Software Requirements Specification

for

<Railway Reservation System>

Version 1.0

Prepared by <6th Group>

<Capgemini>

<date created>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this source is to describe the railway reservation system which provides the train timing details, reservation, billing and cancellation on various types of reservation namely,
• Confirm Reservation for confirm Seat.
• Reservation against Cancellation.

The origin of most software systems is in the need of a client, who either wants to automate the existing manual system or desires a new software system. The software system is itself created by the developer. Finally, the end user will use the completed system. Thus, there are three major parties interested in a new system: the client, the user, and the developer. Somehow the requirements for the system that will satisfy the needs of the clients and the concerns of the users have to be communicated to the developer. The problem is that the client doesn't usually design the software or the software development process and the developer does not understand the client's problem and the application area. This causes a communication gap between the parties involved in the development of the project.

The basic purpose of Software Requirement Specification (SRS) is to bridge this communication gap. SRS is the medium through which the client's and the user's needs are accurately specified; indeed, SRS forms the basis of software development.

Another important purpose of developing an SRS is helping the clients understanding their own needs. An SRS establishes the basis for agreement between the client and the supplier on what the software product will do.

An SRS provides a reference for validation of the final product. A high-quality SRS is a prerequisite to high quality software and it also reduces the development cost.

A few factors that direct us to develop a new system are given below -:

1. Faster System
2. Accuracy
3. Reliability
4. Informative
5. Reservations and cancellations from anywhere to any place

1.2 Document Conventions

- Main Heading: Bold
- Sub Heading: Bold

1.3 Intended Audience and Reading Suggestions

- Administrator
- Passenger
- Developer

1.4 Project Scope

Railways Reservation System" is an attempt to simulate the basic concepts of an online Reservation system. The system enables to perform the following functions

- Search for train.
- Booking of a selected train.
- Payment.
- Cancellation.
- Passenger revenue enhancement.
- Improved and optimized service.

1.5 References

- IEEE SRS Format.
- www.yatra.com
- www.irctc.co.in
- www.indianrail.gov.in
- www.google.com

2. Overall Description

2.1 Product Perspective

It enables us to maintain the railway train details like their timings, number of seats available and reservation billing and cancelling the tickets.

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 led 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.

- 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.

- A railway has many offices around the world, an absence of a link between these offices leads 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 reduces errors because most the calculations are performed by the application only.

- The passenger, reservation, cancellation list can easily be retrieved and any required addition, deletion or updating can be performed.

- The system provides for user-ID validation, hence unauthorized access is prevented.

2.2 Product Features

- Provides the searching facilities based on various factors such as Train, payment and ticket.
- It tracks and manages the information of the booking and Trains.
- All the fields are validated and does not take invalid values.
- Editing, adding and updating of records is improved which results in proper resource management of train data.
- It allows payment through credit or debit cards.

2.3 User Classes and Characteristics

- Knowledgeable user.
- Technical expertise: - User should be comfortable using general purpose applications on the computer system.

2.4 Operating Environment

The software system must operate within common web browser environments using Secure Socket Layer (SSL) / Transport Layer Security (TLS) cryptographic protocols at a minimum encryption level of 128 bits. The minimum set of browsers that must be supported is:

- Google Chrome Version 97+
- Mozilla Firefox Version 97+

2.5 Design and Implementation Constraints

There are a number of factors in the client's environment that may restrict the choices of a designer. Such factors include standards that must be followed, resource limits, operating environment, reliability and security requirements and policies that may have an impact on the design of the system. An SRS (Software Requirements Analysis and Specification) should identify and specify all such constraints.

- **Standard Compliance: -** This specifies the requirements for the standards the system must follow. The standards may include the report format and accounting properties

- **Hardware Limitations: -** The software may have to operate on some existing or predetermined hardware, thus imposing restrictions on the design. Hardware limitations can include the types of machines to be used, operating system available on the system, languages supported and limits on primary and secondary storage

- **Reliability and Fault Tolerance: -** Fault tolerance requirements can place a major constraint on how the system is to be designed. Fault tolerance requirements often make the system more complex and expensive. Requirements about system behavior in the face of certain kinds of faults are specified. Recovery requirements are often an integral part here, detailing what the system should do I some failure occurs to ensure certain properties. Reliability requirements are very important for critical applications.

- **Security: -** Security requirements are particularly significant in defense systems and database systems. They place restrictions on the use of certain commands, control access to data, provide different kinds of access requirements for different people, require the use of passwords and cryptography techniques and maintain a log of activities in the system.

2.6 Assumptions and Dependencies

- Users will be having a valid user name and password to access the software.
- Software is dependent on access to internet.

3. System Features

3.1 Functional Requirements

- **User Satisfaction: -** The system is such that it stands up to the user expectations.
- **Response Time**: -The response of all the operation is good. This has been made possible by careful programming.
- **Error Handling: -** Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- **Safety and Robustness: -** The system is able to avoid or tackle disastrous action. In other words, it should be foul proof. The system safeguards against undesired events, without human intervention.
- **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.

4. External Interface Requirements

4.1 User Interfaces

*Application screenshot goes here

4.2 Communications Interfaces

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

5. Other Nonfunctional Requirements

- **Security:**
  The system uses 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 users.

- **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. Thus, the overall stability of the system depends on the stability of operations performed and its underlying operating system.

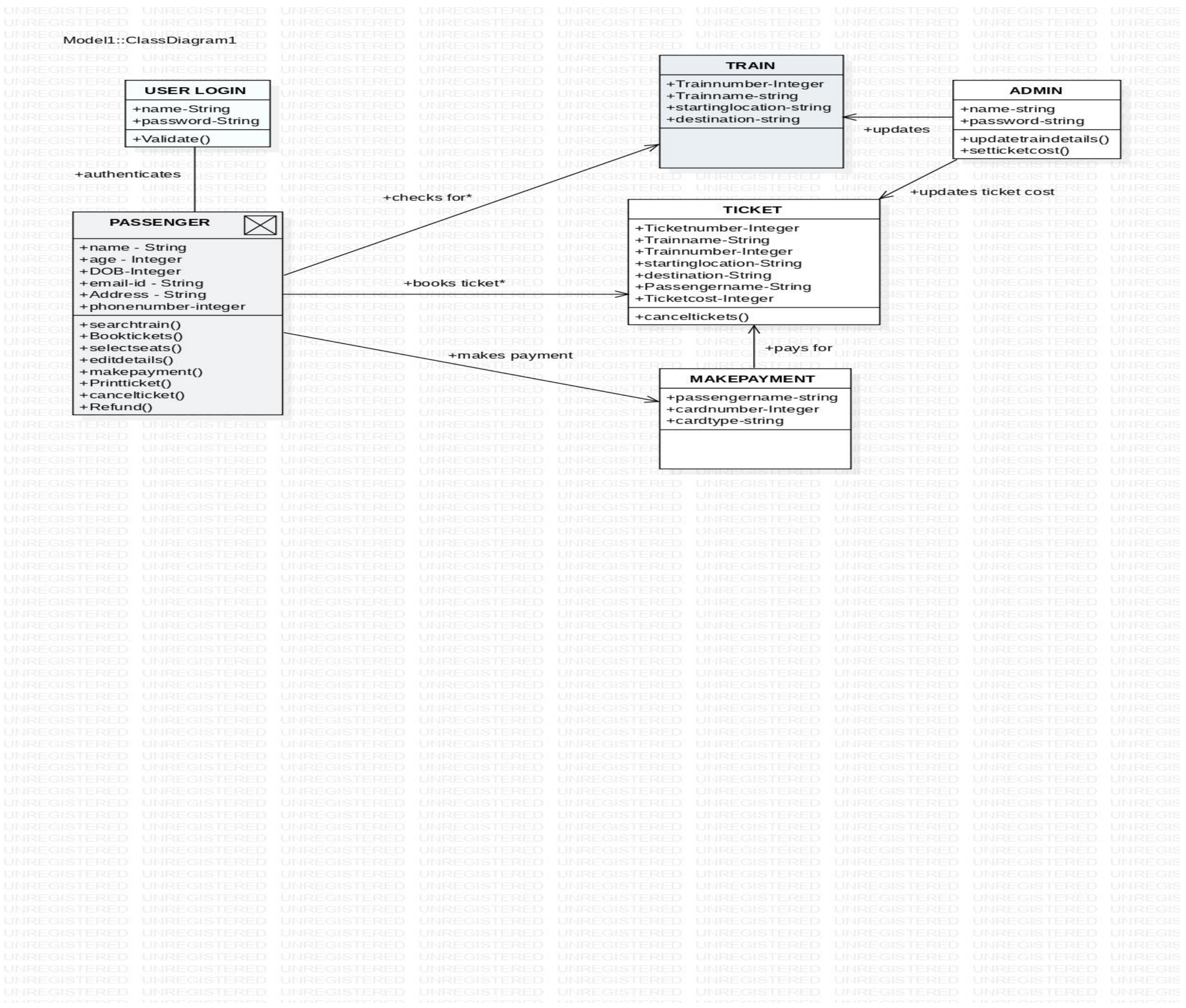
- **Availability:**
  The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs.

- **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.

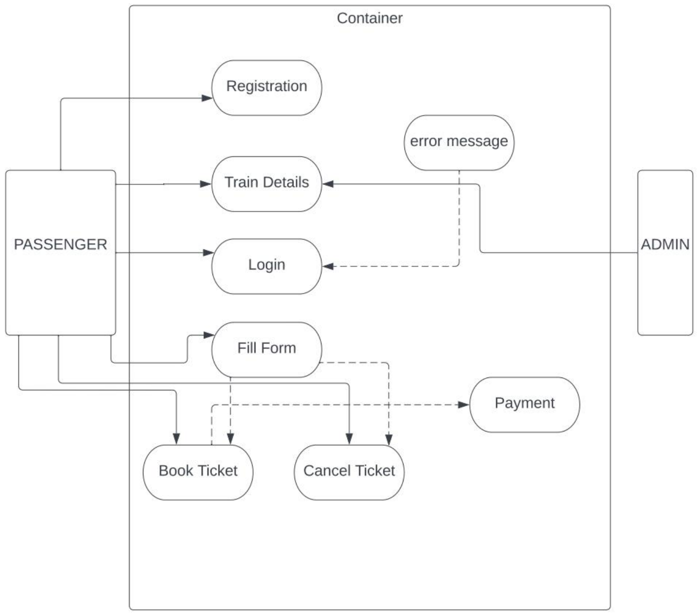
- **Supportability:**
  The code and supporting modules of the system will be well documented and easy to understand. Online User Documentation and Help System Requirements.

UML DIAGRAMS

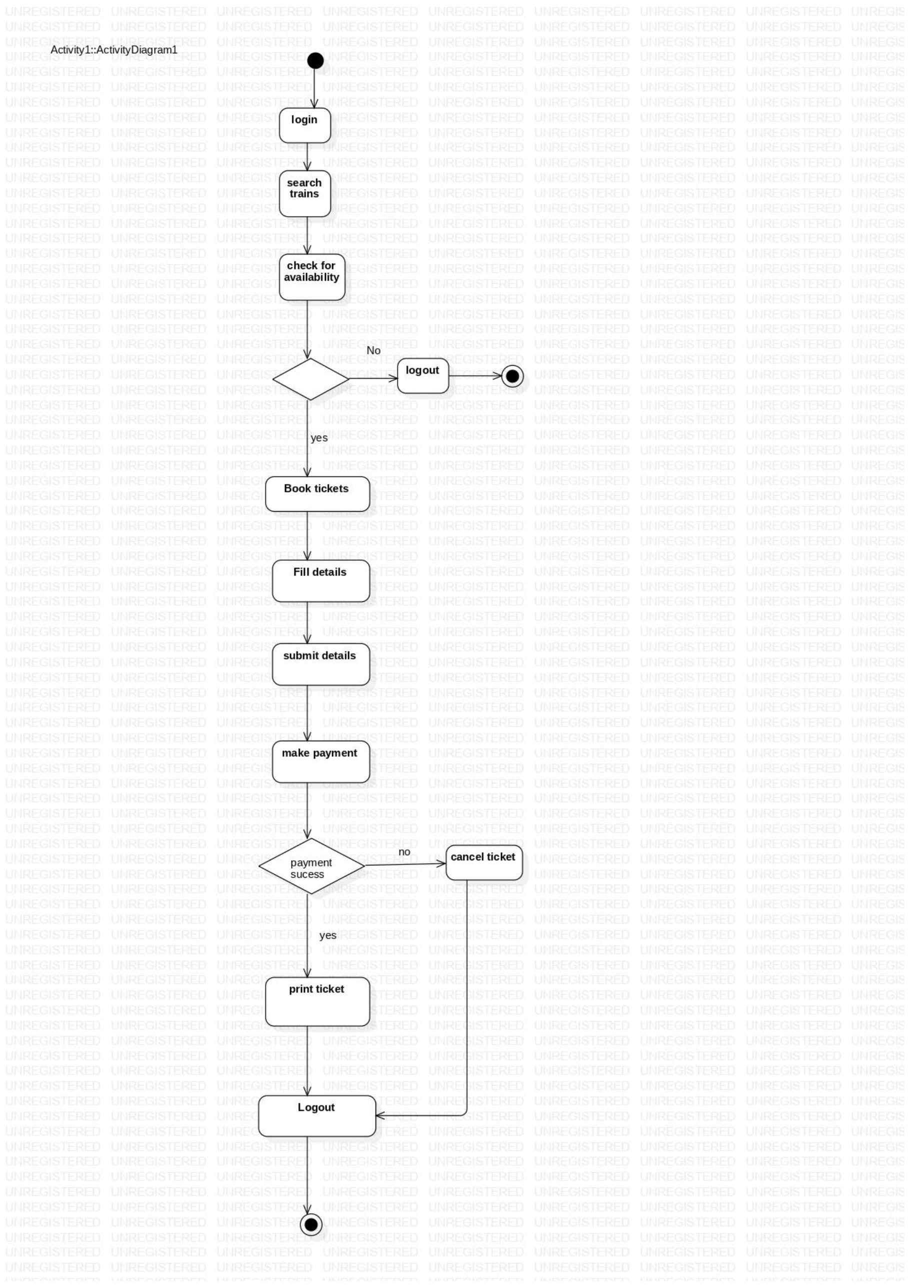
Class diagram



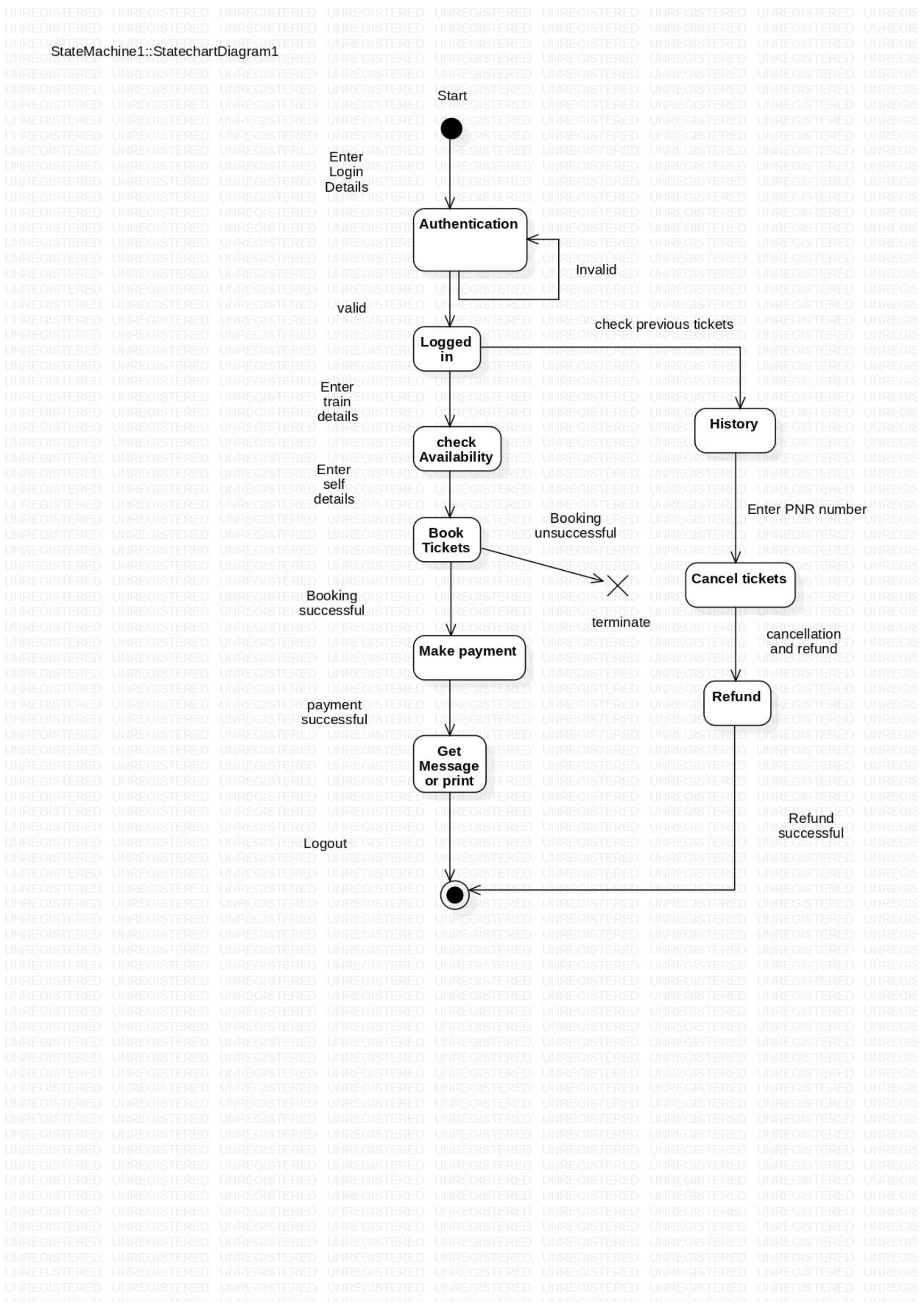
Use case diagram



Activity diagram

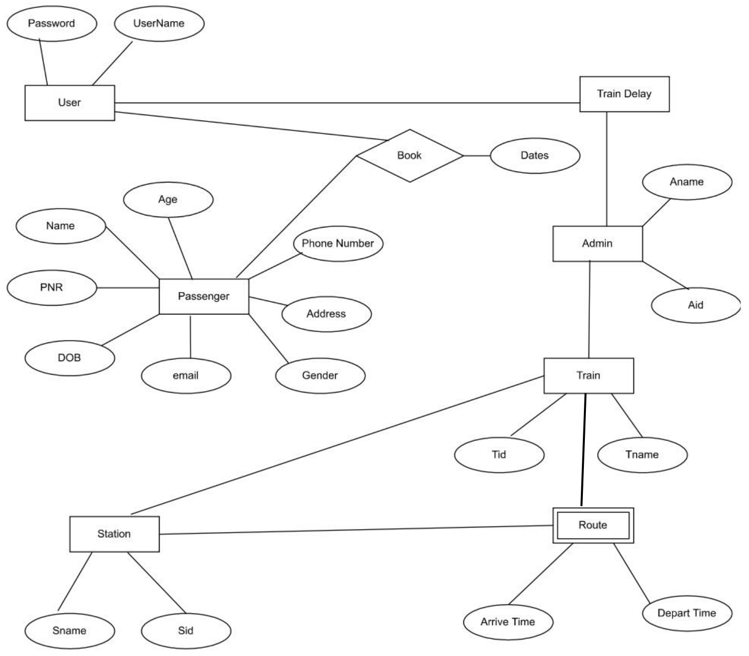


State chart diagram



ER diagram

5/24/22, 9:47 AM diagram1



1/2