

# **EXPERIMENT-1**

## **AIM:**

Install UML Tool and explore its menus and options.

## **Requirements:-**

Make sure that your system should installed with StarUML software.

## **Theory:**

Star UML contains many windows. Some of them are main window, menu, tool bars, windows, dialog box, quick dialog.

## **Main Menu:**

The main menu is one of the top of the screen most of star uml functions are accessible through the main menu. This section describes in detail all menu items included in staruml main menu.

1.File Menu 2.Edit Menu 3. Format Menu 4.Model Menu

5.View Menu 6.Tools Menu 7.Help Menu 8.Shortcuts.

StarUML contains these models:-

1.Use case Diagrams 2. Class Diagrams 3. Object Diagrams

4.ComponentDiagrams 5.Deployment Diagrams 6.Activity Diagrams

7.State chart Diagrams 8. Collaboration Diagrams 9. Sequence Diagrams

## **1.Use Case Diagrams:-**

To create a use case diagram

1.Select first an element where a new use case diagram to be contained a child.

2.Select model add diagram/use case diagram in menu bar or selected add diagram use case diagram in content menu.

We will use following elements in the use case diagram.

1.Package 2. Use case Subject 3. Actor 4. Use case 5.Extension Point

6.Association 7.Dependancy 8. Generation 9.Include 10.Extend

## **Use case Subject:**

To create an use case subject:

1. Select use case subject from tool box.
2. Drag on the diagram as the size of the use case subject.

### **ACTORS:**

To create an actor:

1. Select actor in tool box.
2. Drag on the diagram as the size of the actor. To create an actor (model element only) by menu.

1. Select an element where a new actor to be contained
2. select model/add/ actor in menu bar or add / actor in context menu

### **Extension point:-**

To add an Extension point

1. Select an use case
2. Select model / add/extension point in menu bar (or) add/extension point in context menu.

To edit an extension point you can do following

### **Actions:-**

Use Quick an extension point by double click or press enter a selected extension point.

### **Include:-**

To create an include.

- 1 Select include in tool box
- 2 Drag from a case and drop on another use case.

### **Extend:-**

To create an extend

- 1 select extend in Tool box
- 2 Drag from a usecase and drop on another usecase

## **Sequence Diagram:-**

To create sequence diagram.

1. Select an element where a new sequence diagram to be contained as a child.
2. Select model/add diagram / sequence diagram in menu bar (or) select add diagram / sequence diagram in context menu.

In sequence diagram we can use following elements

1. Lifeline
2. Message
3. End point
4. Gate
5. State invariant
6. Continuation
7. Combined fragment
8. Interaction use

## **Deployment Diagram:-**

1. Component
2. Artifact
3. Interface
4. Dependency
5. Interface realization
6. Artifact instance
7. Component instance
8. Component realization
9. Node
10. Deployment
11. Communication path
12. Object
13. Node instance
14. Link instance

## **Class Diagram:-**

1.Class 2 .Attributes 3. Operation 4.Parameter5.Templateparameter  
6.Interface 7.Signal 8.Data type 9.package 10.Model System

## **Object diagram**

1. Object
2. Slot
3. Artifact
4. Component instance
5. Node instance
6. Link

## **Component Diagram:-**

1. State
2. Region
3. Component
4. Deployment
5. Artifact
6. Interface
7. Link
8. Object
9. Node instance
- 10.Interface realization
- 11.Node

## **State chart Diagram**

1. Internal activity
2. State
3. Initial state
4. Region
5. Choice
6. Final state
7. Shallow history
8. Join
9. Deep history
- 10.Fork
- 11.Function
- 12.Entry point
- 13.Exit point

- 14. Terminate
- 15. Connection point reference
- 16. Transition

## **Activity Diagram**

- 1. Action
- 2. Joint
- 3. Event
- 4. Activity final
- 5. Initial
- 6. Fork
- 7. Input pin
- 8. Send signal
- 9. Accept signal
- 10. Object node
- 11. Flow signal
- 12. Object flow
- 13. Control flow
- 14. Merge
- 15. Decision
- 16. Swimlane
- 17. i/p pin
- 18. o/p pin

## **Experiment - 2 (PROBLEM STATEMENT)**

**Aim :-** To identify a problem and create a problem statement for Railway Reservation System

### **Requirements:-**

Clear idea on the project should be required to identify the problem.

### **Theory:-**

### **Introduction:-**

#### **Emergence of Railways:-**

The Indian Railways(IR) is one of the largest railway networks around the world. It was established in the year 1853.Its headquarters situated at New Delhi. It serves the highest of number of people in India. Indian Railways is the backbone of India nation in the way of both social and economical. The services of IR are reliable, safe, secure and fast. It is one of the best and cheapest travelling resources in India.

#### **Problems without Reservation in Railways:-**

As a millions of people in India who are travelling across the nation for their needs like employees who work in cities to go office in time and it is difficult for old people who want to go pilgrimages. It requires more time to book tickets offline, standing in queues and wastage of money if they accidentally miss their journey and there is a big problem incase of emergency. To Overcome all these flaws Railway Reservation System technology comes into existence.

#### **Advantages of having Railways and Reservation:-**

Railway Reservation System Technology provides the facility to book rail tickets online and also provides details on a variety of schemes and special offers. This system made the life much easier for people because they can sit at their living room or office and book tickets in no time. Booking tickets in online is easy and simple process. It provides various facilities and opportunities for the travellers. In this we will store each and every person record and status of the person to track.we will use PNR(Passenger Name Record) for this.

- 1.You can shift larger and heavier luggage in train very comfortably.
- 2.Seat system is very comfortable. In AC coaches they will provide blankets and pillows also.
- 3.Better experience for families, aged persons, kids, physically challenged persons etc.

- 4.By this, we can decrease the unemployment in India. More than 15 lakh people are working in Indian Railways.
- 5.Concession in ticket booking for senior citizens, handicapped persons, and ladies etc.
- 6.Insurance of passengers in case of accidents for injures or death.
7. Online ticketing changed the face of indian railways ticketing system. You can book the ticket from your PC, mobile phone and tablet.
- 8.Tatkal scheme is one of the fastest rail tickets booking system. Booking starts one day prior to the departure of the train from source station.
- 9.Catering food in railways is also one of the major advantage.
- 10.You can dial to 139 and get all information from the Indian Railways.

### **Conclusion:-**

So, for this we can say that Indian Railways is more reliable in transport system especially in India. For every organisation there is a lot of pros and cons. But there is lot of improvements in Indian Railways. It serves every people.

## **EXPERIMENT - 3 (SRS)**

## **AIM:**

To prepare document of software requirements specification. Designing SRS document for Railway Reservation System.

## **Requirements:**

To prepare SRS document we need to have good knowledge on the project and also editor tools like MS Word required to prepare it.

## **Theory:**

# **Software Requirements Specification**



## **Railway Reservation System**

31-01-2019



## **Table of Contents:-**

### **1.Introduction**

- 1.1 Purpose
- 1.2 Intended Audience
- 1.3 Product Scope
- 1.4 Glossary

### **2.Overall Description**

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Assumption and Dependencies

### **3.Requirement Specification**

- 3.1 Software Specification
- 3.2 Hardware Specification

### **4.System Features**

- 4.1 Hardware Features
- 4.2 Software Features

### **5.Testing Requirements**

- 5.1 Integration Testing
- 5.2 Performance and Load Testing
- 5.3 Security Testing

### **6.References**

## 1.Introduction:-

The introduction of Software Requirement Specification provides an overview of the entire SRS purpose, scope, definitions, acronyms, abbreviations, references and overview of SRS.

It is a complete description of the behaviour of a system to be developed. It includes Functional Requirements and Non-Functional Requirements.

### 1.1 Purpose

The purpose of the SRS document is to provide a detailed **overview** of our software product, its **parameters** and goals. This document describes the project's target **audience** and its user **interface**, hardware and software requirements. It defines how our client, team and audience see the product and its functionality.

### 1.2 Intended Audience

This document is intended for every aspect of people like employees who work in cities to go office intime and it is difficult for old people who want to go pilgrimages. People easily book tickets and travel around the world.

### 1.3 Product Scope

“Railway Reservation System” is an attempt to simulate the basic concepts of an online Reservation System. The System enables to perform the following actions:

- Search for a Train
- Booking of a Selected Flight
- Payment
- Cancellation
- Freight Revenue enhancement
- Passenger Revenue enhancement
- Improved & optimized service
- PNR Status which is used to know live prediction of waitlisted train tickets.
- Flexible Date/Time to attract the users.

- Confirmation message or call for their reservation.

## 1.4 Glossary

This should define all technical terms and abbreviations used in this document.

- ❖ PRS :- passenger reservation system
- ❖ NTES:- National Train Enquiry System
- ❖ IVRS :- Interactive Voice Response system
- ❖ SRS :- Software Requirements Specification
- ❖ STD :- State Transition Diagram
- ❖ PNR :- Passenger Name Record
- ★ Freight :- Transportation cost
- ★ Hamper :- Impede the movement or progress of
- ★ Erroneous :- incorrect
- ★ Sluggish :- slow moving or inactive

## 2. Overall Description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the Railway Reservation System. It further lists and briefly describes the major features and a brief description of each of the proposed system.

### 2.1 Product Perspective

Before the automation, the system suffered from the following **DRAWBACKS:-**

- The existing system is highly manual involving a lot of Paper work and calculation and therefore may be erroneous. This has lead to inconsistency and inaccuracy in the maintenance of data.

- The data, which is stored on the paper only, may be lost, stolen or destroyed due to natural calamity like fire and water.
- The existing system is sluggish and consumes a lot of time causing inconvenience to customers and the airlines staff.
- Due to manual nature, it is difficult to update, delete, add or view the data.
- Since the number of passengers have drastically increased therefore maintaining and retrieving detailed record of passenger is extremely difficult.
- An railways has many offices around the world, an absence of a link between these offices lead to lack of coordination and communication.

Hence the railways reservation system is proposed with the following:-

- The computerization of the reservation system will reduce a lot of paperwork and hence the load on the airline administrative staff.
- The machine performs all calculations. Hence chances of error are nil.
- The passenger, reservation, cancellation list can easily be retrieved and any required addition, deletion or updation can be performed.
- The system provides for user-ID validation, hence unauthorized access is prevented.

## **2.2 Product Functions**

Booking agents with varying levels of familiarity with computers will mostly use this system. With this in mind, an important feature of this software is that it be relatively simple to use. The scope of this project encompasses:-

### **A . Search :**

This function allows the booking agent to search for train that are available between the two travel cities, namely the "Departure city" and "Arrival city" as desired by the traveller. The system initially prompts the agent for the departure and arrival city, the date of departure, preferred time slot and the number of passengers. It then displays a list of train available with different airlines between the designated cities on the specified date and time.

**B.Selection:**

This function allows a particular train to be selected from the displayed list. All the details of the train are shown :-

1. train Number
2. Date, time and place of departure
3. Date, time and place of arrival
4. TRAIN Duration
5. Fare per head
6. Number of stoppages – 0, 1, 2...

**C. Review:**

If the seats are available, then the software prompts for the booking of train. The train information is shown. The total fare including taxes is shown and flight details are reviewed.

**D. Traveller Information:**

It asks for the details of all the passengers supposed to travel including name, address, telephone number and e-mail id.

**E. Payment :**

It asks the agent to enter the various credit card details of the person making the reservation.

1. Credit card type
2. Credit card number
3. CVC(Card Verification Code)number of the card
4. Expiration date of the card
5. The name on the card

**F. Cancellation:**

The system also allows the passenger to cancel an existing reservation. This function registers the information regarding a passenger who has requested for a cancellation of his/her ticket. It includes entries pertaining to the train No., Confirmation No., Name, Date of Journey, Fare deducted.

## 2.3 User Classes and Characteristics

### A. Educational Level:-

At least user of the system should be comfortable with English language.

### B. TECHNICAL EXPERTISE:-

User should be comfortable using general purpose applications on the computer system.

## 2.4 Assumption and Dependencies

- Booking Agents will be having a valid user name and password to access the software.
- The software needs booking agent to have complete knowledge of railways reservation system.
- Software is dependent on access to internet.

## 3. Requirement Specification

All the requirements like Software and Hardware are explained here.

### 3.1 Functional Requirements:-

The functional requirements are those requirements which deal with what the system should do or services that provide for users.

- Describes the behaviour of the system as it relates to the system's functionality.
- Includes the description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.
- Specified by users themselves.

#### 3.1.1 Performance Requirements

1. **User Satisfaction:-** The system is such that it stands up to the user expectations.

- 2. Response Time:-** The response of all the operation is good. This has been made possible by careful programming
- 3. Error Handling:-** Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- 4. Safety and Robustness:-** The system is able to avoid or tackle disastrous action. In other words, it should be fool proof. The system safeguards against undesired events, without human intervention.
- 5. Portable:-** The software should not be architecture specific. It should be easily transferable to other platforms if needed.
- 6. User friendliness:-** The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

### **3.1.2 Non-Functional Requirements:-**

Non-functional requirements are those requirements which elaborate the performance characteristic of the system and define the constraints on how the system will do so.

- Defines the constraints, targets or control mechanisms for the new system.
- Describes how, how well or to what standard a function should be provided.
- Specified by technical peoples e.g. Architect, Technical leaders and software developers.
- They are sometimes defined in terms of metrics (something that can be measured about the system) to make them more tangible.
- Identify realistic, measurable target values for each service level.
- These include reliability, performance, service availability, responsiveness, throughput and security.

### **Reliability:-**

The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also the system will be functioning inside a container. Thus the

overall stability of the system depends on the stability of container and its underlying operating system.

#### **Availability:-**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. A customer friendly system which is in access of people around the world should work 24 hours. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the Organizer. Then the service will be restarted. It means 24 x 7 availability.

#### **Maintainability:-**

A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the project will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.

#### **Supportability:-**

The code and supporting modules of the system will be well documented and easy to understand. Online User Documentation and Help System Requirements.

#### **Security:-**

The system use SSL (secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.

## **4.System Features**

### **Hardware Requirements**



It deals with the hardware requirements like Ram size, Cache size, Processor type, Internal Memory size etc.. required for the software to run efficiently.

- Intel core I3 processor
- 250GB of Internal Memory
- Minimum 2GB of Ram
- Cache 2MB

### **Software Requirements**

Any Operating System like windows(any version like 7,8.1,10),Linux(Fedora, Zorin, Ubuntu) are primary requirements for Software Development.

The Systems should have Internet connection is secondary requirement.

The Softwares should have security, portability, Efficiency and Flexibility.

## **5.Testing Requirements**

In order to publish the project, software testing is one of the main aspects that a project should undergo. It should be comprehensive and should be carried out at each stage of development as well as implementation.

It includes integration testing, performance and load testing, security testing(include penetration and vulnerability testing) etc..

### **Integration Testing:-**

Integration of different modules is undertaken once they have been coded and unit tested.

During the integration and system testing phase, modules are integrated in a planned manner.


### **Performance and Load Testing:-**

Railways are the leading transportation in India. So, we expect high traffic. So, we need to maintain web server fields. Because if we don't maintain customers may face severe problems for their journey.


### **Security Testing:-**


- If the users want to do registration they will provide their personal details like aadhaar, phone number, and mail address.
- In the reservation process users may provide their bank account details to pay the money.
- So, we need to secure their information by providing some tools in our website.

## Registration Page



31-Jan-2019 [18:40:20] A+ A- A\*
CONTACT US
AGENT LOGIN
ALERTS
हिंदी


IRCTC Exclusive
TRAINS
HOLIDAYS
FLIGHTS
LOYALTY
MEALS
PROMOTIONS
SERVICE AT STATIONS
IRCTC eWallet
MORE



### Individual Registration

**GARBAGE/JUNK VALUES IN PROFILE MAY LEAD TO DEACTIVATION**  
Please use a valid E-Mail ID and mobile number in registration.

User Name \* :

Password \* :

Confirm Password \* :

Security Question \* :

Select Security Question ▼

Security Answer \* :



Preferred Language \* :

Select Preferred Language ▼

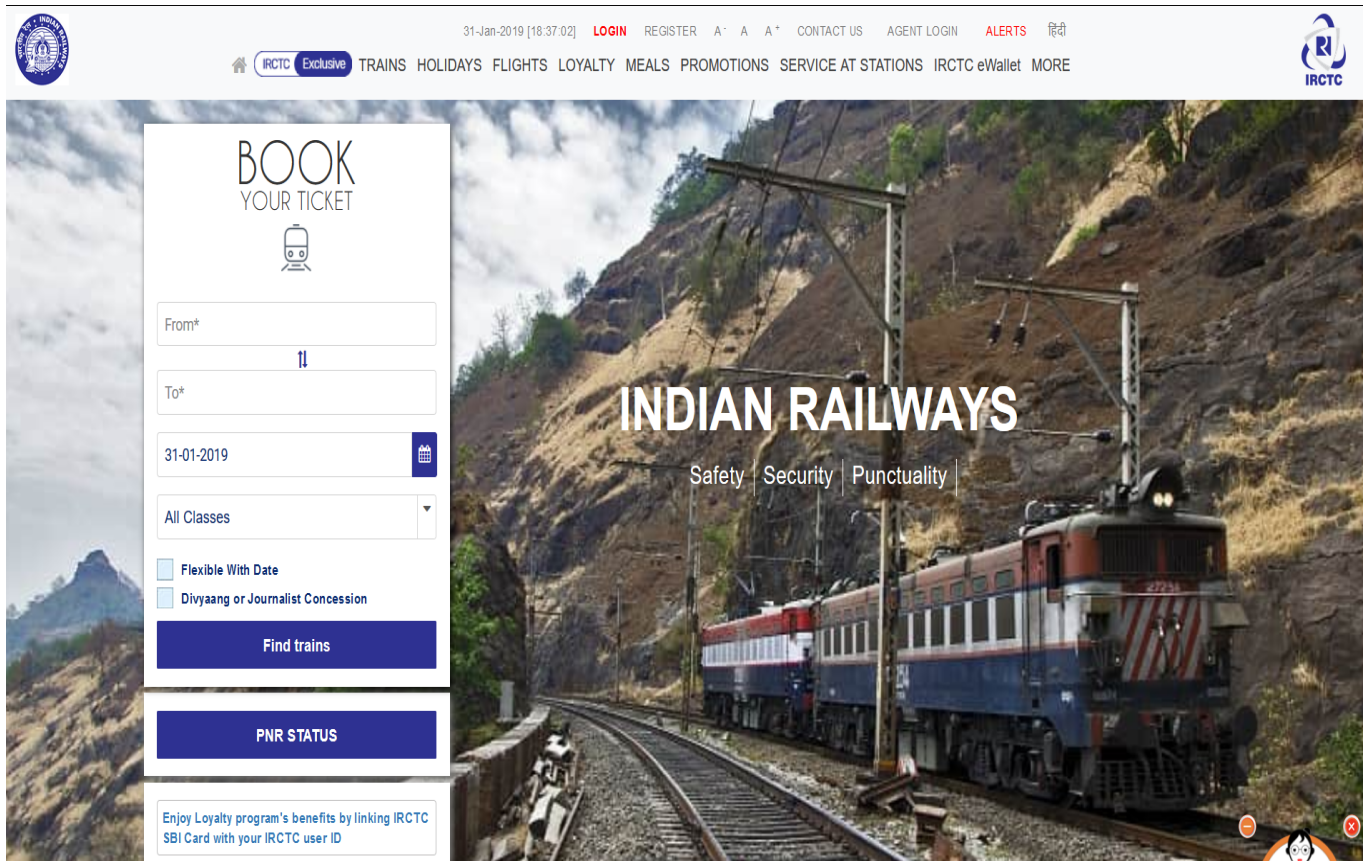
Personal Details

Name \* :

**Info!**Please provide your exact name as per Aadhaar to avail Aadhaar based benefits on IRCTC eTicketing website.

## Indian Railways official website



The screenshot displays the Indian Railways official website (IRCTC). The header includes the IRCTC logo, a date and time stamp (31-Jan-2019 [18:37:02]), and links for LOGIN, REGISTER, A+, A, A\*, CONTACT US, AGENT LOGIN, ALERTS, and हिंदी. Below the header, there is a navigation bar with links for IRCTC, Exclusive, TRAINS, HOLIDAYS, FLIGHTS, LOYALTY, MEALS, PROMOTIONS, SERVICE AT STATIONS, IRCTC eWallet, and MORE. The main content area features a large background image of a train traveling through a mountainous landscape. Overlaid on this image is the text "INDIAN RAILWAYS" in large white letters, with "Safety | Security | Punctuality" written below it. On the left side, there is a white box containing the "BOOK YOUR TICKET" section. This section includes a train icon, input fields for "From\*", "To\*", and "31-01-2019", a dropdown menu for "All Classes", and checkboxes for "Flexible With Date" and "Divyaang or Journalist Concession". Below these fields are buttons for "Find trains" and "PNR STATUS". At the bottom of the white box, there is a small text box that reads "Enjoy Loyalty program's benefits by linking IRCTC SBI Card with your IRCTC user ID".

## 6.References

[www.google.com](http://www.google.com)

[www.irctc.co.in](http://www.irctc.co.in)

[www.indianrail.co.in](http://www.indianrail.co.in)

### **Conclusion:-**

SRS document for Railway Reservation System is created successfully.

## **EXPERIMENT-4(USE CASE DIAGRAM)**

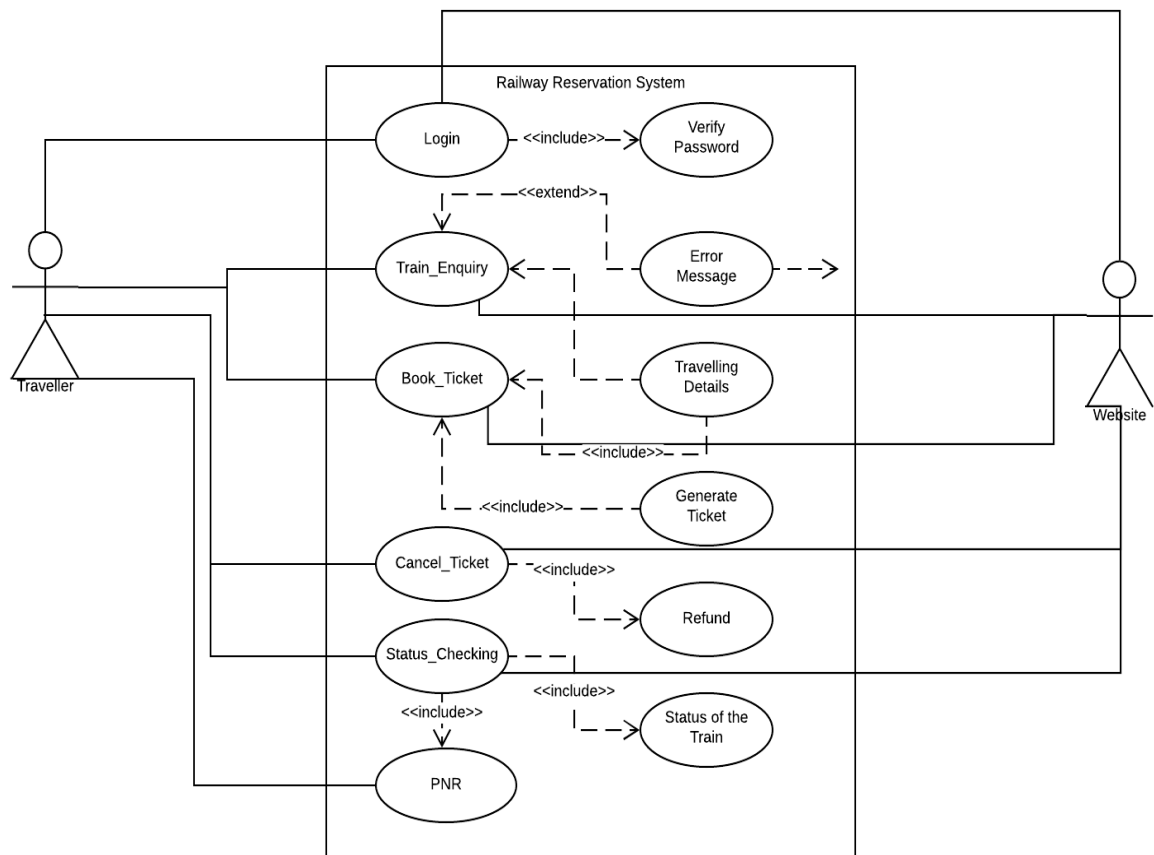
### **AIM:**

To draw a USE case diagram for the Railway Reservation System.

### **Requirements:**

StarUML application is required and also knowledge on the project is must to prepare sequence diagram.

### **Theory:**



### **Conclusion:**

Use case diagram for Railway Reservation System is drawn using StarUML.

## EXPERIMENT - 5 (ACTIVITY DIAGRAM)

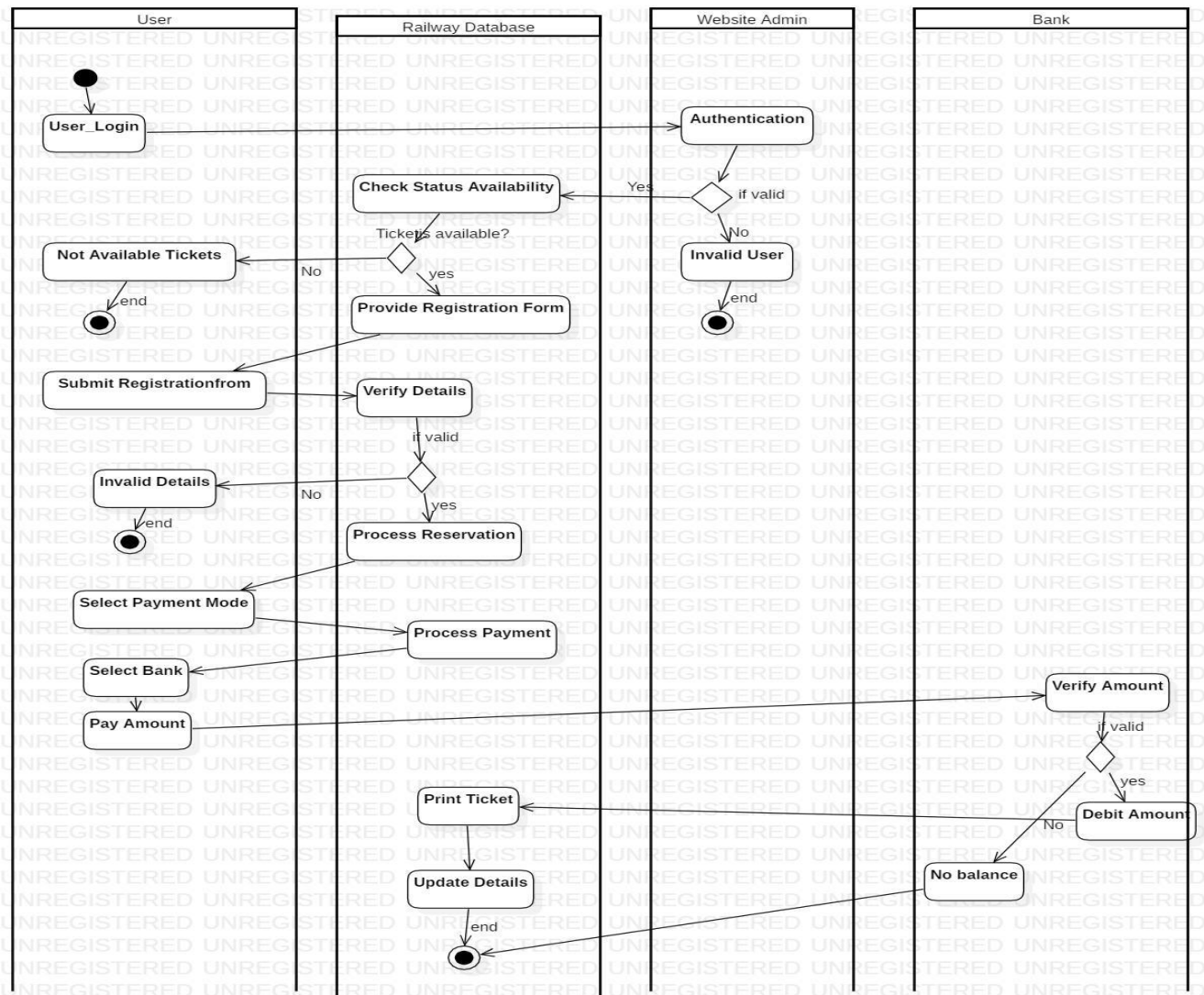
### AIM:-

To draw an Activity Diagram for the Railway Reservation System.

### Requirements:-

StarUML application is required and also knowledge on the project is must to prepare Activity diagram.

### Theory:-



### Conclusion:-

Activity Diagram for Railway Reservation System is drawn using StarUML.

## **EXPERIMENT - 6 (STATE CHART DIAGRAM)**

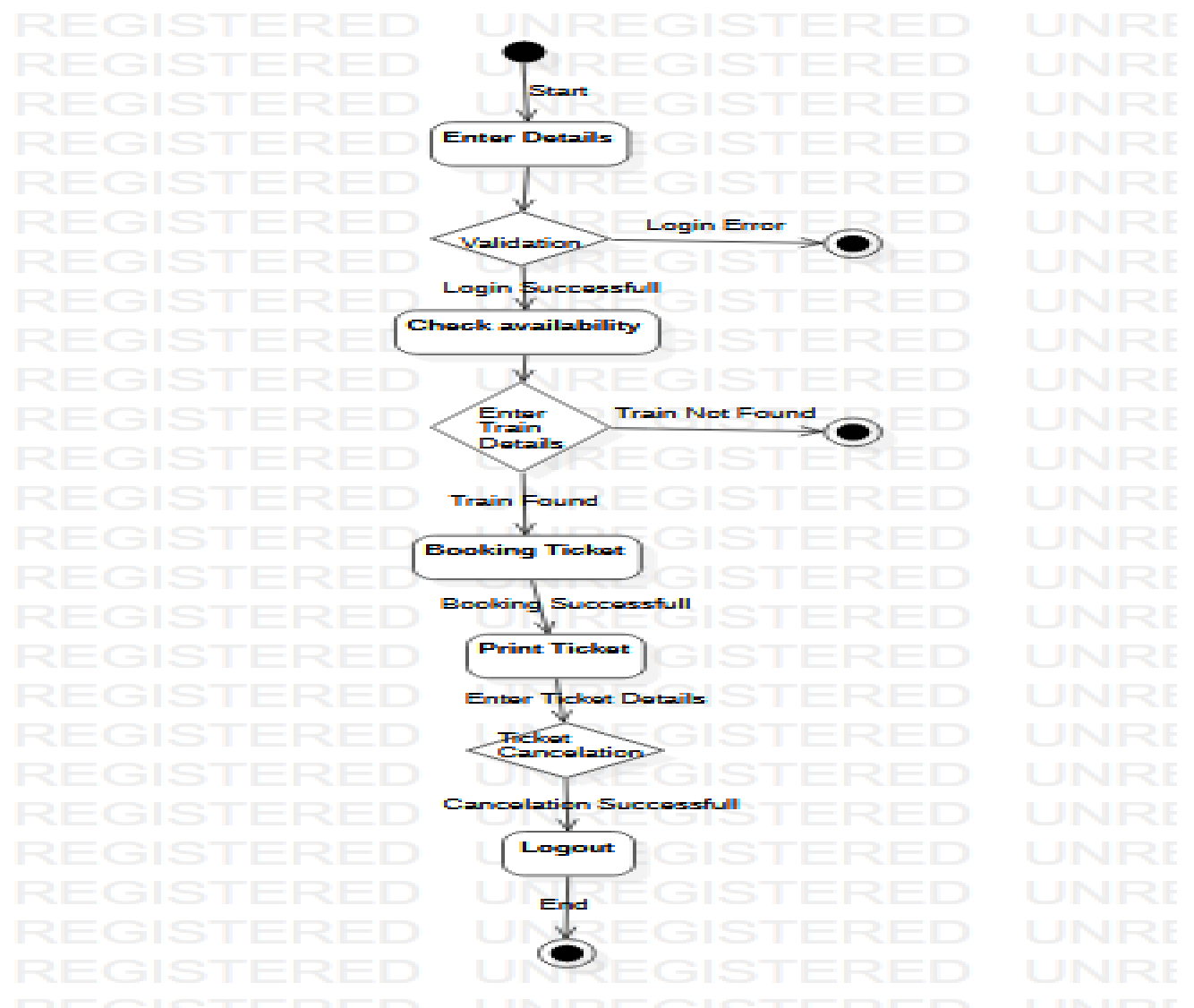
### **AIM:-**

To draw a state chart diagram for Railway Reservation System using StarUML.

### **Requirements:**

StarUML application is required and also knowledge on the project is must to prepare start chart diagram.

### **Theory:-**



### **Conclusion:**

State Chart Diagram for Railway Reservation System is prepared using StarUML.

## **EXPERIMENT - 7 ( DFD)**

### **AIM:**

To draw Data Flow Diagram for Railway Reservation System project.

### **Requirements:**

StarUML application is required and also knowledge on the project is must to draw Data Flow Diagram.

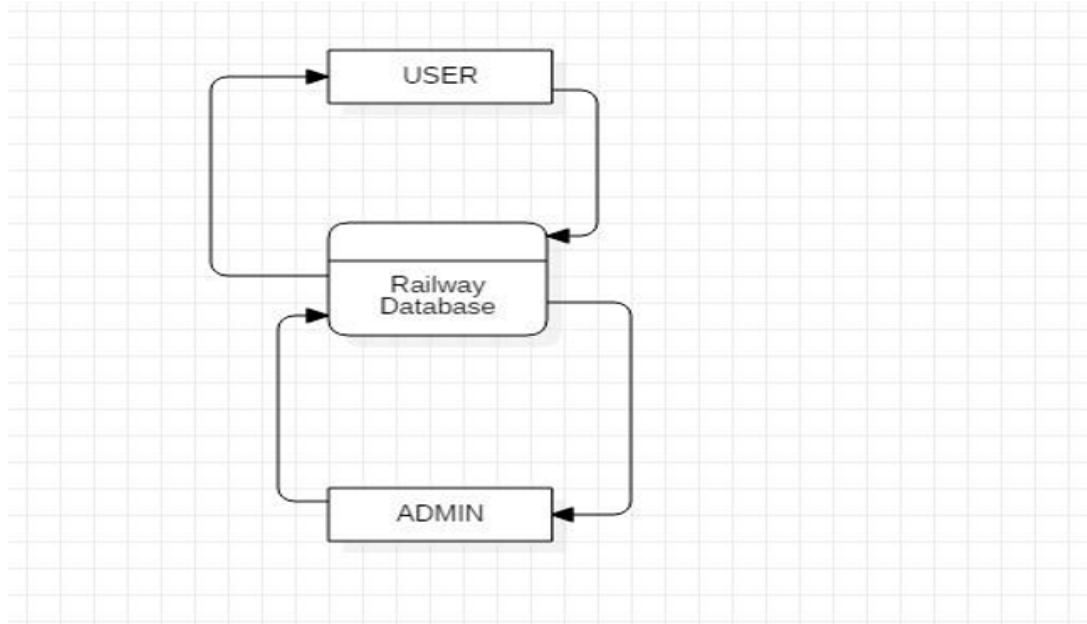
### **Theory:**

There are total 3 levels in drawing Data Flow Diagrams.

They are :

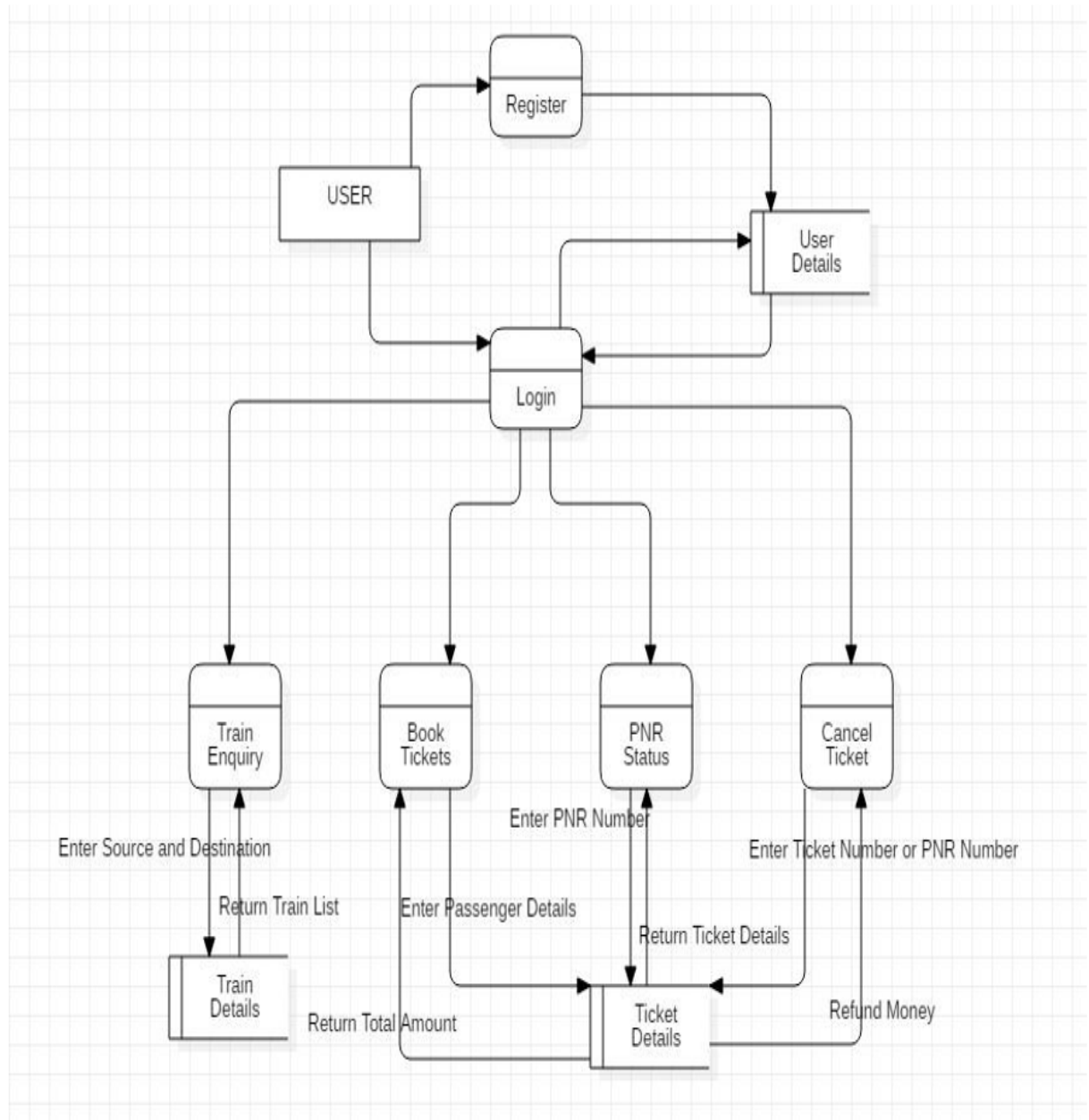
1. Level 0
2. Level 1
3. Level 2

# DFD Level-0

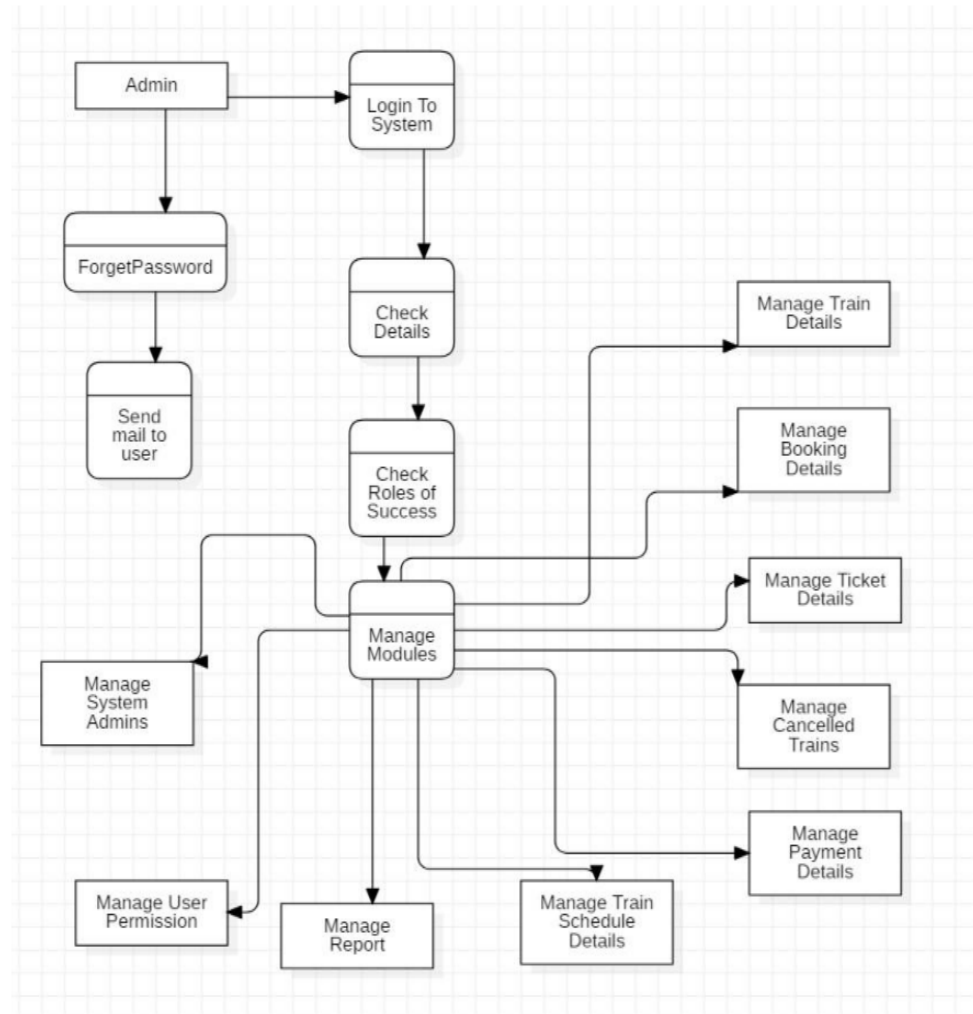




# DFD Level-1



# DFD Level-2



## Conclusion:

Data Flow Diagrams for Railway Reservation System are drawn using StarUML.

## **EXPERIMENT - 8 ( DESIGN DOCUMENT)**

### **AIM:**

To develop a Software Design Document for Railway Reservation System.

### **Requirements:**

StarUML application is required and also knowledge on the project is must to develop Software Design Document.

### **Theory:**

## **Table of Contents**

### 1.Introduction

- a. Purpose
- b. Product Scope
- c. Definitions
- d. References
- e. Overview

### 2. Use Cases

- a. Actors
- b. List of Use Cases
- c. Use Case Diagrams
- d. Use Cases

### 3. Design Process

- a. Introduction
- b. System Features
  - a. Hardware Requirements

- b. Software Requirements
  - c. Design Constraints and Assumptions
- 4. Object Description (Different classes in the project)
- 5. Data Design
  - a. ER Diagram
  - b. Data Flow Diagram
- 6. Dynamic Model
  - a. Sequence Diagrams
  - b. State Diagrams
- 7. Non Functional Requirements
  - a. Performance Requirement

## **1. Introduction: -**

The introduction of Software Requirement Specification provides an overview of the entire SRS purpose, scope, definitions, acronyms, abbreviations, references and overview of SRS.

It is a complete description of the behaviour of a system to be developed. It includes Functional Requirements and Non-Functional Requirements.

### **1.2 Product Scope: -**

“Railway Reservation System” is an attempt to simulate the basic concepts of an online Reservation System. The System enables to perform the following actions:

- Search for a Train
- Booking of a Selected Flight

- Payment
- Cancellation
- Freight Revenue enhancement
- Passenger Revenue enhancement
- Improved & optimized service
- PNR Status which is used to know live prediction of waitlisted train tickets.
- Flexible Date/Time to attract the users.
- Confirmation message or call for their reservation.

### **1.3 Definitions: -**

This should define all technical terms and abbreviations used in this document.

- ❖ PRS: - passenger reservation system
- ❖ NTES: - National Train Enquiry System
- ❖ IVRS: - Interactive Voice Response system
- ❖ SRS: - Software Requirements Specification
- ❖ STD: - State Transition Diagram
- ❖ PNR: - Passenger Name Record
- ★ Freight: - Transportation cost
- ★ Hamper: - Impede the movement or progress of
- ★ Erroneous: - incorrect
- ★ Sluggish: - slow moving or inactive

### **1.4. References**

[www.google.com](http://www.google.com)

[www.irctc.co.in](http://www.irctc.co.in)

[www.indianrail.co.in](http://www.indianrail.co.in)

## **1.5. Overview**

This document contains the designing techniques according to problem statement that the current system is facing which is hampering the growth opportunities of the Railway Reservation System. It further lists and describes briefly about design document techniques like Function Oriented Design and Object Oriented design techniques to effectively designing the required Software Product.

## **2. Use Cases**

A Use Case diagram in the Unified Modeling Language(UML) is a type of behavioral diagram defined by and created from a Use-Case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actor, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Interaction among actors is not shown on the use case diagram. If this interaction is essential to a coherent description of the desired behavior, perhaps system or use case boundaries should be reexamined. Alternatively, interaction among actors can be part of the assumptions used in the use case.

### **Use Case:**

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

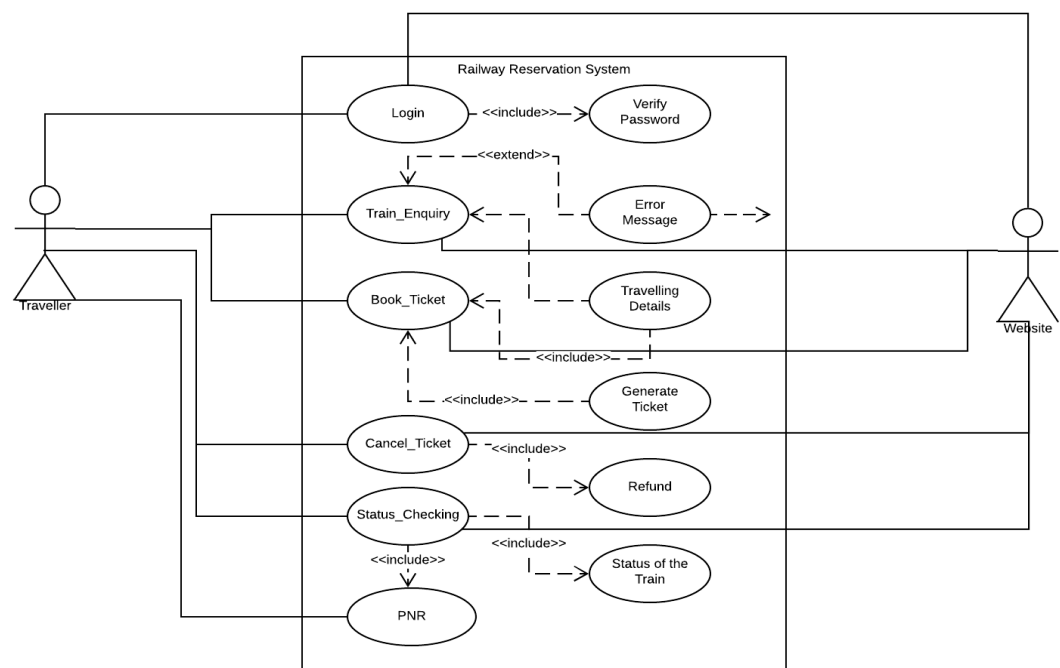
### **Actors:**

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

### **System Boundary Boxes:**

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

### Use Case Diagram for Railway Reservation System



## 3. Design Process

**Introduction:-**During the software design phase, the design document is produced, based on the customer requirements as documented in

the SRS document. We can state the main objectives of the design phase, in other words, as follows: -

**The activities carried out during the design phase (called as design process) transform the SRS document into the design document.**

## **A. System Features**

### **a. Hardware Requirements**

It deals with the hardware requirements like Ram size, Cache size, Processor type, Internal Memory size etc. required for the software to run efficiently.

- Intel core I3 processor
- 250GB of Internal Memory
- Minimum 2GB of Ram
- Cache 2MB

### **b. Software Requirements**

Any Operating System like windows (any version like 7,8.1,10), Linux (Fedora, Zorin, Ubuntu) are primary requirements for Software Development.

The Systems should have Internet connection is secondary requirement.

The Software should have security, portability, Efficiency and Flexibility.

## **Design Constraints and Assumptions**

Design Constraints can be imposed by other standards, hardware limitations etc...



### **Standard Compliance:**

Specify the requirements derived from existing standards or regulations. They might include:

1. Report Format

2. Data Naming

3. Accounting Procedures

4. Audit Tracing. For example, this cloud specifies the requirement for software to trace processing activity.

Such traces are needed for some applications to meet minimum government or financial standards. An audit trace requirement might, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

### **Hardware Limitations:**

Identify the requirements for the software to operate inside various hardware constraints.

### **Quality Characteristics:**

There are a number of quality characteristics that can apply to software. Pick the ones most important to this product and develop a section for each one. Definitions of the quality characteristics follow.

#### **Correctness**

- extent to which program satisfies specifications, fulfills user's mission objectives

#### **Efficiency**

- amount of computing resources and code required to perform function.

#### **Flexibility**

- effort needed to modify operational program

### **Integrity/security**

- extent to which access to software or data by unauthorized people can be controlled.

### **Interoperability**

- effort needed to couple one system with another.

### **Maintainability**

- effort required to locate and fix an error during operation.

### **Portability**

- effort needed to transfer from one h/w or s/w environment to another.

### **Reliability**

- extent to which program performs with required precision.

### **Reusability**

- extent to which it can be reused in another application

### **Testability**

- effort needed to test to ensure performs as intended.

### **Usability**

- effort required to learn, operate, prepare input, interpret output.

## **4.Object Description**

In Object Description we will discuss all the classes used in our Software Product and their functionalities. Some classes in our Software Product are

1.Login

2.Train Enquiry

3.Book Ticket

4. Train Status

5.Cancel Ticket.

## **1.Login**

We are providing Login class in which users will login to their existed account and used provided services of the Software Product.

## **2.Train Enquiry**

In this class users are going to enquire the trains about the source and destination. If the requested trains are not present, we will notify them about their trains regarding time, date etc..

## **3.Book Ticket**

After completion of above two classes, user is going to book the ticket and choose berths like AC, Non-AC, General ,Sleeper etc..

After filling all these details in the book ticket class, it will redirect to payment page which is used to pay the money by using third party frameworks like paytm, phonepe, Google Pay etc..

## **4. Train Status**

In this class, we will check the status of the train by using PNR(Passenger Name Record).

## **5.Cancel Ticket**

If the user don't want to travel after booking the ticket then the user use this class for cancelling the ticket.

**5.Data Design:-**The **data** objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in **data** dictionary provide a base for **data design** activity.

It includes ER(Entity Relationship) diagrams and DFD(Data Flow Diagrams)etc..

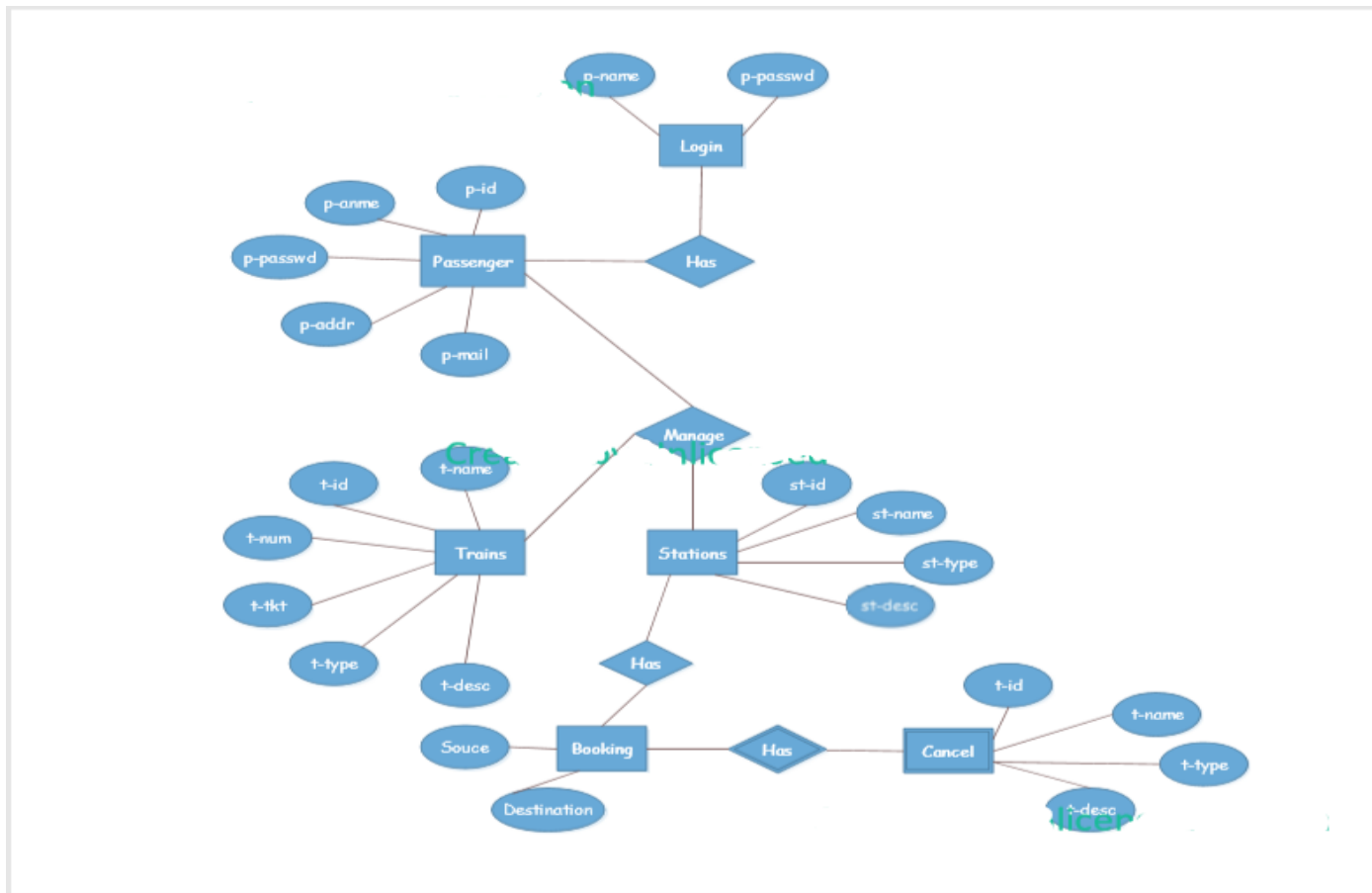
#### **Entity Relationship Diagrams:**

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

#### **Data Flow Diagrams:**

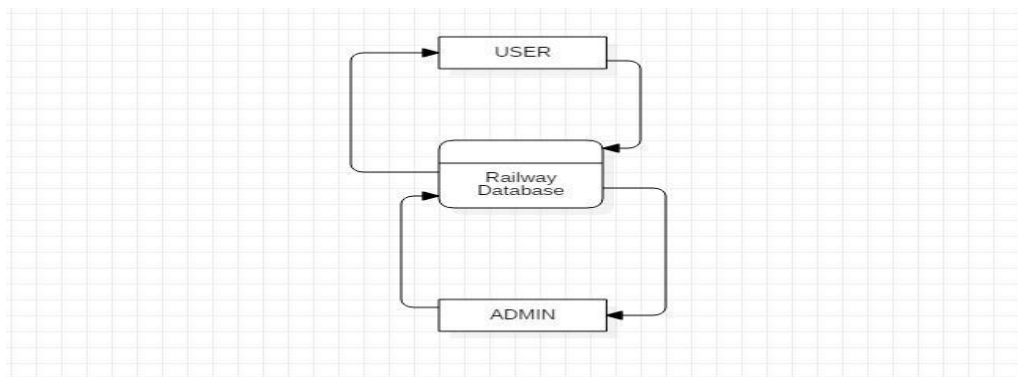
A data-flow diagram is a way of representing a flow of a data of a process or a system The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

##### **A. Entity Relationship Diagram**

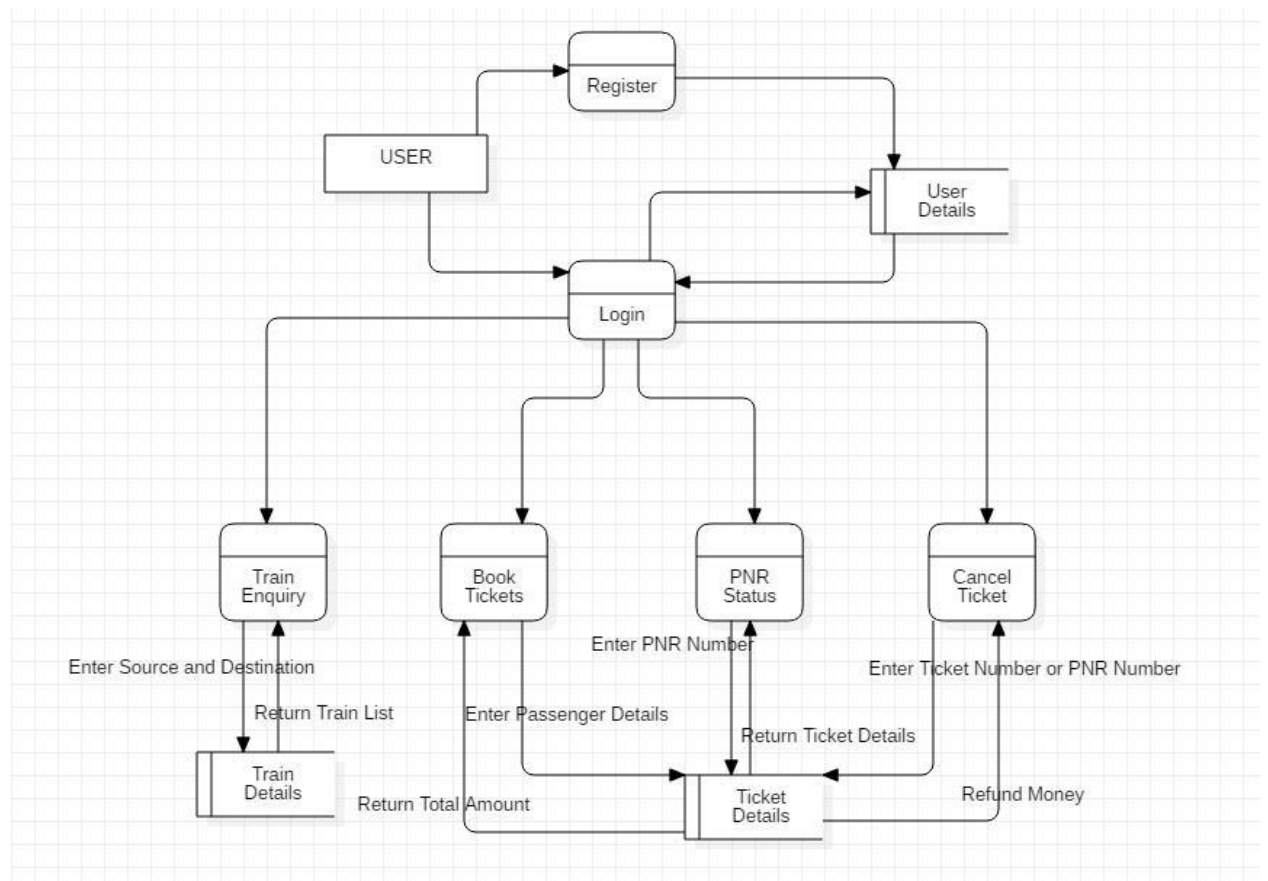


## B.Data Flow Diagrams: -

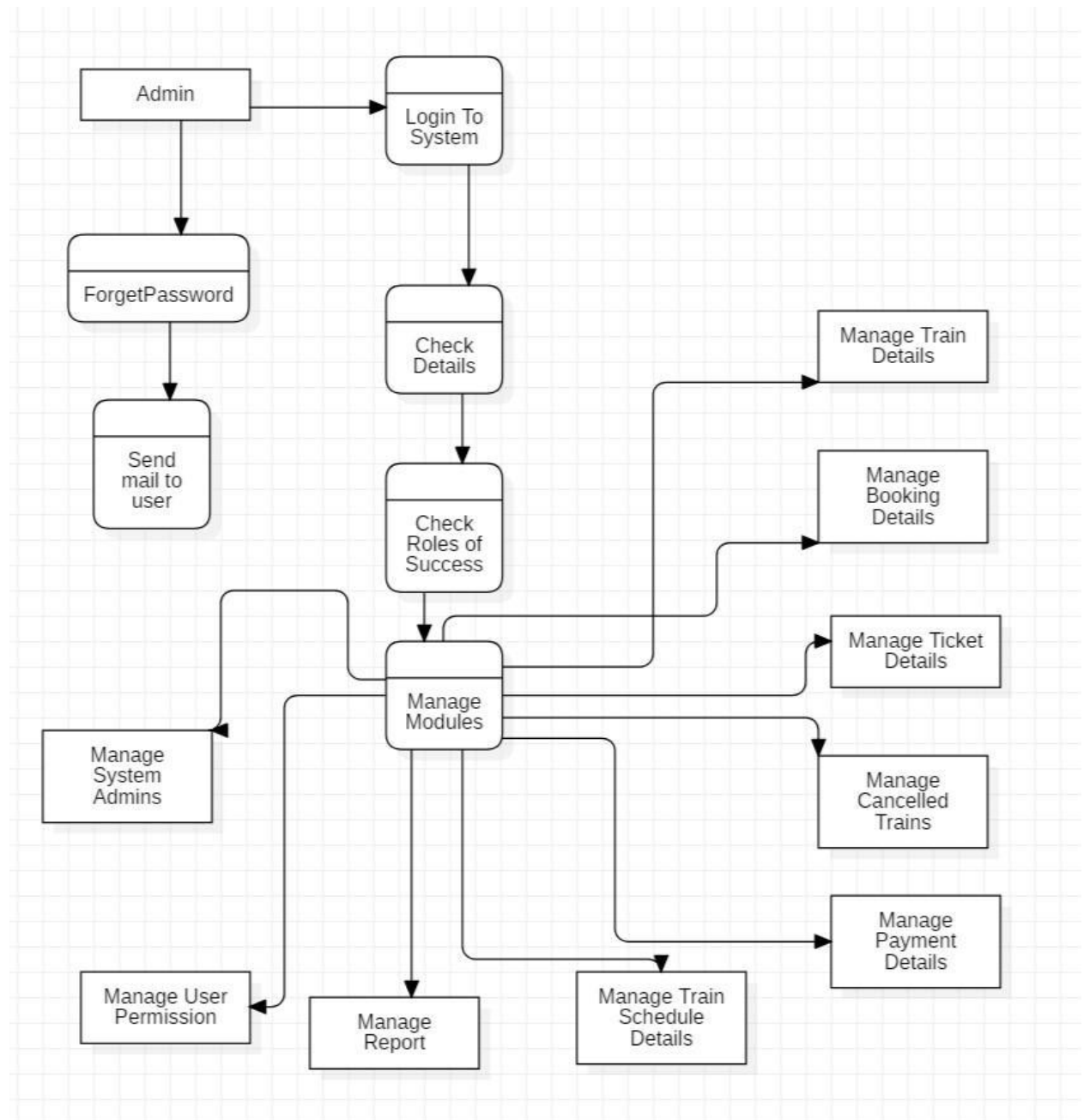
### DFD-Level0



### DFD-Level1



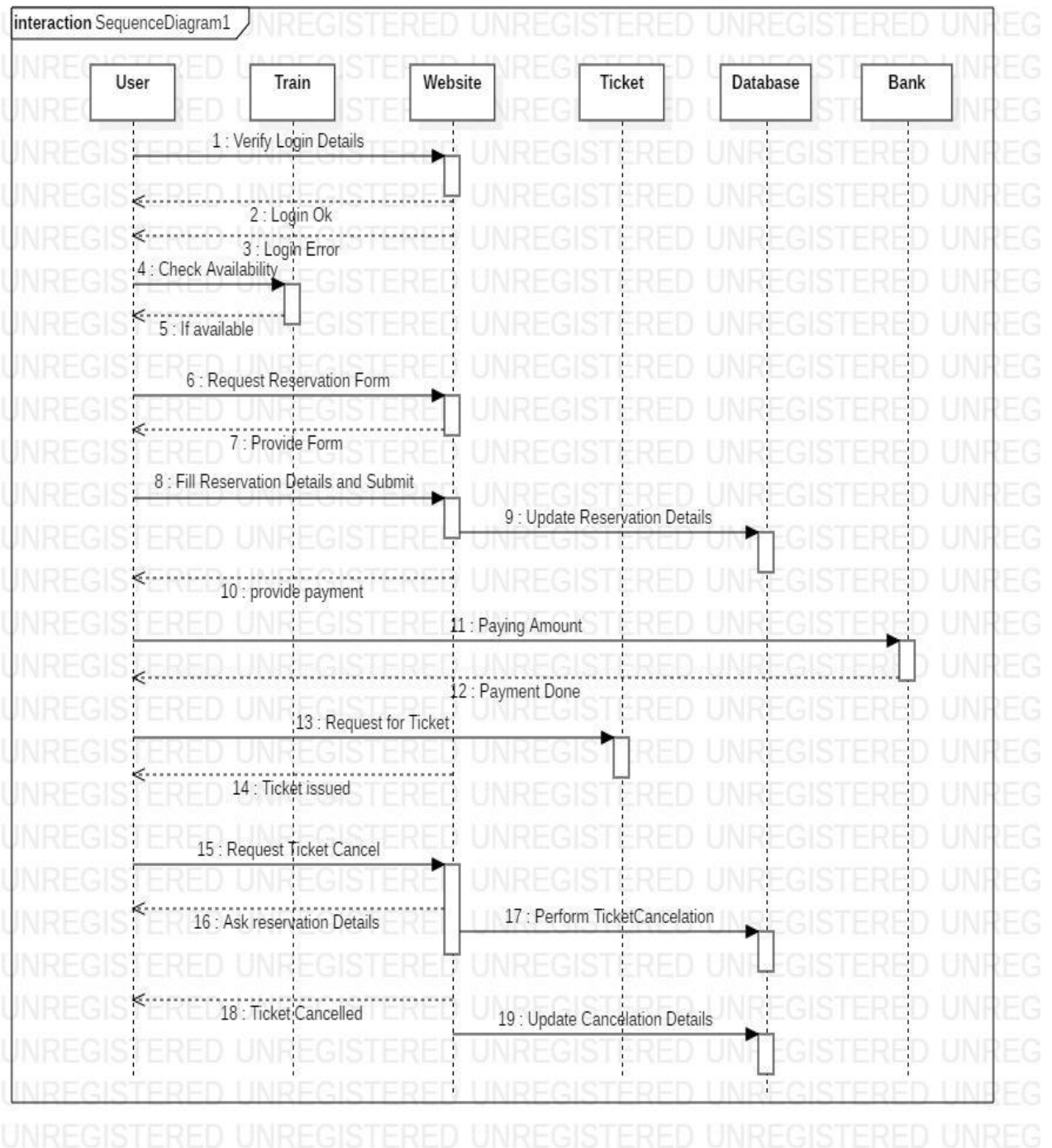
## DFD-Level2



## 6. Dynamic Model:

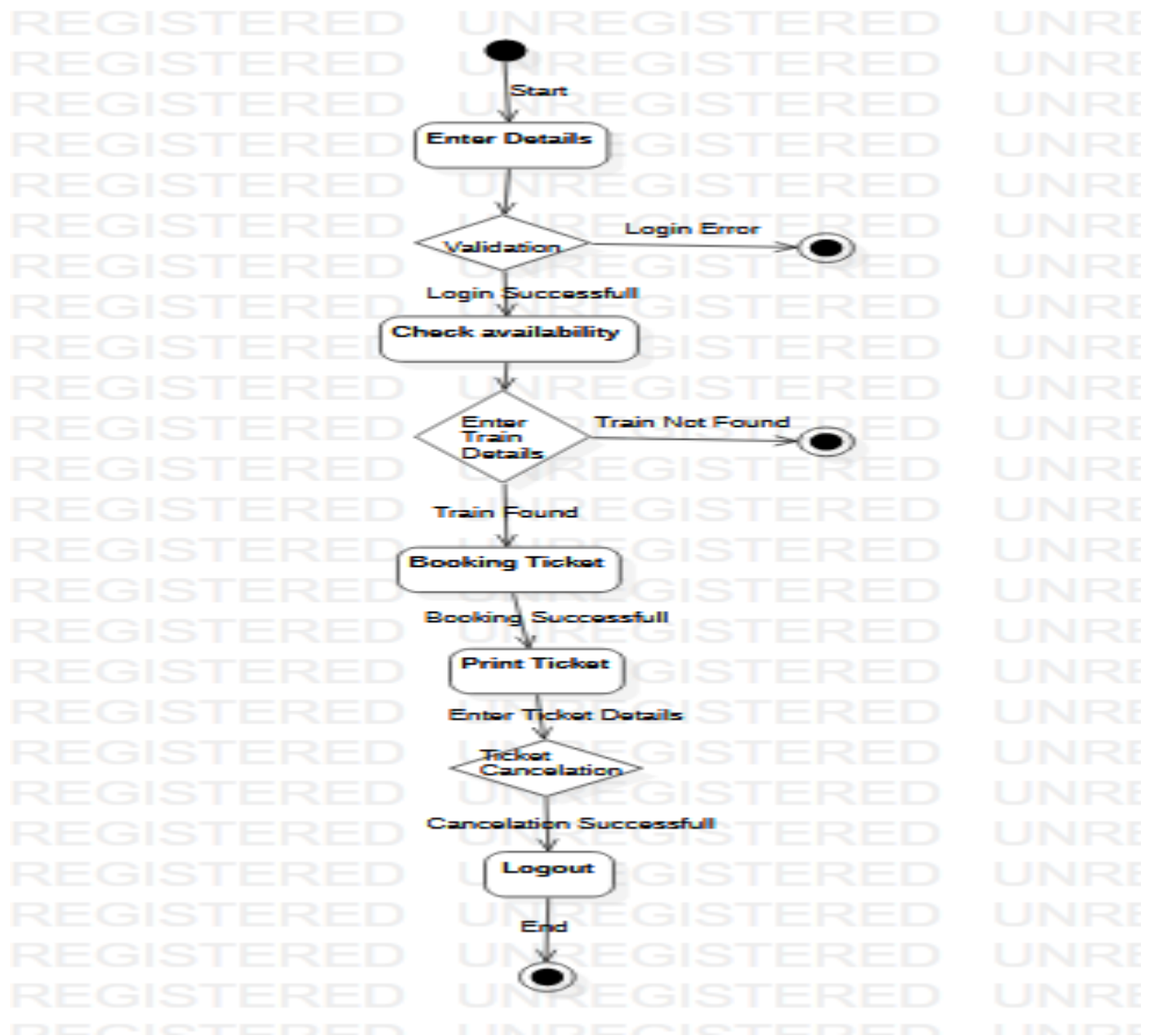
The **dynamic model** is used to express and **model** the behavior of the system over time. Sequence diagrams are used to display the interaction between users, screens, objects and entities within the system.

## Sequence Diagram



## B. State Chart Diagram





## 7. Non Functional Requirements

Non-functional requirements are those requirements which elaborate the performance characteristic of the system and define the constraints on how the system will do so.

- Defines the constraints, targets or control mechanisms for the new system.
- Describes how, how well or to what standard a function should be provided.
- Specified by technical peoples e.g. Architect, Technical leaders and software developers.

- They are sometimes defined in terms of metrics (something that can be measured about the system) to make them more tangible.
- Identify realistic, measurable target values for each service level.
- These include reliability, performance, service availability, responsiveness, throughput and security.

#### **A. Performance Requirements**

1. **User Satisfaction:** -The system is such that it stands up to the user expectations.
2. **Response Time:** -The response of all the operation is good. This has been made possible by careful programming
3. **Error Handling:** - Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
4. **Safety and Robustness:** - The system is able to avoid or tackle disastrous action. In other words, it should be fool proof. The system safeguards against undesired events, without human intervention.
5. **Portable:** - The software should not be architecture specific. It should be easily transferable to other platforms if needed.

**User friendliness:** - The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

**Conclusion:** Software Design Document is prepared successfully for the Railway Reservation System.

## **EXPERIMENT - 9 ( FUNCTIONAL POINT ANALYSIS)**

### **AIM:**

To develop a Functional Point Analysis for Railway Reservation System.

### **Requirements:**

Good knowledge on the Online e-trading project is required and also no of inputs and outputs are to be known.

### **Theory:**

Three steps are to be carried out to perform calculation function point. They are

- 1) Compute the unadjusted function point (UFP) using a heuristic expression.
- 2) Refine UFP to reflect the actual complexities of the different parameters used in UFP computation.
- 3) Compute FP by further refining UFP to account for the specific characteristics of the project that can influence the entire development effort.

#### **In Step1: -**

**$$\text{U.F.P} = (\text{Number of inputs} \times 4) + (\text{Number of outputs} \times 5) + (\text{Number of inquiries} \times 4) + (\text{Number of files} \times 10) + (\text{Number of interfaces} \times 7)$$**

#### **In Step2: -**

##### **Refinement of Function Point Entities**

<b>Type</b>	<b>Simple</b>	<b>Average</b>	<b>Complex</b>
Input	3	4	6
Output	4	5	7
Inquiry	3	4	6
No of Files	7	10	15
No of Interfaces	5	7	10

#### **In Step3: -**

There are 14 parameters that affect the development of the overall software. Each of these 14 parameters is assigned a value from 0 (not present or no influence) to 6 (strong influence).

The resulting numbers are summed, yielding the total degree of influence (DI).

A technical complexity factor (TCF) for the project is computed and the TCF is multiplied with UFP to yield FP.

The TCF expresses the overall impact of the corresponding project parameters on the development effort.

TCF is computed as  $(0.65 + 0.01 * DI)$ .

DI can vary from 0 to 84.

TCF can vary from 0.65 to 1.49.

Finally, FP is given as the product of UFP and TCF. That is,  $FP = UFP * TCF$

**In my Railway Reservation System project,**

No of inputs = 6

- 1) User registration form
- 2) Login form
- 3) Reservation Form
- 4) Ticket Cancellation Form
- 5) Train Enquiry Form
- 6) Train Status form

No of outputs = 4

- 1) Login Time
- 2) Cancel Time
- 3) Reservation Time
- 4) Enquiry Time

No of inquiries = 4

- 1) Display Reservation Form
- 2) Display Cancellation Form
- 3) Display Enquiry Form
- 4) Providing Help Information

No of interfaces = 1

- 1) Communication with the ADMIN

No of Files = 2

- 1) User Database
- 2) Railway System Database

### **Calculation of Functional Point: -**

Here we take the complexity of the refining parameters as Average. Just look at the average values for the refining parameters.

$$\begin{aligned}\text{U.F.P(Railway Reservation System)} &= (4*6) + (5*4) + (4*4) + (10*2) \\ &+ (7*1) \\ &= 24 + 20 + 16 + 20 + 7 \\ &= 87\end{aligned}$$

### **Calculation of D.I:**

Here we take the complexity values of 14 parameters is taken as average (3).

0 = No influence

6 = Strong influence

$$\text{D.I} = 14*3 = 42$$

### **Calculation of T.C.F:**

Formula provided by albrecht by observing the projects as taken as

$$\begin{aligned}\text{T.C.F} &= 0.65 + (0.01 * \text{D.I}) \\ &= 0.65 + (0.01 * 42) \\ &= 0.65 + 0.42 \\ &= 1.07\end{aligned}$$

**Calculation of F.P:**

Formula: -

$$\begin{aligned}\text{F.P} &= \text{U.F.P} * \text{T.C.F} \\ &= 87 * 1.07 \\ &= 93.09\end{aligned}$$

**The Functional Point of Railway Reservation System = 93.09**

## **EXPERIMENT - 10 (COCOMO MODEL)**

### **AIM:-**

Estimating effort cost for Railway Reservation System using COCOMO model.

### **Requirements:-**

Good knowledge on the Railway Reservation System is required to estimate the size estimation of the Railway Reservation System.

### **Theory:-**

COCOMO(Constructive Cost Model) is a regression model based on LOC(Lines of Code).It prescribes a three stage process for project estimation. In the first stage, an initial estimate is arrived at. Over the next two stages, the initial estimate is refined to arrive at a more accurate estimate. COCOMO uses both single and multivariable estimation models at different stages of estimation. This model is used to predict the Size, Effort, Cost, Time and Quality of any Software Project.

**Effort** : Measured in person-months unit.

**Scheduled** : Measured in weeks,months.

Different model of COCOMO have been proposed to predict the cost estimation at different levels.

COCOMO applied on three classes Software Project.

1. Organic Mode
2. Semidetached Projects
3. Embedded Projects

### **1.Organic Mode:**

In this mode relatively small simple software projects with a small team are handled.

Project size : 2-50 kloc

Nature of the Project : Small size, Experience developers.

Innovations : Little(small)

Deadlines of the Project : Not Tight Schedule.

## **2.Semidetached Mode:-**

In this class, an intermediate projects in which teams with mixed experienced levels are handled.

Project size : 50-300 kloc

Nature of the Project : Medium size.

Innovations : Medium

Deadlines of the Project : Medium Time.

## **3.Embedded Project Mode:-**

In this class, large projects are developed.

Project size : above kloc

Nature of the Project : large projects real time systems.

Innovations : Significant

Deadlines of the Project : Tight Schedule(within time).

## **COCOMO Models:-**

1.Basic COCOMO Model

2.Intermediate COCOMO Model

3.Detailed (or) Complete COCOMO Model

### **1.Basic COCOMO Model:-**

This model is used to estimate the software size, effort , development time and number of persons required to develop the project.

It can be used to quick and slightly rough calculations of software costs.

$$E = a_b(KLOC)^{bb}$$

$$D = c_b(E)^{db}$$

$$P = E / D$$



Where E is Effort applied in Person-month(PM),

D is the development time ,

P is the total number of persons required to complete the project.

$a_b, b_b, c_b, d_b$  are the coefficients and they are :-

Software Projects	$a_b$	$b_b$	$c_b$	$d_b$
Organic	2.4	1.05	2.5	0.38
SemiDetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

The above formula is used for the cost estimation for the software project in the basic cocomo model.

## **2.Intermediate Model:-**

It is the extension of Basic COCOMO model and it includes 15 cost driver attributes.

15 cost driver attributes are divided into four types. Those are product attribute, hardware attributes, personal attributes, and project attributes.

$$E = a((KLOC)^b) * EAF$$

S/w projects	a	b
Organic	3.2	1.05
Semi detached	3.0	1.12
Embedded	2.8	1.20

## **Detailed cocomo model:-**

It incorporates all characteristics of the intermediate version with an assessment of the cost drivers impact on each step of the s/w engineering.

Six phases of cocomo are

1. Planning & requirements
2. System design
3. Detailed design
4. Module code & test
5. Cost constructive model

The effort is calculated as a function of program size and set of cost drivers are given according to each phase of the s/w life cycle

**Problem:-** Estimate that our project to be 400kloc calculate the effort for the 3-modules i.e

Organic .semi detached ,embedded in basic cocomo model

**Equation:-**  $E = a_b (kloc)^b_b$

**Organic model**  $E = 2.4 (400)^{1.05}$   
 $= 1295.31 \text{ PM}$

**Semi detached**  $E = 2462.79 \text{ PM}$

**Embedded**  $E = 4772.81 \text{ PM}$

Where **E** is the effort in staff months.

**a** and **b** are coefficient to be determined

**KLOC** is thousands of Lines of code.

### **Selected project:-**

In java , function point takes 10.6 hrs.

Function point for SRMS

For function point  $= 130.45 \times 10.6 = 1382.7$

In one day employee's can work 8 hrs

Number of days to complete  $= \text{total hours} / 8$

$= 1382.7 / 8$

$= 172.84$  days are required

### **CONCLUSION:**

COCOMO model analysis successfully done for student result management of system.