ISD TERM PROJECT

<Requirements Specification>

< Bus Ticketing System>



Project Advisor

**<Sajid Saleem>**

Presented by:

**Group ID: xxxxxxx**

Student Reg# Student Name

L1F19BSCS0390 Muhammad Ali Zain

L1S20BSCS0010 Umer Ahmed

L1S20BSCS0005 Mohsin Iqbal (E1)

**Faculty of Information Technology**

**University of Central Punjab**

Software Requirements Specification

Version <Version #>

< Bus Ticketing System >

Advisor: <Sajid Saleem>

Group <Group ID>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Member Name | Primary Responsibility | % Contribution | Roll No | Serial # | Sign |
| Ali Zain |  | 33.3 | L1F19BSCS0390 |  |  |
| Umer Ahmed |  | 33.3 | L1S20BSCS0010 |  |  |
| Mohsin Iqbal (E1) |  | 33.3 | L1S20BSCS0005 |  |  |
|  |  |  |  |  |  |

Table of Contents

Table of Contents iii

1. Introduction and Background 1

1.1 Product (Problem Statement) 1

1.2 Background 1

1.3 Scope 1

1.4 Objective(s)/Aim(s)/Target(s) 1

1.5 Challenges 1

1.6 Learning Outcomes 1

1.7 Nature of End Product 1

1.8 Completeness Criteria 1

1.9 Business Goals 1

1.10 Related Work/ Literature Survey/ Literature Review 1

1.11 Document Conventions 1

2. Overall Description 2

2.1 Product Features 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

~~~~~~~~~~ Phase II ~~~~~~~~~~ 3

3. Project Management. 3

3.1 Work Breakdown Structure (WBS). 3

3.2 Develop an Activity Graph. 3

3.3 Find the Critical Path. 3

3.4 Create a Gantt chart. 3

4. Functional Requirements 3

4.1 Name of Use-Case 1 3

4.2 Name of Use-Case 2 (and so on) 4

Requirements Analysis and Modeling 4

5. Nonfunctional Requirements\* 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Additional Software Quality Attributes 5

6. Other Requirements 5

~~~~~~~~~~ Phase III ~~~~~~~~~~ 6

7. Designing 6

7.1 Complete class diagram 6

7.2 Complete Data Flow Diagram (DFD) 6

7.3 Complete ER Diagram 6

7.4 Physical design of your database 6

7.5 Information on use of design patterns while designing the modules 6

7.6 Make a Sequence and a Collaboration diagram of following. 6

7.6.1 Ordering a New Raw Material Item 6

~~~~~~~~~~ Phase IV ~~~~~~~~~~ 6

8. Estimation 6

8.1 Cost Benefit Estimation 6

8.1.1 Return of Investment (ROI) {1st year} 6

8.1.2 %gain on ROI {1st year} 7

8.1.3 Payback Period in years 7

8.2 FP based Estimation 7

8.3 COCOMO Estimation 7

9. References 7

Appendix A: Glossary 8

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction and Background

Earlier to reserve a ticket people had to waste a lot of time by standing in a long queue. To book ticket instantly “Bus Ticketing System” came into existence. It is an application that allows users to check the bus ticket availability, book the bus tickets and payment transactions are done through online. It provides facilities like reservation; cancellation of seats and various types of information’s regarding buses etc. The QR code will be generated as ticket, this is scanned at the entrance and allowed inside the bus. User needs to login by providing necessary details. The user must enter the source and destination place and the number of passengers in the fields provided by the app and then the respective amount is been deducted directly through application wallet or bank account which is registered at the time of login and the ticket will be generated that will be unique. This will be scanned at entrance and valid passengers are allowed to get in.

The application consists of two accounts:

Admin account: Using Admin account, the admin can recharge the user balance and view all the processed tickets of the user.

User Account: User can easily check the availability of buses and can book a ticket even 10 days ago by using this app. When user log in to application they can search by adding their source and destination and can book their preferred one. User can easily book Ticket by filling a single form and paying amount by Application wallet, Bank Account or any online Banking app like jazz cash etc.

## Product (Problem Statement)

A problem statement always expresses the words that will be used to keep the effort on tracking

things out to represent a solvable problem with the existing system (Linus, 2014). Below are the

problems with the current system.

¬ Systems that are used by the staff at the counter currently are an internal system and

just used to sell the bus ticket at the counter. Customer has to go to the counter to buy

bus ticket or ask for bus schedule.

¬ Furthermore, customers need to pay cash when they buy the bus ticket and sometimes

needs to queue up long time to get the bus ticket. Besides that, customers are also not

allowed to buy bus ticket through telephone and the bus company’s telephones are

always busy line.

A problem statement always expresses the words that will be used to keep the effort on tracking

things out to represent a solvable problem with the existing system (Linus, 2014). Below are the

problems with the current system.

¬ Systems that are used by the staff at the counter currently are an internal system and

just used to sell the bus ticket at the counter. Customer has to go to the counter to buy

bus ticket or ask for bus schedule.

¬ Furthermore, customers need to pay cash when they buy the bus ticket and sometimes

needs to queue up long time to get the bus ticket. Besides that, customers are also not

allowed to buy bus ticket through telephone and the bus company’s telephones are

always busy line.

A problem statement always express to the world that there is a solvable problem with the existing system. So current system problems are written bellow.

* Systems are using by the staff right now are just use to sell tickets. Besides customers has to go to counter to buy the ticket or check the availability.
* Sometimes, customer needs to queue up long time to get a bus tickets.
* Counter time is limited to the office hours.

A problem statement always expresses the words that will be used to keep the effort on tracking

things out to represent a solvable problem with the existing system (Linus, 2014). Below are the

problems with the current system.

¬ Systems that are used by the staff at the counter currently are an internal system and

just used to sell the bus ticket at the counter. Customer has to go to the counter to buy

bus ticket or ask for bus schedule.

¬ Furthermore, customers need to pay cash when they buy the bus ticket and sometimes

needs to queue up long time to get the bus ticket. Besides that, customers are also not

allowed to buy bus ticket through telephone and the bus company’s telephones are

always busy line.

ISSN: 2289-7615 Page 36

¬ Operation times of the bus station to book ticket are limited to the office hours only.

¬ People need to go to the bus station or terminal to book the ticket.

A problem statement always expresses the words that will be used to keep the effort on tracking

things out to represent a solvable problem with the existing system (Linus, 2014). Below are the

problems with the current system.

¬ Systems that are used by the staff at the counter currently are an internal system and

just used to sell the bus ticket at the counter. Customer has to go to the counter to buy

bus ticket or ask for bus schedule.

¬ Furthermore, customers need to pay cash when they buy the bus ticket and sometimes

needs to queue up long time to get the bus ticket. Besides that, customers are also not

allowed to buy bus ticket through telephone and the bus company’s telephones are

always busy line.

ISSN: 2289-7615 Page 36

¬ Operation times of the bus station to book ticket are limited to the office hours only.

¬ People need to go to the bus station or terminal to book the ticket.

A problem statement always expresses the words that will be used to keep the effort on tracking

things out to represent a solvable problem with the existing system (Linus, 2014). Below are the

problems with the current system.

¬ Systems that are used by the staff at the counter currently are an internal system and

just used to sell the bus ticket at the counter. Customer has to go to the counter to buy

bus ticket or ask for bus schedule.

¬ Furthermore, customers need to pay cash when they buy the bus ticket and sometimes

needs to queue up long time to get the bus ticket. Besides that, customers are also not

allowed to buy bus ticket through telephone and the bus company’s telephones are

always busy line.

ISSN: 2289-7615 Page 36

¬ Operation times of the bus station to book ticket are limited to the office hours only.

¬ People need to go to the bus station or terminal to book the ticket.

To overcome those problems in the existing system, Bus Ticketing system allows Users to book their Bus tickets online. User can use app to book tickets and receive the online booking receipt.

This Bus Ticketing Android application includes all the details of the Buses in all cities and include all the network and areas which they will cover. After booking a ticket a unique bar code and a seat number will be assigned to you. By just scanning Barcode to the barcode reader at the entrance you can have your seat easily.

## Background

On this day with the use of technology always advancing in our daily life, people tend to use

technology more than the way they study in various aspects. Innovation is a fundamental piece

of our lives today and few can envision living without. We accomplished a great deal with the

assistance of innovation, for instance we have the likelihood to travel, stay in contact with

companions on the opposite side of the earth and cure numerous ailments. It implies more

flexibility and decisions for individuals however in the meantime we need to consider the social

lopsidedness, weapons of mass pulverization and regular asset consumption, according to Jane

Godall (2014). Furthermore we know that choices one individual takes are not really useful for

the general public overall. This infers to make innovation as positive as workable for

humankind and nature over the long haul. The innovation we make now makes the future and it

ought to be for the general population and make things simpler and not more entangled.

*Today, the technology always advancing in our daily life day by day, people tend to use technology. Innovation is a fundamental piece of our lives today and even some can’t consider living without it. As we have the likelihood to travel, stay in contact with companions or go for work daily. Every invention is for human benefits. So, we uses the technology to tend people life in easy way by making this app. This infers to make innovation as positive as workable for humankind and nature over the long haul. The innovation we make now makes the future and it ought to be for the general population and make things simpler.*

## Scope

*It is to help people in booking a bus ticket easily. It can make the highest scope in market where many users wants to know the availability or book bus tickets by just sitting home. It could be really useful for students those who don’t have any vehicles to reach their institute.*

## Objective(s)/Aim(s)/Target(s)

* To save the users effort and time.
* To Book a ticket easily.
* 24/7 service.
* To check the availability of their desired one.
* This system made user friendly so anyone can easily use.
* To make easy, reliable and corrective.
* Less time consuming as compared to manual work.

## Challenges

*Providing a solution that effectively caters to the user's need at the desired cost is another contest. Primary challenges include:*

* *Project Infrastructure*
* *Friendly user Interface*
* *Development Expectations and Outcome*
* *Quality Assurance*
* *Adapting the Latest Market Trends*
* *System & Application Integration*
* *More Reliable and easy to use*
* *Needs of the users can satisfy this software*
* *Security Infrastructure*

## Learning Outcomes

* *Project management, including scope, time, cost, and quality.*
* *Gathered detailed information on the models of the software for our better communication between the customer and user.*
* *Identifies, analysis, and responds to project risk, among other things.*
* *Project management methods are strategically applied in several organizational and international situations.*
* *Learned several language and Software Development kit.*

## Nature of End Product

*The End product is meant to be Bus Ticketing system. This will be an android as well as ISO and windows application which can be easily accessible on all types of devices.*

*It will be available for two types of users, Administrator and local user.*

## Completeness Criteria

*After completing the project, there will be a phase of testing and quality assurance. Soon after completion, the project will be online, where we will ensure its objectivity and requirement fulfillment. After the project is deployed, if there is no error on runtime, and it provides exactly the required outcomes. The project is considered to be complete.*

## Business Goals

It must help the people who don’t have vehicles. In the specific time period the applications account should be more than a specific numbers. There should be at least one admin in the range of 3km so user could easily recharge if he wants.

## Related Work/ Literature Survey/ Literature Review

*We arrange the conversation between students who don’t have any vehicles. We manage the meetings in our university A building lobby for the understanding of our project.*

* **Bookme.com**

It’s a web based application and android version app too. It’s also use to book buses or see the availability of buses for next coming days.

* **Swvl**

It’s also use to book local buses or see the availability of buses.

## Document Conventions

*Arial Font has been used throughout the document while being followed by Italics and bold (on some places) combination. The primary headings are Times font with 14 font-size where needed.*

# Overall Description

## Product Features

* *Buy ticket request.*
* *Ability to modify a reservation.*
* *Payment Records.*
* *When a passenger makes reservation, the seat will change color from blue to red for male and yellow for female.*
* *Has a default seat number.*
* *Dashboard for User and Admin.*
* *Responsive UI.*
* *User friendly.*
* *Reserve a seat even 10 days before.*
* *Check the availability any time.*

## User Classes and Characteristics

*There are 5 classes in Bus Ticketing System: User, Admin,* *Dashboard, Payments and Booking.*

* *Dashboard is the main class in our software as it shows options and takes details from the user and provide it to the admin.*
* *User is the class which will provide its personal information along with unique username and password.*
* *Admin will also be provided his own username and password and he will have access to the User’s information. So, different users will never be mixed in tables and they can be initialized by all these characteristics. He also can recharge the user app wallet.*
* *All User’s payments are stored in payments records and admin will access user’s payment records.*
* *This option known as booking help user to book tickets.*

## Operating Environment

*The software will operate with the following software components and applications:*

*An Admin Cell has been established under the supervision of administration section for addressing user’s Payments and Booking to redress them accordingly. The Admin Cell then forward the Booking to the specific department which will resolve the user’s query.*

*I will be efficient with Android, ISO and windows operating system environment.*

## Design and Implementation Constraints

* *The information of all Users, Admin and their details will be stored in database that will be accessible to application.*
* *MS SQL will be used as SQL engine and database.*
* *The Bus Ticketing System will be running 24/7.*
* *User can access from Computer or mobile device that has internet connection.*
* *User and admin must have to remember their username and password in order to sign in to their login account and do actions.*

## Assumptions and Dependencies

*The product needs the third-party products.*

*SQLite databases to store the information and details.*

*Flutter to develop the application.*

# ~~~~~~~~~~ Phase II ~~~~~~~~~~

# Project Management.

## Work Breakdown Structure (WBS).

## Develop an Activity GraphC:\Users\SH Ali Zain\Downloads\Untitled-Page-2.jpg

## Find the Critical Path.

## 

## 3.4 Create a Gantt chart.

## C:\Users\SH Ali Zain\Downloads\Bus Tickting System.png

# Functional Requirements

## Name of Use-Case 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identifier** | | | UC-1 | | | |
| **Purpose** | | | Purchase of ticket | | | |
| **Priority** | | | High | | | |
| **Pre-conditions** | | | User | | | |
| **Post-conditions** | | | Ticket details returned to user | | | |
| **Typical Course of Action** | | | | | | |
| **S#** | | **Actor Action** | | | **System Response** | |
| **1** | | Login | | | Email || Phone Number | |
| **2** | | Password | | | Password || Pin | |
| **3** | | Update | | | System || Status Update | |
| **4** | | Editing | | | Email, Password | |
| **5** | | User book a ticket by giving details of the place where user wants to go. | | | System checks details and then updates user record | |
| **6** | | Weekly, Monthly, Yearly Reports | | | Reports of Desired Time | |
| **7** | | Comparison between Reports | | | Difference (Profit or Loss) between two or more than two reports of tickets purchase | |
| **8.** | | Currency Exchange (no Currency Barrier) | | | Exchange of Currency | |
| **Alternate Course of Action** | | | | | | |
| **S#** | | **Actor Action** | | | **System Response** | |
| **1** | | Two steps Verification | | | Alert via message or Email | |
| **2** | Details doesn’t match ticket requirements | | | User give wrong details data violate requirements.  System prompts the portal to re-enter user details. | |
|  | |  | | |  | |
| **3** | | Database | | | Making user records and update. | |

Table : UC-1

## Name of Use-Case 2 (and so on)

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-2 | |
| **Purpose** | | Assign ticket to user | |
| **Priority** | | High | |
| **Pre-conditions** | | User has filled details | |
| **Post-conditions** | | ticket ID is returned to the user. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User book a ticket by giving details of the place where user wants to go. | | System checks details and then updates user record |
| **2** | Ticket ID is returned to the user. | | System assigns the ticket to that user and update the user details as well. |
| **3** | user can check ticket status. | | Message is sent to user’s account. |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If there are multiple users for that area. | | If there are multiple officers for that area.   * System assign the ticket to the user who appear first. |

**Name of Use-Case 3:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-3 | |
| **Purpose** | | User can check No of Ticket | |
| **Priority** | | Medium | |
| **Pre-conditions** | | User checks the status of his ticket. | |
| **Post-conditions** | | status of tickets | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Citizen enters the Ticket ID | | System searches for the ticket ID and displays the details of the ticket. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If ticket ID is not found | | If complaint ID is not found   * System informs the user and asks to re-enter the user ID. |

**Name of Use-Case 4:**

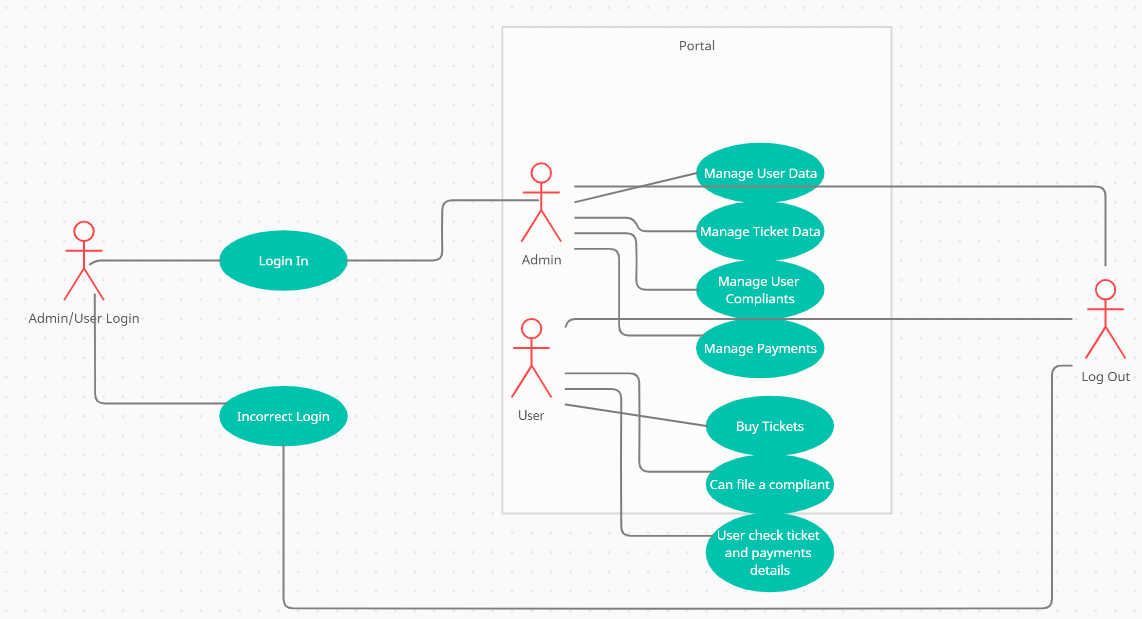
|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-4 | |
| **Purpose** | | user can check status of complaints | |
| **Priority** | | Medium | |
| **Pre-conditions** | | Admin has logged in | |
| **Post-conditions** | | Complaint status will be returned to user | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Admin updates the complaint status | | System updates the complaint and notifies the User. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | |  |

**Name of Use-Case 5:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-5 | |
| **Purpose** | | Ticket issue resolved | |
| **Priority** | | Medium | |
| **Pre-conditions** | | Admin has set the status of the complaint as resolved | |
| **Post-conditions** | | User get the notification of case resolved. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Admin updates the complaint status | | System updates the complaint and notifies the User. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** |  | |  |

## Requirements Analysis and Modeling

<Include the following analysis models: **use-case diagram**, entity-relationship diagram, **abstract class diagram**, sequence diagram (at least model interactions between system and external world). Additional diagrams may be added for example **state diagram**, data flow diagram (model interactions between system and external world), decision table, event table etc.>



# Nonfunctional Requirements\*

**\*make sure all your requirements functional or non functional are testable**

## Performance Requirements

**Response Time:** Application Loading, screen open and refresh times etc.

**Processing Time:** Functions, calculation

**Rationale:** Individual user performance is not a higher priority than overall system performance. Good design practice requires a fallback process in times of unexpected system load.

**Query and reporting Time:** Initial Loads and Subsequent Loads.

**Server response:** Server response time for rendering user interface should be not more than 1 second.

**IP Access:** Average user interface load time should be 3 seconds and Maximum user interface should be 6 seconds.

## Safety Requirements

1. Download App only from trusted Sources.

2. Maintain confidentiality of your wallet (Avoid sharing your account info with the third party).

3. Say No to Public Wi-Fi (Avoid Using public Wi-Fi as the hackers can snoop into the connection and steal your wallet).

## Security Requirements

**Network Integrity:**

1. Fingerprint the mobile device to enable cloning detection
2. Reject or block abnormal payment origination (i.e., behavior is inconsistent with prior use).

**Data Integrity:**

1. The system must meet PA-DSS security standards.

2. The system must ensure only authorized users can create/read/update/delete protected data. Rationale: Data integrity cannot be achieved if anyone can alter the information.

3. The system must ensure minimum required data elements are provided before creating a record.

**Authentication:**

1. The system must allow every individual user to have a unique user account with independent authentication credentials.

2. The system may support complex passwords consisting of at least 20 printable characters, including combinations of numbers, letters, symbols, and punctuation

3. When a new account is created, the system shall automatically generate a random, unique password that meets administrator-defined complexity requirements.

4. The system shall require that the user change the password on first login after authenticating with the automatically generated initial password.

## Additional Software Quality Attributes

1. **Adaptability**

The adaptability of the app should be good means the app should be good enough to adapt the circumstance of any platform it can be either IOS, Android.

1. **Availability**

App should be available Globally the user can interact with the app any time any place.

1. **Reliability**

The app should be failure-free, the app should perform it functions under static conditions for specific period, it should full-fill it assigned tasks.

# Other Requirements

**External Interface requirements:**

**User Interface:**

* The design or layout of every form will be very clear and very interactive to the user.
* When the user opens the software, the welcome window will appear.
* In the login window the user can easily entered the desired password and login name.
* Then it will give the successfully login message.
* The overall performance of the software will reliable and enable the users to work efficiently.

**Security Requirements:**

* There will be proper security regarding to the accessing of data.
* The external security can be provided by given the login authentication.
* The data that are stored in the database must be private.
* There is also required a user authentication.

**Availability:**

100% of the time available server and enough capacity of database should be needed to this Bus Ticketing System.

**Reliability and Performance:**

Less failure of the system, readily available functionality and allow concurrent users.

**Real Time Feedback:**

The new Bus Ticketing system should display the daily music and should also show the changes in real time as the new entry is added or removed.

**Graphic user Interface (GUI):**

The GUI is generated in Java. Therefore, the Java package Java. Swing was most used. The Java enables to create the window design as shown as shown in the User Manual. A further explanation of the GUI is illustrated in the User Manual and the Test Scenarios.

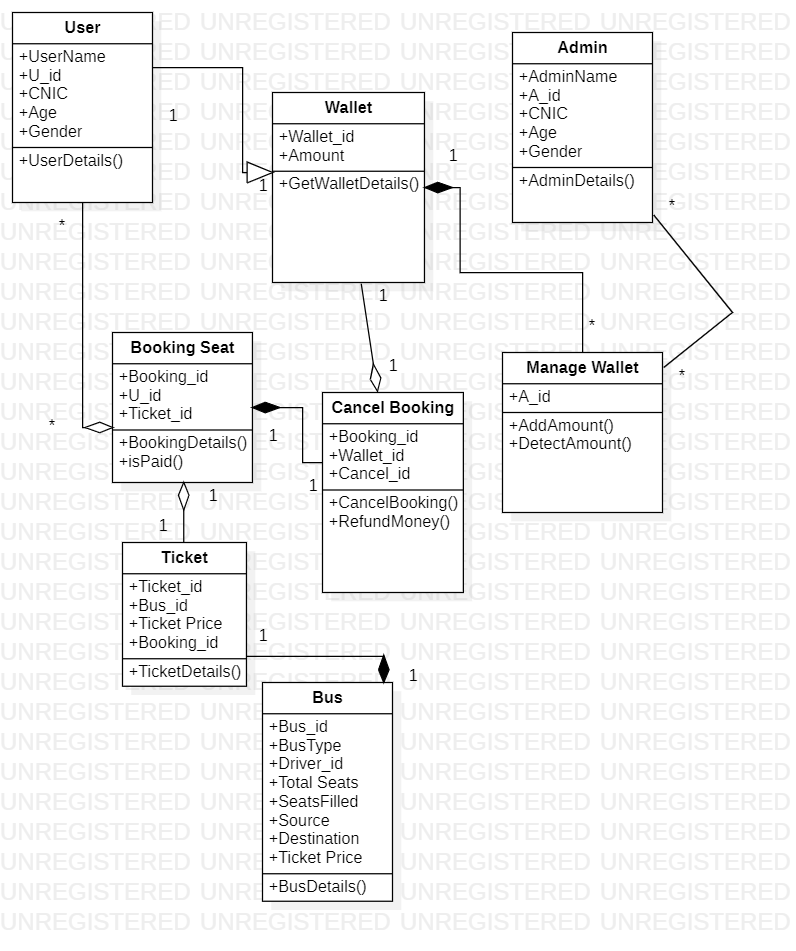
# ~~~~~~~~~~ Phase III ~~~~~~~~~~

# Designing

## Complete class diagram

*<including class attributes and methods (develop wrt Arch Style of PD2)>*

## 



## Complete Data Flow Diagram (DFD)

*< Context level DFD, level 0 DFD and level 1 DFD are required >*

***Context Level:***

## *C:\Users\SH Ali Zain\Downloads\Blank diagram.png*

***Level 0:***

## *C:\Users\SH Ali Zain\Downloads\Blank diagram - Page 2.png*

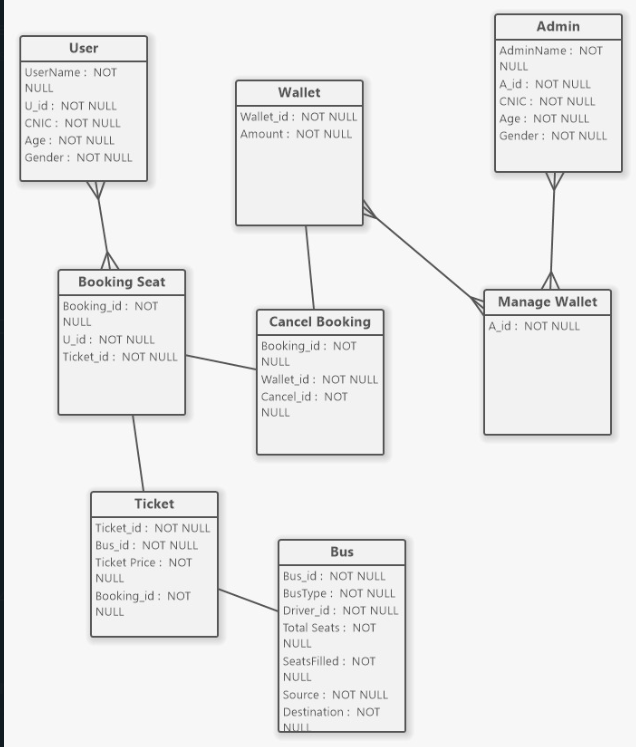
## Complete ER Diagram

*< Complete ERD is required >*

## C:\Users\SH Ali Zain\Downloads\image.png

## Physical design of your database

<i.e. table names with attributes and their types>



|  |  |
| --- | --- |
| **User** | |
| **U\_id** | **Int (Primary key)** |
| UserName | Varchar(100) |
| CNIC | Varchar(100) |
| Age | Int |
| Gender | Varchar(1) |

|  |  |
| --- | --- |
| **Wallet** | |
| **Wallet\_id** | **Int (Primary key)** |
| Amount | Int |

|  |  |
| --- | --- |
| **Booking Seat** | |
| **Booking\_id** | **Int (Primary key)** |
| Ticket\_id | Int (Foreign key) |
| U\_id | Int (Foreign key) |

|  |  |
| --- | --- |
| **Admin** | |
| **A\_id** | **Int (Primary key)** |
| CNIC | Varchar(20) |
| Age | Int |
| Gender | Varchar(10) |
| AdminName | Varchar(20) |

|  |  |
| --- | --- |
| **Cancel Booking** | |
| **Cancel\_id** | **Int (Primary key)** |
| Booking\_id | Int (Foreign key) |
| Wallet\_id | Int (Foreign key) |

|  |  |
| --- | --- |
| **Manage Wallet** | |
| A\_id | Int (Foreign key) |

|  |  |
| --- | --- |
| **Ticket** | |
| **Ticket\_id** | **Int (Primary key)** |
| Ticket Price | Int |
| Bus\_id | Int (Foreign key) |
| Booking\_id | Int (Foreign key) |

|  |  |
| --- | --- |
| **Bus** | |
| **Bus\_id** | **Int (Primary Key)** |
| BusType | Varchar (50) |
| Driver\_id | Int |
| Total Seats | Int |
| SeatsFilled | Varchar (10) |
| Source | Varchar (20) |
| Destination | Varchar (20) |

## Information on use of design patterns while designing the modules

User - Wallet, Admin – Manage Wallet, User – Booking Seat, Booking Seat – Ticket, Ticket – Bus, Booking Seat – Cancel Booking.

**USER CLASS:**

The user class checks and uses different classes. It checks the records through booking seat class also it can check wallet class records.it can also access cancel booking class with the help of booking seat class.it can also access bus class and ticket class.it can also manage wallet by accessing manage wallet class by surpassing the wallet class.

**ADMIN CLASS:**

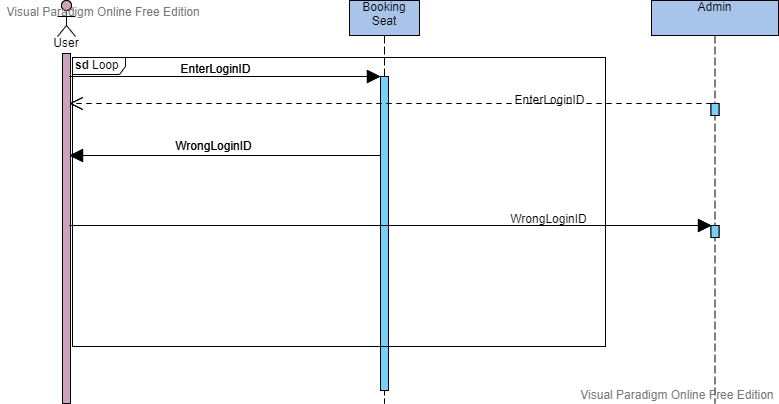
The admin class will coordinate with the main class which is user class to check all records and maintain integrity. It will not communicate with the user class directly but when the user will book a ticket of bus admin have access to check details, the admin will check user wallet class also.

The rest of the classes are not prominent classes. They are used by admin and user class to fulfill the definition and purpose of the software following are the classes

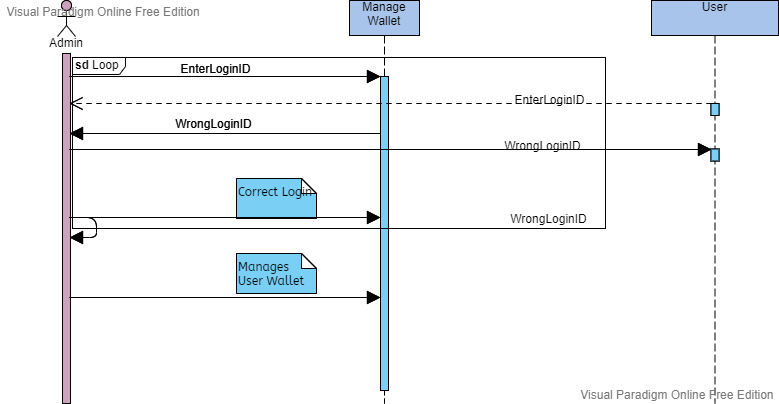
* WALLET CLASS
* CANCEL BOOKING CLASS
* MANAGE WALLET CLASS
* TICKET CLASS
* BUS CLASS
* BOOKING SEAT CLASS

## Make a Sequence and a Collaboration diagram of following.

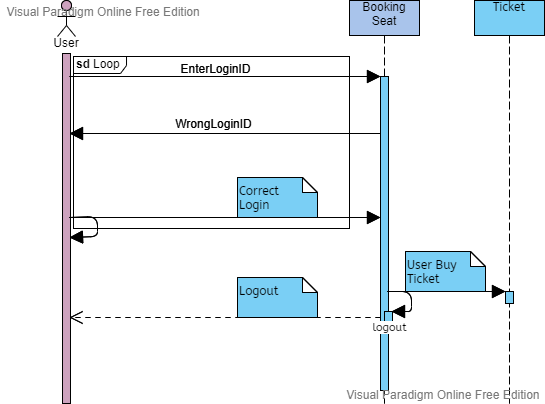
### Login



### Admin Login and Manages User Wallet



### User Login and Buy Ticket



# ~~~~~~~~~~ Phase IV ~~~~~~~~~~

# Estimation

## Cost Benefit Estimation

### Return of Investment (ROI) {1st year}

<Approximate estimations should be done after approximate estimation of values. Formula of ROI = Benefits / Cost >

### %gain on ROI {1st year}

< Values should be based on estimates of 8.1.1. Formula of %gain on ROI = (Benefits - Cost) / Cost x 100>

### Payback Period in years

<Values should be based on estimates of 8.1.1. Formula of payback period = Cost / Benefit >

## FP based Estimation

<Approximate estimation should be done using FP >

1. Does the system require reliable backup and recovery?

Answer: 1

2. Are specialized data communications required to transfer information to or from the application?

Answer: 3

3. Are there distributed processing functions?

Answer: 5

4. Is performance critical?

Answer: 1

5. Will the system run in an existing, heavily utilized operational environment?

Answer: 1

6. Does the system require online data entry?

Answer: 1

7. Does the online data entry require the input transaction to be built over multiple screens or operations?

Answer: 2

8. Are the ILFs updated online?

Answer: 3

9. Are the inputs, outputs, files, or inquiries complex?

Answer:5

10. Is the internal processing complex?

Answer: 3

11. Is the code designed to be reusable?

Answer: 1

12. Are conversion and installation included in the design?

Answer: 2

13. Is the system designed for multiple installations in different organizations?

Answer: 1

14. Is the application designed to facilitate change and ease of use by the user?

Answer: 1

∑(fi) = 1 +3 +5+1+1+1+2+3+5+3+1+2+1+1 = 30

|  |  |  |  |
| --- | --- | --- | --- |
| Information Domain Values | Count | Complex | C\*C |
| External Inputs(EIs) | 9 | 6 | 54 |
| External Outputs (EOs) | 3 | 7 | 21 |
| External Inquiries(EQs) | 0 | 6 | 0 |
| Internal Logical Files (ILFs) | 0 | 15 | 0 |
| External Interface Files(EIFs) | 0 | 10 | 0 |
| Total Count: |  |  | 75 |

FP = Total Count \* [0.65 + (0.01 \* ∑(fi))]

FP = 75 \* [0.65 + (0.01 \* 30)]

FP = 75 \* [0.65 +0.3]

FP = 75 \* 0.95

FP = 71.25

## COCOMO Estimation

<Approximate estimation should be done using FP based COCOMO >

# References

# Phase1:

[1] “Bus Ticket Booking System Project - Download Project Source Code and Database | FreeProjectz.” https://www.freeprojectz.com/project-source-code-database-download/bus-ticket-booking-system-project (accessed Apr. 15, 2022).

# Phase2:

[2] “Untitled Workspace - Creately.” https://app.creately.com/d/nmqUCN5WIgP/edit (accessed May 02, 2022).

[3]“Untitled | Visual Paradigm Online.” https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=ActivityDiagram&width=11&height=8.5&unit=inch (accessed May 02, 2022).

[4] “GanttPRO.” https://app.ganttpro.com/#/project/1651508340979/export (accessed May 02, 2022).

**Phase3:**

[5] “Blank diagram: Lucidchart.” https://lucid.app/lucidchart/1b698e03-0733-4a64-a4ea-0fe1b74559a6/edit?beaconFlowId=901C7C093BC77479&invitationId=inv\_68b5c98e-ad84-49c8-9933-624e732102e4&page=0\_0# (accessed May 29, 2022).

[6] “UML Class Diagram (1) -- SmartDraw.” https://cloud.smartdraw.com/editor.aspx?templateId=fb001e47-7c0b-4c7e-84ef-173278eb2298&flags=128#depoId=34868370&credID=-38740264 (accessed May 29, 2022).

[7] “Online UML diagram tool | Lucidchart.” https://www.lucidchart.com/pages/landing/uml-diagram-software?utm\_source=google&utm\_medium=cpc&utm\_campaign=\_chart\_en\_tier3\_desktop\_search\_nb\_bmm\_&km\_CPC\_CampaignId=2083917352&km\_CPC\_AdGroupID=82392367408&km\_CPC\_Keyword=%2Bsequence%20%2Bdiagram%20%2Bmaker&km\_CPC\_MatchType=b&km\_CPC\_ExtensionID=&km\_CPC\_Network=g&km\_CPC\_AdPosition=&km\_CPC\_Creative=436416066130&km\_CPC\_TargetID=aud-536921399221:kwd-44587777232&km\_CPC\_Country=1011082&km\_CPC\_Device=c&km\_CPC\_placement=&km\_CPC\_target=&gclid=CjwKCAjwkMeUBhBuEiwA4hpqEESrRzj0vVkqXHhqT4a6KDQGifB2Ncpyg1rnBaQS7gdbmAWXPRrtpRoCy1sQAvD\_BwE (accessed May 29, 2022).

[8] “Diagram CASE Tool for Software Modeling & Analysis - UML, BPMN, ERD.” https://www.softwareideas.net/ (accessed May 29, 2022).

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>