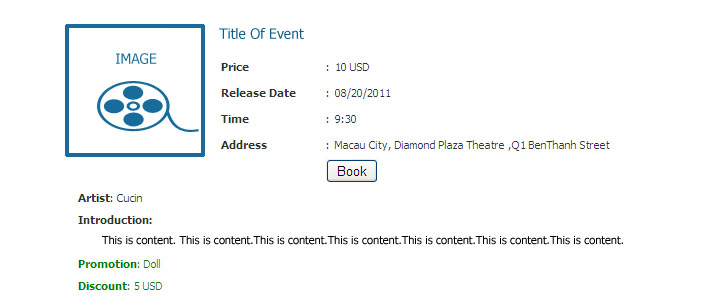
***15.4. Show Tickets Form***



Screen description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Show Tickets | | | |
| **Description** | Show all ticket by event to user can booking | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| Page number(1) | Link to a next page | Hyperlink | - | - |
| More(2) | Open detail of ticket | Hyperlink | - | - |
| Book(3) | - | Button | You must login | Redirect to booking page |

***15.5. View Detail Of Ticket***



Close

Screen description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | View Detail Of Ticket | | | |
| **Description** | Show all ticket by event to user can booking | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| Close(\*) | Close detail of ticket | Hyperlink | - | - |

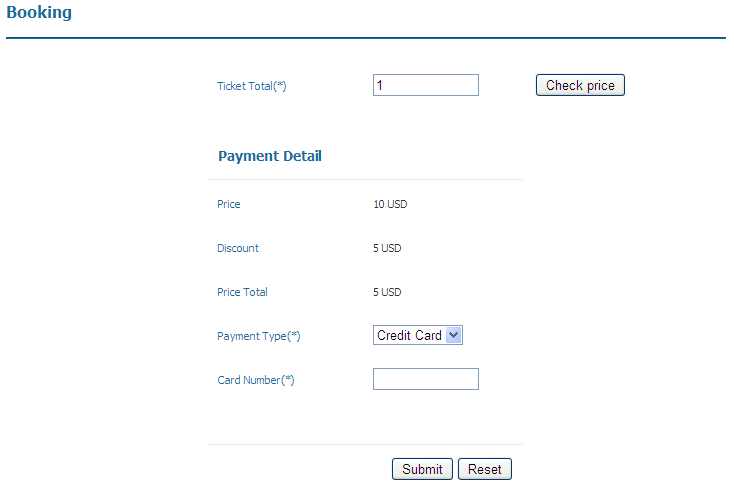
***15.6. Show Next Available Ticket***



Screen description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Show Next Available Ticket | | | |
| **Description** | If ticket is not available, this form wil show next available ticket | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| Time(\*) | This is next time of available ticket | Hyperlink | - | - |

***15.7. Book Ticket***

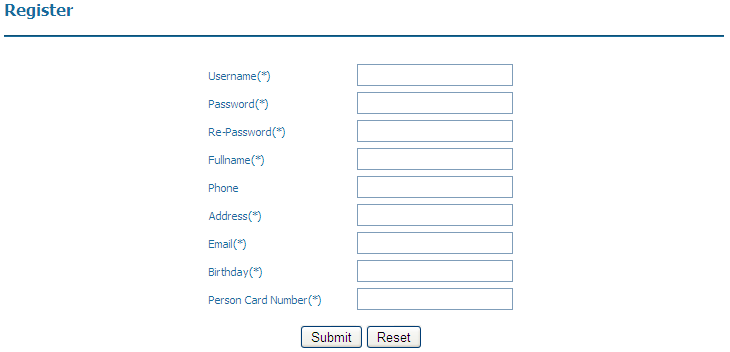


Screen description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Book Ticket | | | |
| **Description** | Customer book the ticket | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| Ticket Total(1) | Ticket Total | Textbox | - Required to enter.  - Only number. | - |
| Check Price(2) | Update price by ticket total. | Button | - | Price, discount was shown. |
| Price(3) | Price that customer must pay for a ticket. | Label | - | - |
| Discount(4) | Discount | Label | - | - |
| Price Total(5) | Price total that customer must pay. | Label |  | - |
| Payment Type(6) | - | Combobox | - | - |
| Card Number(7) | - | Textbox | Required to enter | - |
| Submit(8) | - | Button | - | Redirect to show ticket page, you can view other ticket. |
| Reset(9) | - | Button | - | Remove data on form. |

***15.8. Additional***

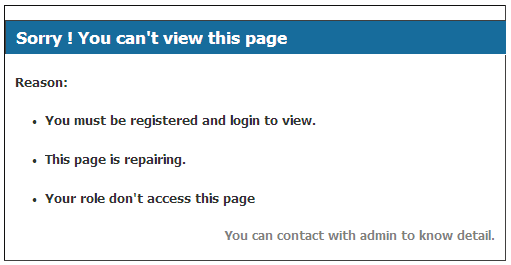
***15.8.1 Register***



Screen description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Register | | | |
| **Description** | User want login must register. | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| Username(1) | Username is account which user use to login. | Textbox | Required to enter. | - |
| Password(2) | Use to verify account. | Textbox | Required to enter. | - |
| Re-password(3) | Check correct. password with password have entered. | Textbox | Required to enter. | - |
| Fullname(4) | Name of customer | Textbox | Required to enter. | - |
| Phone(5) | Phone of customer | Textbox | Data must be integer. | - |
| Address(6) | - | Textbox | Required to enter. | - |
| Email(7) | - | Textbox | Required to enter. | - |
| Birthday(8) | - | Textbox | Required to enter. | Click into it to show date table. |
| Person Card Number(9) | - | Textbox | - Required to enter.  - Data must be integer. | - |
| Submit(10) | - | Button | - | Information of customer was stored. |
| Reset(11) | - | Button | - | Remove data on form. |

***15.8.2. Alert Form***



\*

Screen description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Alert Error | | | |
| **Description** | User access page which don’t allow, application will redirect to this form | | | |
| **Control** | **Description** | **Type** | **Requirement** | **Event** |
| You can contact with admin to know detail(\*) | Contact to admin | Hyperlink | - | Redirect to contact us |

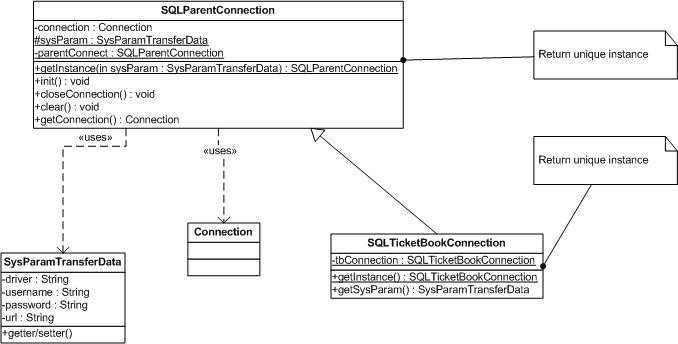
**16. TASK SHEET**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Project Title** | **Activity Plan Prepared By:** | **Date of Preparation of Activity Plan:** | | | |
| **No.** | **Task** | **Actual Start Date**  **(mm/dd/yyyy)** | **Actual Days** | **Team Member Names** | **Status** |
| 01 | Analyze & Design Database | ET | Pham Hong Tam | 07/12/2011 | 6 | Pham Hong Tam  Nguyen Huu Sang  Tran Phan Quoc Hai  Nguyen Duc Viet | Finish |
| 02 | Design Sitemap, Use Case | ET | Pham Hong Tam | 07/13/2011 | 7 | Nguyen Huu Sang  Tran Phan Quoc Hai  Pham Hong Tam | Finish |
| 03 | Design ER, Class Diagram | ET | Pham Hong Tam | 07/14/2011 | 10 | Nguyen Huu Sang | Finish |
| 04 | Design Flowchart, Sequence Diagram | ET | Pham Hong Tam | 07/16/2011 | 10 | Nguyen Huu Sang | Finish |
| 05 | Design Layout, Css | ET | Pham Hong Tam | 07/18/2011 | 3 | Pham HongTam | Finish |
| 06 | Collect Data For Database | ET | Pham Hong Tam | 07/18/2011 | 10 | Nguyen Duc Viet | Finish |
|  | **CODING** |  |  |  |  |  |  |
| 07 | About Us | ET | Pham Hong Tam | 07/18/2011 | 1 | Nguyen Huu Sang | Finish |
| 08 | Login | ET | Pham Hong Tam | 07/18/2011 | 1 | Pham Hong Tam | Finish |
| 09 | Register Member | ET | Pham Hong Tam | 07/18/2011 | 2 | Nguyen Huu Sang | Finish |
| 10 | Contact Us | ET | Pham Hong Tam | 07/18/2011 |  | Tran Phan Quoc Hai |  |
| 11 | View FAQ | ET | Pham Hong Tam | 07/18/2011 |  | Tran Phan Quoc Hai |  |
| 12 | Show Tickets | ET | Pham Hong Tam | 07/20/2011 | 3 | Pham Hong Tam | Finish |
| 13 | Book The Ticket | ET | Pham Hong Tam | 07/20/2011 | 3 | Pham Hong Tam | Finish |
| 14 | Check Booking Tickets | ET | Pham Hong Tam | 07/21/2011 | 1 | Pham Hong Tam | Finish |
| 15 | Manage Ticket | ET | Pham Hong Tam |  |  |  |  |
|  | Insert Ticket | ET | Pham Hong Tam | 07/29/2011 | 3 | Pham Hong Tam |  |
|  | Update Ticket | ET | Pham Hong Tam | 07/29/2011 | 5 | Pham Hong Tam |  |
| 16 | Manage FAQ | ET | Pham Hong Tam |  |  |  |  |
|  | Insert FAQ | ET | Pham Hong Tam | 07/30/2011 | 3 | Tran Phan Quoc Hai |  |
|  | Update FAQ | ET | Pham Hong Tam | 07/30/2011 | 6 | Tran Phan Quoc Hai |  |
|  | Delete FAQ | ET | Pham Hong Tam | 08/05/2011 | 1 | Tran Phan Quoc Hai |  |
| 17 | Answer Contact | ET | Pham Hong Tam | 08/05/2011 | 2 | Tran Phan Quoc Hai |  |
| 18 | Testing | ET | Pham Hong Tam | 08/01/2011 | 7 | Nguyen Duc Viet |  |
| 19 | Installation Guide  User Guide | ET | Pham Hong Tam | 08/9/2011 | 2 | Nguyen Duc Viet |  |
| 20 | Finish Document | ET | Pham Hong Tam | 08/01/2011 | 7 | Nguyen Huu Sang |  |

**17. UNIT TESTING CHECK LIST**

**18. FINAL CHECK**

1. **CONNECTION MODEL**

****

**REFERENCE**

*Figure1: Create object to connect with Database*

**Pattern:** Singleton pattern

**Explain:** Connection is created with a instance**,** avoid for programer open persistent connection and redundant memory.

**Usage:**

String sqljdbc4Driver=” sql.com.driver…”;

String servername=”servername:databasename”;

String user=”sa”;

String pass=”123456”;

SysParamTransferData driver=new SysParamTransferData(sqljdbc4Driver,servername,user,pass);

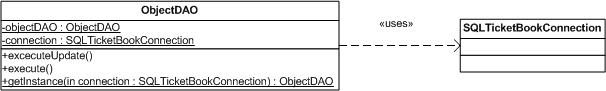
SQLParentConnection connSQL=SQLParentConnection.getInstance(driver);

Connection con=connSQL.getConnection()

OR

SQLTicketBookConnection().getInstance().getConnect() ; //*use to connect to “ticketbook” database*

1. **EXECUTE DATABASE MODEL**

****

*Figure 2: ObjectDAO execute database from Connection Input*

**Pattern:** None.

**Explain:** If connection open, ObjectDAO can execute function on database. Help coder easy control the errors connection.

**Usage:**

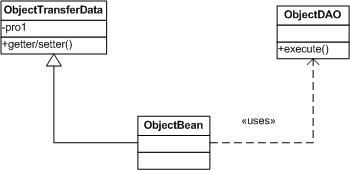
SQLTicketBookConnection conn=SQLTicketBookConnection().getInstance();

ObjectDAO objectDAO=ObjectDAO.getInstance(conn);

// You can call execute function.

objectDAO.execute(); // OR objectDAO.executeUpdate();

1. **ADAPTER MODEL FOR BMP**

****

*Figure 3: ObjectBean as Adapter for ObjectTransferData*

**Pattern:** Adapter pattern.

**Explain:** ObjectBean will extends ObjectTransferData and use function of ObjectDAO.

**Usage:**

class ObjectBean{

void execute(){

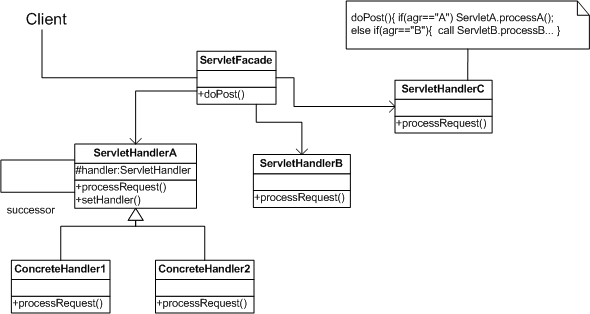
ObjectDAO.getInstance(SQLTicketBookConnection.getInstance()).execute(this.pro)

// pro is a property of ObjectTransferData.

}

}

1. **SERVLET MODEL**

****

*Figure 4: Client send request to ServletFacade, it will call sub servlet to process.*

**Pattern:** Facade + Chain Of Responsibility pattern.

**Explain:** User submit from jsp to Servlet (Façade). It will send request to a other servlet to process. If servlet doesn’t process, it will send to a other servlet… Until request is processed.

**Usage:** None.