A Software Requirements Specification (SRS) - a requirements specification for a software system - is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

Railway reservation system

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.
- Waiting list Reservation.
- Online Reservation.
- Tatkal Reservation.

1.2 DEFINITIONS, ACRONYMS AND ABBREVIATIONS
- NTES – National Train Enquiry System
- IVRS – Interactive Voice Response system
- PRS – passenger reservation system

1.3. SCOPE
- Freight Revenue enhancement.
- Passenger Revenue enhancement.
- Improved & optimized service

1.4 REFERENCES
www.scribd.com

2. OVERALL DESCRIPTION:
2.1. PRODUCT PERSPECTIVE:
It enables us to maintain the railway train details like their timings, number of seat available and reservation billing and cancelling the tickets.

2.1.1. **USER INTERFACE:**
Keyboard and Mouse.

2.1.2. **HARDWARE INTERFACE:**
- Printer
- Normal PC

2.1.3. **SOFTWARE INTERFACE:**
- Front end -> Visual Basic
- Back end -> MS-Access

2.1.4. **COMMUNICATION INTERFACES**
- Indian Railway’s web-site, [www.indianrail.gov.in](http://www.indianrail.gov.in) offers PRS enquiries on the internet Berth/Seat availability, Passenger Status, Fare, Train Schedule etc.,
- Mobile telephone based SMS enquiry service. A new mobile phone based facility for rail users’ viz.,

2.1.5. **OPERATING ENVIRONMENT:**
The OS types are
- Windows NT
- Windows XP
- Windows 98
- Linux

2.1.7. **OPERATIONS**
- Any Reservation counter from 8 am to 8 pm.
- Prior to 60 days of Journey.
- One form for 6 persons only.
- Reserved ticket done through pre defined Logic.
- To save time & queues Agent is others guides.

2.2. **PRODUCT FUNCTIONS:**
2.2.1. **TRAIN DETAILS:**
Customers may view the train timing at a date their name and number of tickets.

2.2.2. **RESERVATION:**
After checking the number of seats available the customers reserve the tickets.

2.2.3. **BILLING:**
After reserving the required amount of tickets, the customer paid the amount.

2.2.4. **CANCELLATION:** If the customers want to cancel the ticket, then half of the amount paid by the customer will be refunded to him.

2.3. **USER CHARACTERISTICS:**
- Knowledgeable user
- No voice user
- Expert user

2.4. **CONSTRAINTS**
- Less than 1 sec for local transactions.
- 3 sec for network transaction.
- Capable for providing transaction for 22 hrs per day.
- Uptime of PRS is 99.5 + %.

**SOFTWARE CONSTRAINTS:**
- Designing -> Rational Rose
- Developing -> Visual Basic

3. **SPECIFIC REQUIREMENTS**

3.1. **EXTERNAL INTERFACES**
- Train Delay Alert Service.
- Booking Terminals.
- Interactive voice Response System.
- Touch Screen.
- Passengers operated Enquiry Terminals.
3.2. **PERFORMANCE REQUIREMENTS:**
It is available during all 24 hours.
- Offered through Mail express, super fast, Rajdhani & Shatabdi Trains. About 1520 Trains runs daily. Variety of compartments based on comfort:
  - AC first class.
  - AC sleeper.
  - First class.
  - AC three tier.
  - AC chair car.
  - Sleeper class
  - Ordinary chair car.

Types of concerns & complexities:
- 44 types of quotas.
- 8 types of trains.
- 9 types of classes.
- 162 types of concessions.
- 127 types of bogies.

3.3. **SOFTWARE SYSTEM ATTRIBUTES:**
- Reliable
- Available
- Secure

4. **DOCUMENT APPROVAL**
The bill passed on any proposals related to railway management needs approval of Ministry of railway department.
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the 'Level 0 DFD') the system’s interactions with the outside world are modelled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.
USE CASE DIAGRAM

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 actors, 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 the system or use case boundaries should be re-examined. Alternatively, interaction among actors can be part of the assumptions used in the use case.

- Use cases

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 (optional)

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.
Railway reservation system
RAILWAY RESERVATION
LEVEL 0 DFD

RAILWAY RESERVATION SYSTEM

PASSenger

LOGIN REQUEST FOR RESERVATION

GET STATUS

RECEIVE TICKET

LOGIN

UPDATE

REPORT GENERATION

ADMINISTRATOR

RESERVATION CLERK

LOGIN

RESERVE SEAT

VIEW STATUS CANCEL SEAT
RAILWAY RESERVATION

LEVEL 1 DFD

LEVEL 1 DFD: RAILWAY RESERVATION SYSTEM