|  |
| --- |
| **FPT-aptech computer education** |
| eProject Document |
| ONLINE BUS TICKET RESERVATION SYSTEM |
|  |
| |  |  | | --- | --- | | **Group 2** | | | **Group Member** | BaHL\_C00106<Eproject\_Code>  ThanhDV\_C00115<Eproject\_Code>  HungPV\_C00100<Eproject\_Code>  TrungBD\_C00097<Eproject\_Code>  DatLQ\_C00112<Eproject\_Code> | | **Instructor** | ThiDK | | **Batch** | C0809G | | **Semester** | 3|4 | |
|  |

|  |
| --- |
| - Ha noi, 10/2010 - |

**THE TABLE OF CONTENTS**

# 

# Introduction

SRC Travel Agencies is a domestic privatized transportation company that runs the vehicles all over the country. They have several branches at different locations of the country, so that they can provide the transportation facilities between the places. They want an online application to be developed, where the details of the bookings done and the customer will be updated from time to time and one can track the details of the available seats immediately.

# Problem Definition

## The Proposed System

< Describe the system under developed. This can be paraphrase from the Customer’s Requirement Statement sent from India>

## Boundaries of the System

< List the scope\boundaries of the System-under-developed. This can be paraphrase from the Customer’s Requirement Statement sent from India>

## Development Environment

<Describe the environment for system development. Include software and hardware requirements>

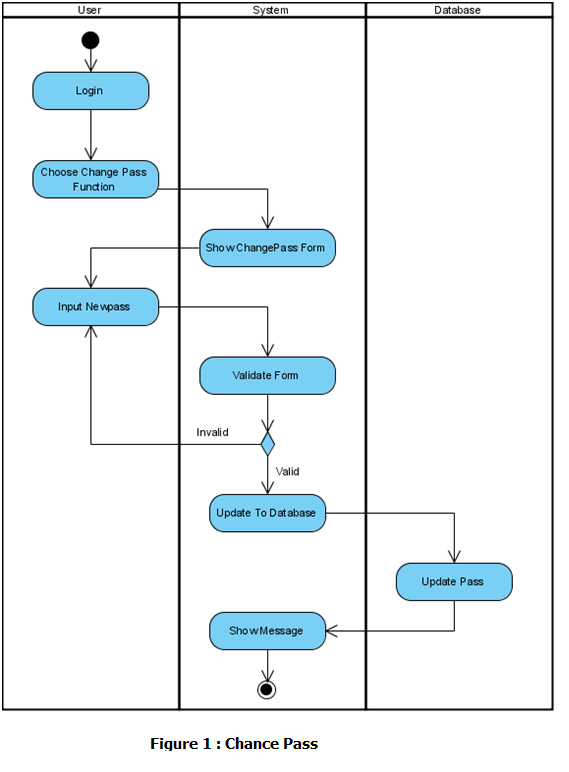
# Requirements and Business Flow

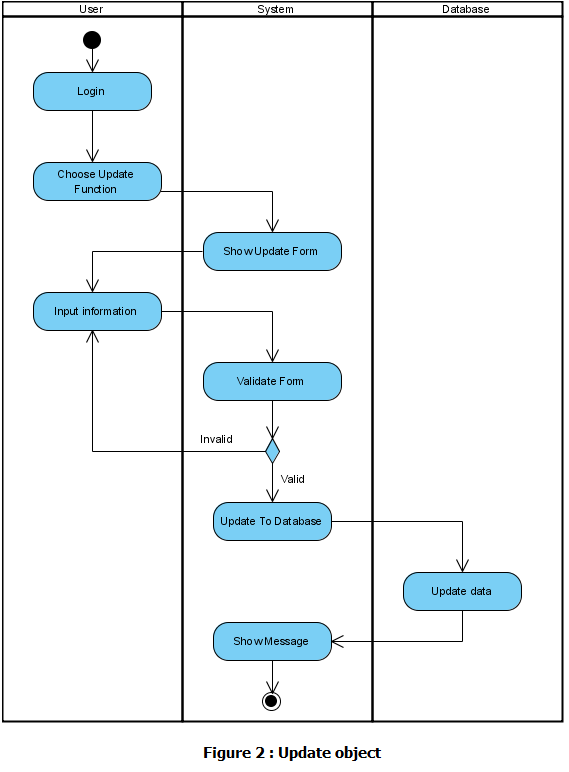
<In this section you will provide the requirements you developed against. This section should be useful, so only provide enough information (requirements, diagrams) to help with the implementation of the software. Focus on the “non-obvious” aspects. If you are dealing with a business process, activity diagrams are good. Making a set of requirements that is compact, simple and covers all the key aspects of the software can. Don’t do the work if you don’t need it!>

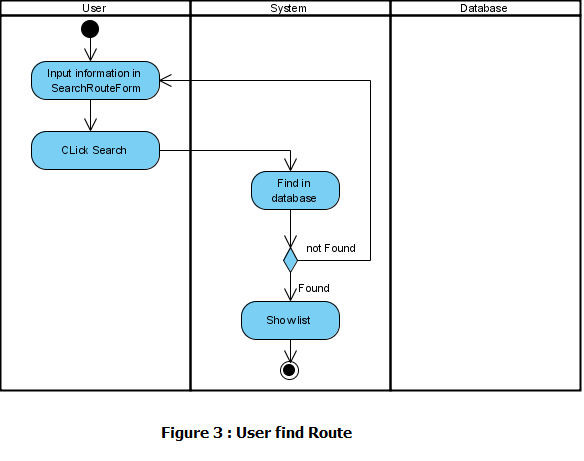
* 1. Customer Requirement Specification

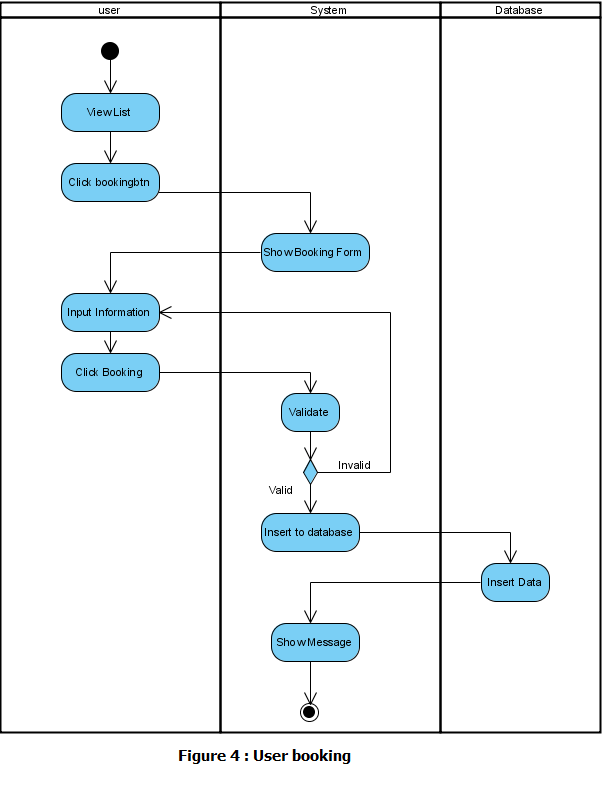
### System administrator

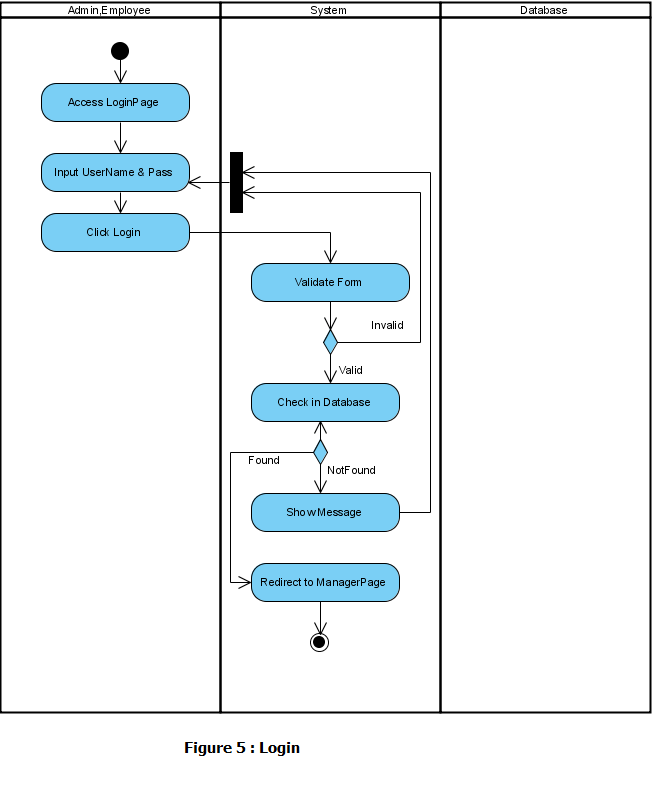
## Activity Diagram

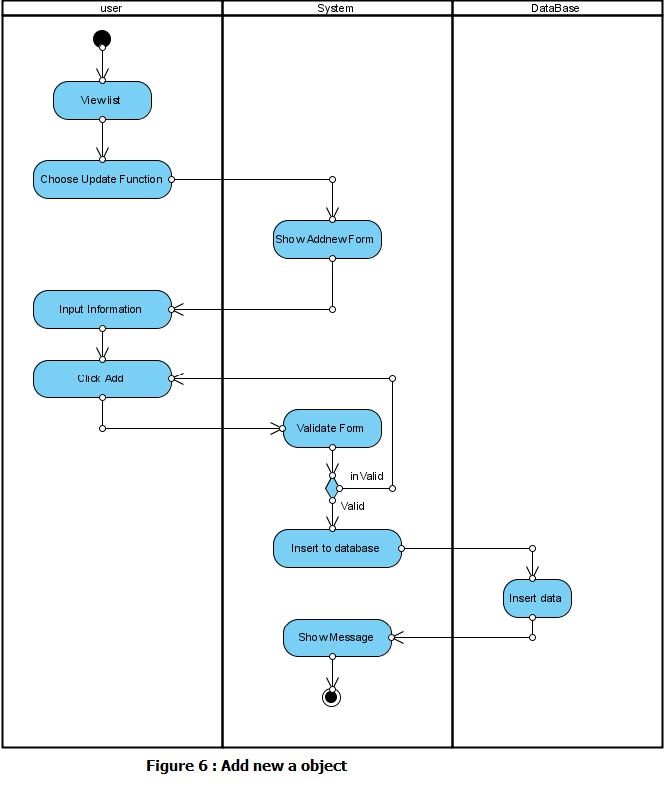


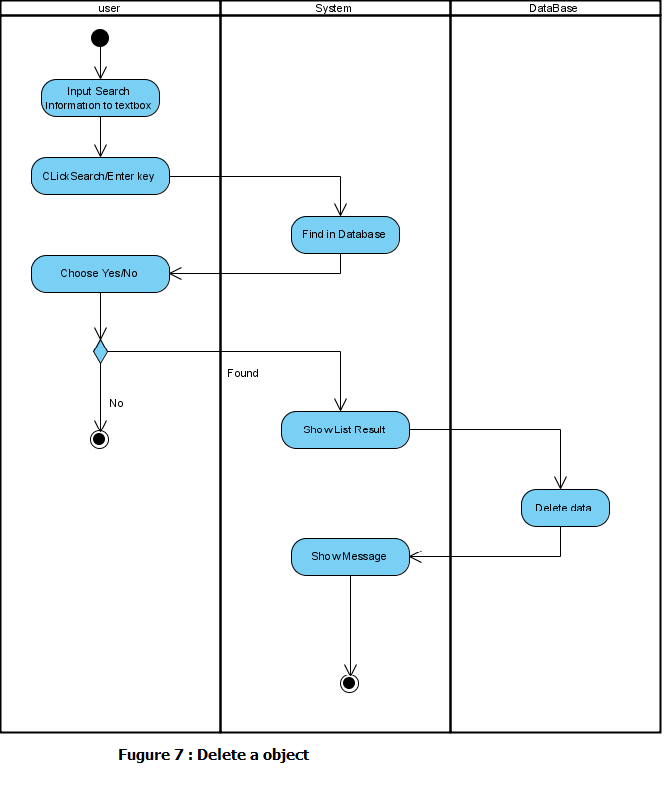


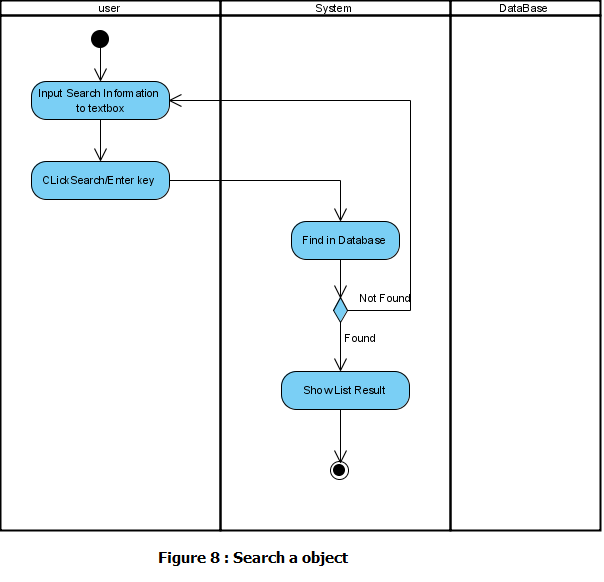




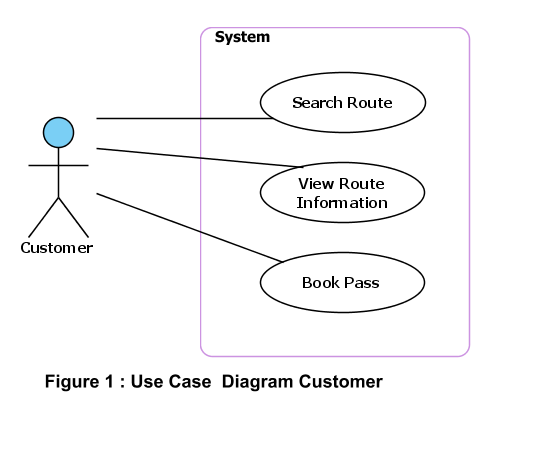


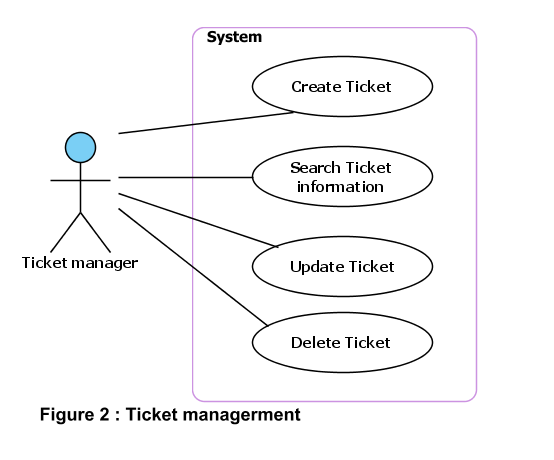


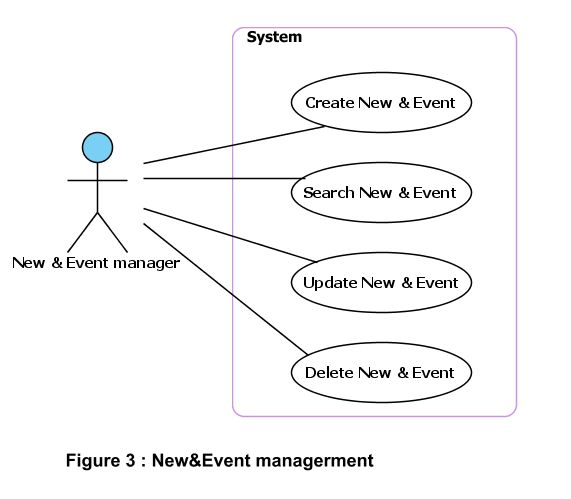


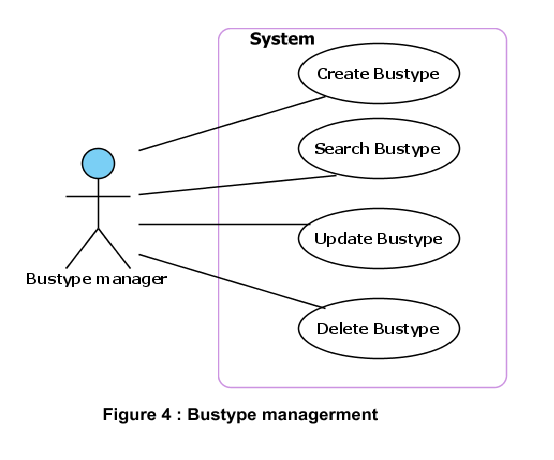


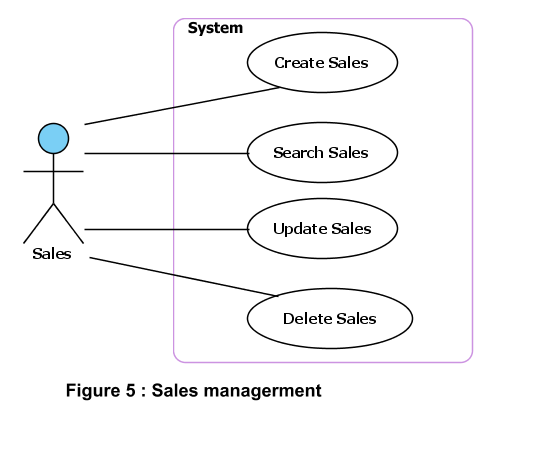
## Use Case Diagram

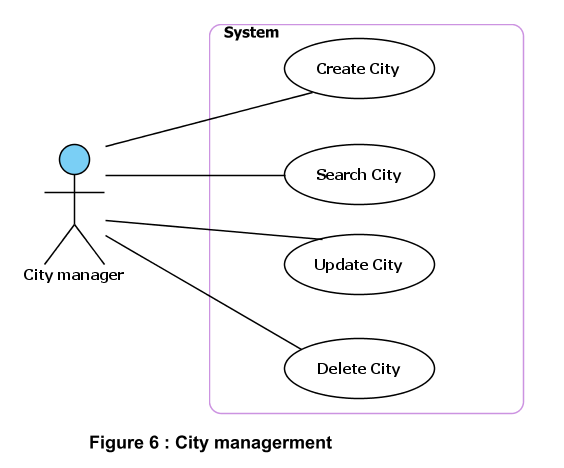


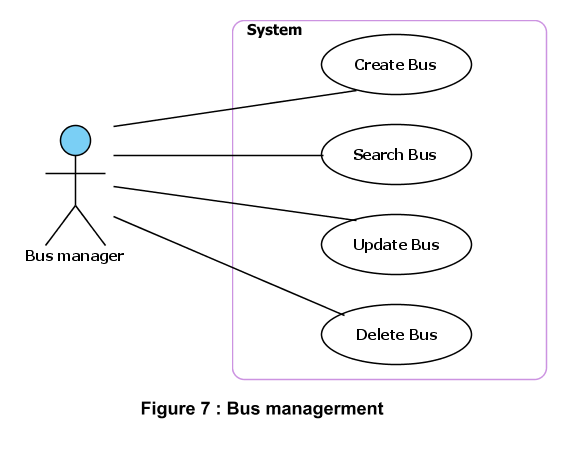


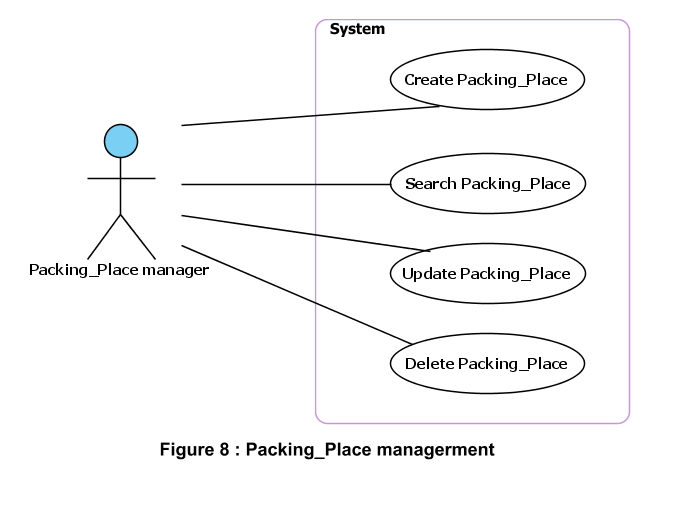


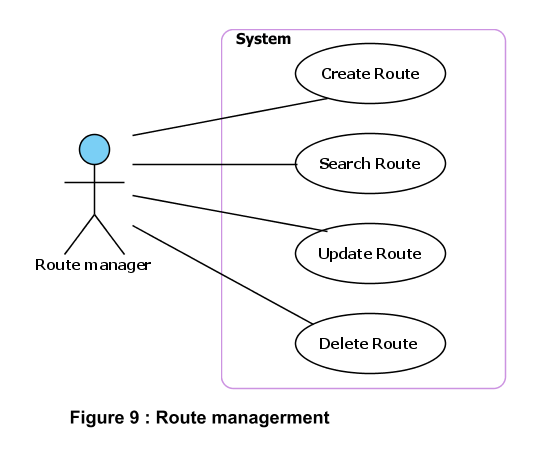


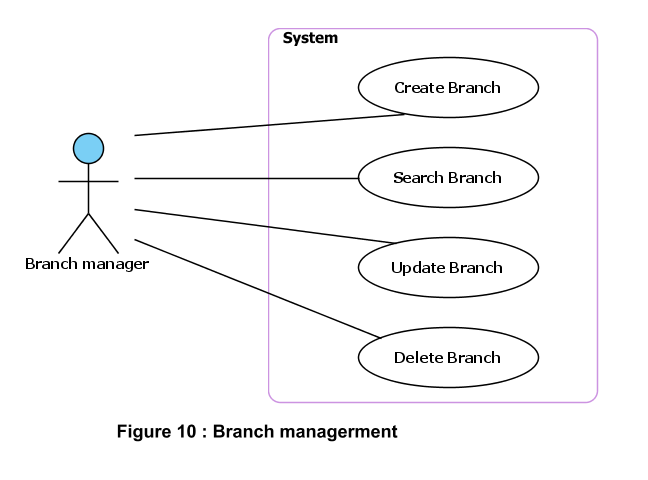


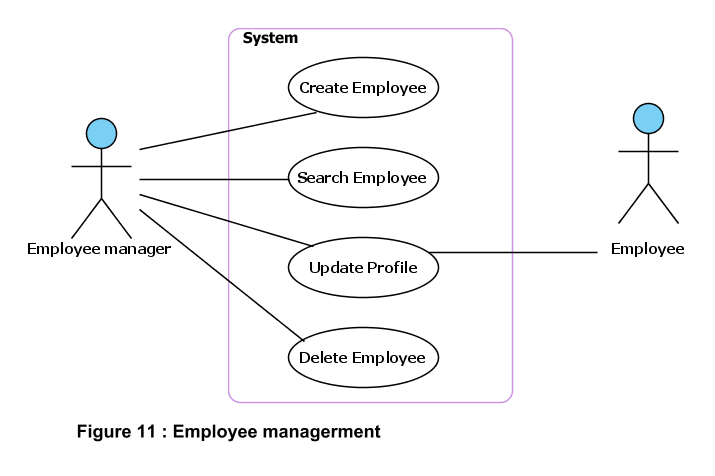


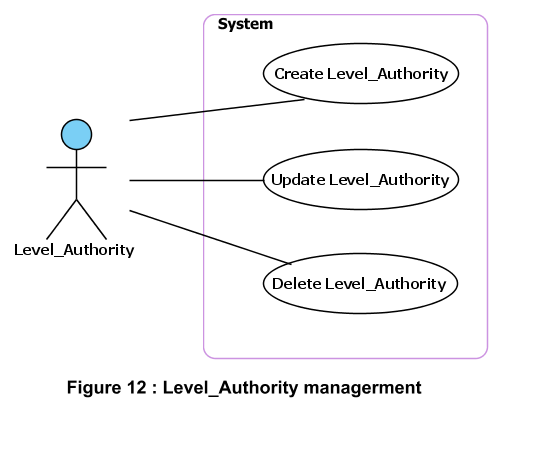












## Use Case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | | |
| **Use-case No.** | UC001 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Search Route | | | |
| **Author** |  | | | |
| **Date** |  | **Priority** | Normal | |
| **Actor:**  Customer  **Summary:**  Visited and search Route information **.**  **Goal:**  Customers can find detailed information about the trip of their choice.Customers can book online through the company's website.  **Triggers**  **Preconditions:**  Customer don’t need register but customer can direct book online.  **Post Conditions:**  **Main Success Scenario:**  1. Customer choose start place,end place,start date of trip  2. System display details information about trip  3. Customer input account information into form register.  4. System confirm information   5. Insert the customer information to database  **Alternative Scenario:**  **Exceptions:**  **Relationships:**  **Business Rules:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | | |
| **Use-case No.** | UC002 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create Ticket | | | |
| **Author** |  | | | |
| **Date** |  | **Priority** | Normal | |
| **Actor:**  Ticket manager (Employee)  **Summary:**  Create new the ticket information for the Customer  **Goal:**  Ticket manager can create the ticket for the customer  **Triggers**  **Preconditions:**  Ticket manager must login to the system  **Post Conditions:**  After Ticket manager created ticket, customers can use to join trip.  **Main Success Scenario:**  1. Ticket manager enters information of trip (customer information and bus information)  2. Ticket manager create new the ticket.  3. System determines the validity of customer information  4. System confirm action   5. Insert the new the ticket to database  **Alternative Scenario:**  **Exceptions:**  **Relationships:**  Ticket manager  **Business Rules:** | | | | |

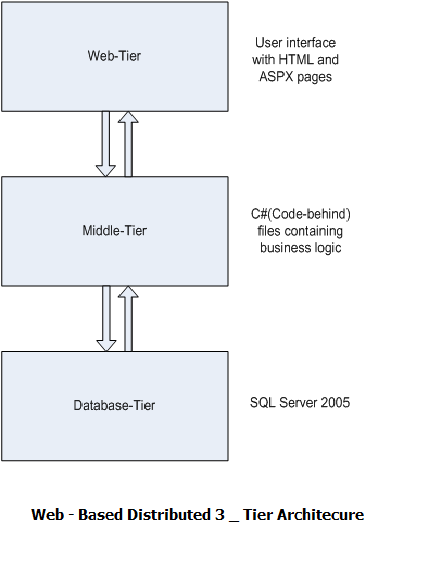
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | | |
| **Use-case No.** | UC003 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Search Ticket | | | |
| **Author** |  | | | |
| **Date** |  | **Priority** | Normal | |
| **Actor:**  Ticket manager (Employee)  **Summary:**  Find information related to tickets  **Goal:**  Find information related to tickets to easily manage the status of customer bookings  **Triggers**  **Preconditions:**  Ticket manager must login to the system  **Post Conditions:**  After Ticket manager created ticket, Ticket manager can use to find information..  **Main Success Scenario:**  1. Ticket manager view information of trip (customer information and bus information)  2. Ticket manager enter customer name or ticket number into text box.Then click button search.  3. System to verify the information entered and compare to list tickets in the database to provide details of the ticket to find.  **Alternative Scenario:**  **Exceptions:**  **Relationships:**  Ticket manager  **Business Rules:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | | |
| **Use-case No.** | UC004 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Update Ticket | | | |
| **Author** |  | | | |
| **Date** |  | **Priority** | Normal | |
| **Actor:**  Ticket manager (Employee)  **Summary:**  Update the Ticket information of Ticket.  **Goal:**  Edit ticket details information of ticket.  **Triggers**  **Preconditions:**  Ticket manager (Employee) must login to the system  **Post Conditions:**  Employee can updated , after admin agrees for it  **Main Success Scenario:**  1. Employee login  2. Employee select his account  3. Edit information that he want to change  4. System determines the validity of the information  5. System confirm action  6. Admin agrees the account  7. Update into the database  **Alternative Scenario:**  **Exceptions:**  **Relationships:**  Account Manager  **Business Rules:** | | | | |

# Design

## System Architecture

The application will be made of a Web-based distributed three-tier architecture to support multiple user transaction at the same time



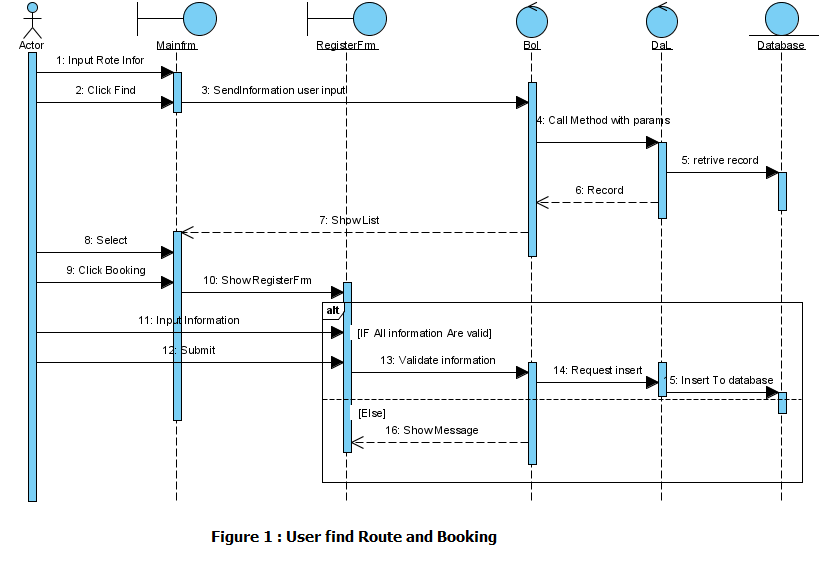
## Class Diagram

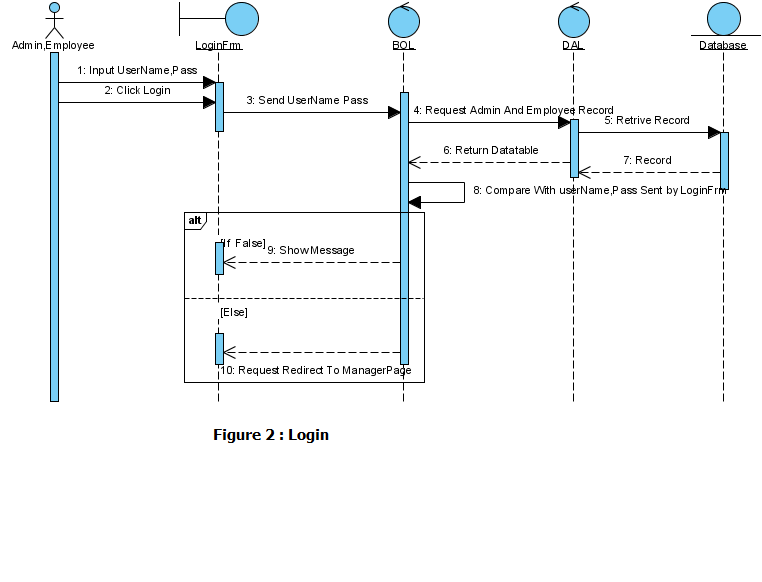
<Provide class diagrams for the project>

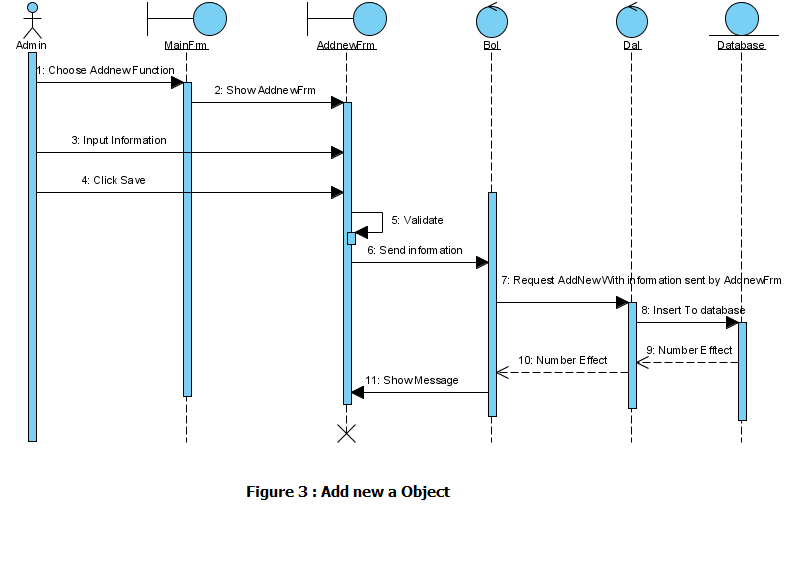
## Class Diagram Explanation

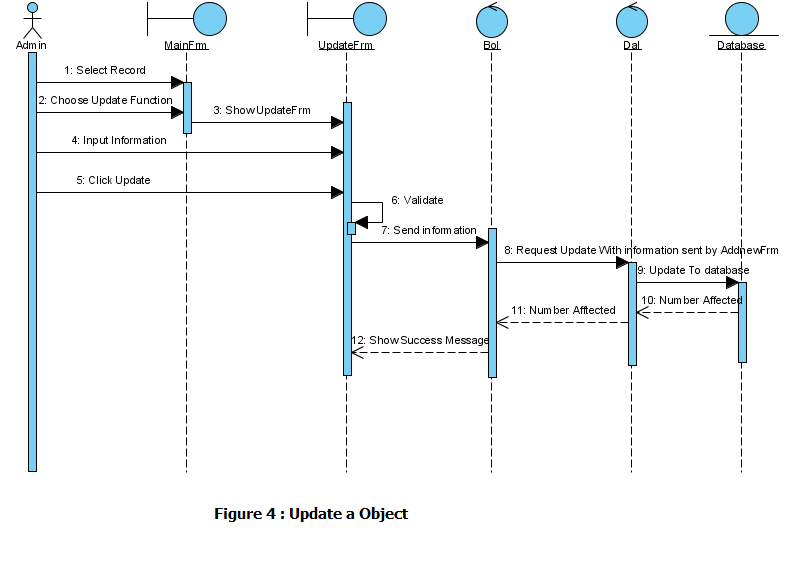
<Provide brief explanation about the class diagram above. You do not need to explain “obvious” parts of your class diagram. For example, I know what a “Login” class is. Don’t say “The login class was created to store login information.”>

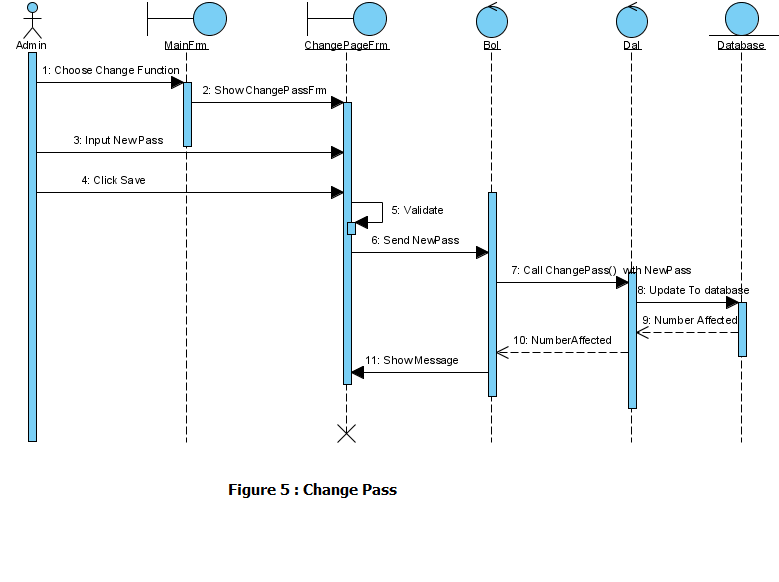
## Sequence Diagram (Optional)

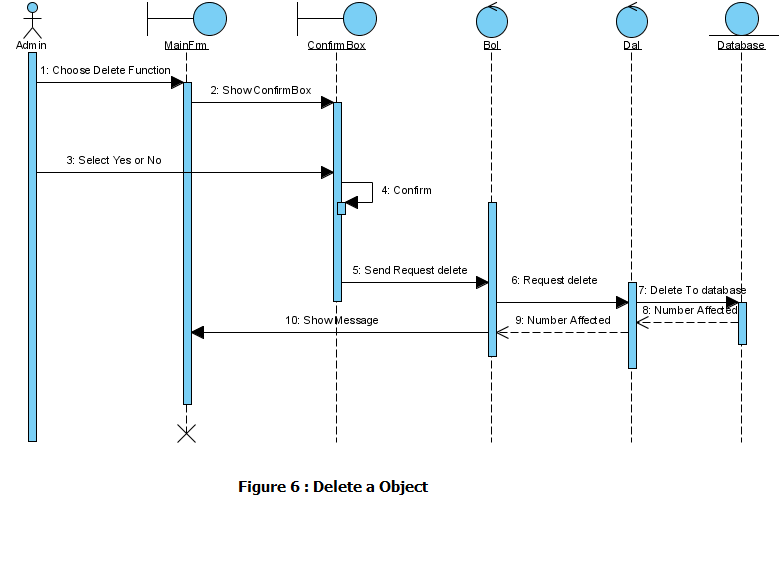


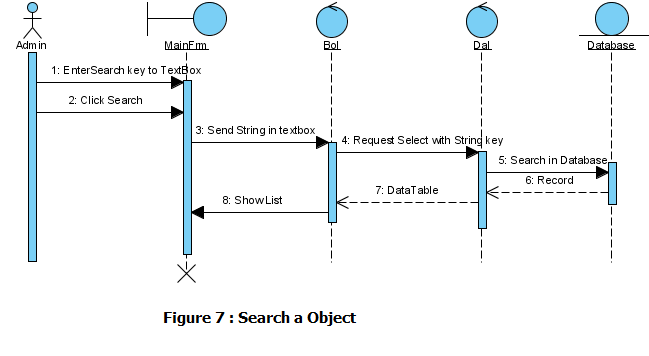




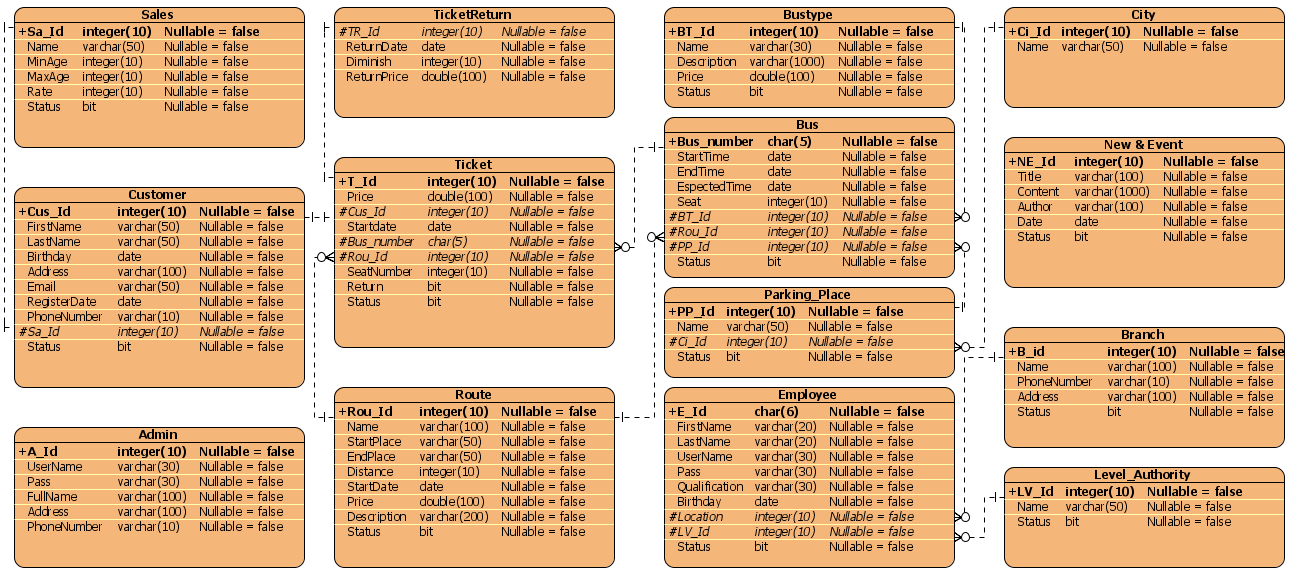








## Entity Relationship Diagram



## Database Design

 Bustype

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| BT\_Id | integer(10) | PK | No | Code of bustype |
| Name | varchar(30) |  | No | Name of Bustype |
| Description | varchar(1000) |  | No | Description of Bustype |
| Price | double(100) |  | No | Price of Bustype |
| Status | bit(0) |  | Yes | Status activity |

 City

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| Ci\_Id | integer(10) | PK | No | Code of city |
| Name | varchar(50) |  | No | Name of City |

 Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| Rou\_Id | integer(10) | PK | No | Code of route |
| Name | varchar(100) |  | No | Name of Route |
| StartPlace | varchar(50) |  | No | StartPlace of Route |
| EndPlace | varchar(50) |  | No | End place of Route |
| Distance | integer(10) |  | No | Distance of Route |
| StartDate | date(0) |  | No | StartDate of Route |
| Price | double(100) |  | No | Price of Route |
| Description | varchar(200) |  | No | Description of Route |
| Status | bit(0) |  | Yes | Status open route |

 Parking\_Place

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| PP\_Id | integer(10) | PK | No | Code of parking\_place |
| Name | varchar(50) |  | No | Name of Parking\_Place |
| Ci\_Id | integer(10) | FK ([City.Ci\_Id](#2M5mxdSAUFkyUwK7)) | No | Code of city |
| Status | bit(0) |  | Yes | Status bus put |

 Bus

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| Bus\_number | char(5) | PK | No | Code of Bus |
| StartTime | date(0) |  | No | Start time of trip |
| EndTime | date(0) |  | No | End time of trip |
| EspectedTime | date(0) |  | No | Espected time of trip |
| Seat | integer(10) |  | No | Total seat of Bus |
| BT\_Id | integer(10) | FK ([Bustype.BT\_Id](#.22EJdSAUFkyUwkq)) | No | Code of bustype |
| Rou\_Id | integer(10) | FK ([Route.Rou\_Id](#m8AoJdSAUFkyUwRX)) | No | Code of route |
| PP\_Id | integer(10) | FK ([Parking\_Place.PP\_Id](#oUNEJdSAUFkyUww_)) | No | Code of parking\_place |
| Status | bit(0) |  | Yes | Status start of bus |

 Sales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| Sa\_Id | integer(10) | PK | No | Code of sales |
| Name | varchar(50) |  | No | Name of Sales |
| MinAge | integer(10) |  | No | Min age of Sales |
| MaxAge | integer(10) |  | No | Max age of Sales |
| Rate | integer(10) |  | No | Rate discount of Sales |
| Status | bit(0) |  | Yes | Status discount apply |

 Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| Cus\_Id | integer(10) | PK | No | Code of customer |
| FirstName | varchar(50) |  | No | FirstName of Customer |
| LastName | varchar(50) |  | No | LastName of Customer |
| Birthday | date(0) |  | No | Birthday of Customer |
| Address | varchar(100) |  | No | Address of Customer |
| Email | varchar(50) |  | No | Email of Customer |
| RegisterDate | date(0) |  | No | Ticket register date of Customer |
| PhoneNumber | varchar(10) |  | No | PhoneNumber of Customer |
| Sa\_Id | integer(10) | FK ([Sales.Sa\_Id](#7CYBxdSAUFkyUwum)) | No | Code of Sales |
| Status | bit(0) |  | Yes | Status account maintain |

 Branch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| B\_id | integer(10) | PK | No | Code of branch |
| Name | varchar(100) |  | No | Name of Branch |
| PhoneNumber | varchar(10) |  | No | Phone number of Branch |
| Address | varchar(100) |  | No | Address of Branch |
| Status | bit(0) |  | Yes | Status exist |

 Level\_Authority

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| LV\_Id | integer(10) | PK | No | Code of level\_authority |
| Name | varchar(50) |  | No | Name of Level\_Authority |
| Status | bit(0) |  | Yes | Status apply |

 Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| E\_Id | char(6) | PK | No | Code of employee |
| FirstName | varchar(20) |  | No | FirstName of Employee |
| LastName | varchar(20) |  | No | LastName of Employee |
| UserName | varchar(30) |  | No | UserName of Employee |
| Pass | varchar(30) |  | No | Pass of Employee |
| Qualification | varchar(30) |  | No | Qualification of Employee |
| Birthday | date(0) |  | No | Birthday of Employee |
| Location | integer(10) | FK ([Branch.B\_id](#n39EJdSAUFkyUwy8)) | No | Code of Branch |
| LV\_Id | integer(10) | FK ([Level\_Authority.LV\_Id](#TW7EJdSAUFkyUw50)) | No | Code of Level\_Authority |
| Status | bit(0) |  | Yes | Status account maintain |

 Ticket

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| T\_Id | integer(10) | PK | No | Code of ticket |
| Price | double(100) |  | No | Price of trip |
| Cus\_Id | integer(10) | FK ([Customer.Cus\_Id](#CJQJxdSAUFkyUxTs)) | No | Code of Customer |
| Startdate | date(0) |  | No | Start date of trip |
| Bus\_number | char(5) | FK ([Bus.Bus\_number](#Cn.EJdSAUFkyUwnC)) | No | Code of Bus |
| Rou\_Id | integer(10) | FK ([Route.Rou\_Id](#m8AoJdSAUFkyUwRX)) | No | Code of Route |
| SeatNumber | integer(10) |  | No | Seat Number of Bus |
| Return | bit(0) |  | Yes | Status ticket return |
| Status | bit(0) |  | Yes | Status ticket issue |

 TicketReturn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| TR\_Id | integer(10) | FK ([Ticket.T\_Id](#aqQQJdSAUFkyU.aH)); Unique | No | Code of ticket |
| ReturnDate | date(0) |  | No | Date ticket return of Customer |
| Diminish | integer(10) |  | No | Rate diminish of customer |
| ReturnPrice | double(100) |  | No | Return price give customer |

 New & Event

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| NE\_Id | integer(10) | PK | No | Code of new & event |
| Title | varchar(100) |  | No | Title of New & Event |
| Content | varchar(1000) |  | No | Content of New & Event |
| Author | varchar(100) |  | No | Author of New & Event |
| Date | date(0) |  | No | Date of New & Event |
| Status | bit(0) |  | Yes | Status new & event issue |

 Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date type | Constraints | Nullable | Documentation |
| A\_Id | integer(10) | PK | No | Code of admin |
| UserName | varchar(30) |  | No | UserName of Admin |
| Pass | varchar(30) |  | No | Pass of Admin |
| FullName | varchar(100) |  | No | FullName of Admin |
| Address | varchar(100) |  | No | Address of Admin |
| PhoneNumber | varchar(10) |  | No | PhoneNumber of Admin |

## Algorithms (optional)

<Provide the detailed description about algorithms used in the system. You can use Flow Chart or Activity Diagram to represent algorithms. Focus on the important and complex algorithms>

## Others (optional)

<Any design concerns or diagrams can be put here>

# System Prototype

<Put the system prototype or mock UI here. Focus on **important forms** and the **screen flows** between forms. If you use RAD and .NET, prototyping really help you to reach your development goals quickly>

# Management and Project Planning

## Management Approach

< Briefly describe the management approach that your team selects. Is your team self-managed or managed by one leader? How do you assign tasks to team members? How often do you meet? What do you do during meeting? Etc.>

## Project Plan

<The detailed project plan is put here. You can use WBS Excel sheet, Sprint Backlog (see Sprint Backlog\_Template.xls), Task sheet, Gantt chart, etc. to present your team’s plan. You can capture the Gantt chart in PMS if you use it to plan your project>

## Task Sheet

<Write down the tasks in Task Sheet-compatible format, this Task Sheet works as the activity report of the projec or the plan of the project (not recommended); see eProject Guide for detailed Task Sheet>

## Meeting Minutes (Optional)

<Put all minutes of your team meetings here>

# Checklists

## Check List of Validation

< Put the checklist here; describe how it is used and the resulted checklist>

## Submission Checklist

< Put the checklist here; describe how it is used and the resulted checklist>

# Screenshots

<Capture some intuitive and main screens of the software and put them here>

# Coding Convention

<Provide the coding convention for your team. If you simply want to use the existing code standard(s) such as ‘Java Code Convention’, you can refer to it\them by name or URL>

# Other Concerns<Optional>

<If you have any other information you want to add to this document, place it here. This could include thoughts on the eProject, improvements, etc.>

# Appendix

## Glossary [Optional]

<Place all definitions or abbreviation used in this document >

## References [Optional]

<Place all referenced materials used in this document >

## Others<Optional>