**SRS**

**On**

**Railway Reservation System**

**Submitted by**

* Adarsh Chaudhary 12000216059
* Ayushi Jha 12000216044
* Disha Sinha 12000216040
* Ashutosh Jha 12000216046

**TABLE OF CONTENTS**

**TOPICS PAGE NO.**

1. INTRODUTION 1
2. PURPOSE OF THE SYSTEM 3
3. SCOPE OF THE SYSTEM
4. OVERVIEW

1. GENERAL DESCRIPTION 7
2. PRODUCT PERSPECTIVE
3. SYSTEM INTERFACE
4. HARDWARE INTERFACE
5. SOFTWARE INTERFACE
6. COMMUNICATION INTERFACE
7. PRODUCT FUNCTIONS 8
8. NON-FUNCTIONAL REQUIREMENTS 11
9. UTILITY 13
10. RELIABILITY
11. SUPPORTABILITY
12. INTERFACING
13. STANDARDS 11
14. DESIGN CONSTRAINT
15. PORTABILITY
16. STANDARD DEVELOPMENT TOOLS
17. FUNCTIONAL REQUIREMENTS
18. RESERVATION OF TICKET 13
19. CANCELLATION OF TICKET
20. DOCUMENT APPROVAL 18

1. DIAGRAMS
   * + - 1. E-R DIAGRAM 19
         2. DATA FLOW DIAGRAMS

0 LEVEL 20

1 LEVEL 21

* + - * 1. UML DIAGRAMS

CLASS DIAGRAM 22

SEQUENCE DIAGRAM 21

* + - * 1. DATA DICTIONARY 22

7) COST ESTIMATION 23

1. REFERENCES AND BIBLIOGRAPHY 24

**INRODUCTION**

In this emerging world of computers, almost all-manual system has switched to automated and computerized system. Therefore, we are developing the software for “Railway Reservation System” to model the present system and to remove the drawbacks of the present system. This project explores how computer technology can be used to solve the problem of user.

This being a big step in terms of improvement in the railway system it is widely accepted across the country. Rather than designing manually, we have made use of computer. Use of computer has solved many problems, which are faced during manual calculation. Once data are fed, it can perform accurate functions. Therefore, to reduce the complexity and efficiency a versatile and an outsourcing railway reservation system has been developed.

This project introduces railway reservation system. It explains how reservation is being done in Indian Railways. The systematic procedure is explained. This project is developed in C++ language. All most all the header files have been used in this project. Proper comments have been given at desired locations to make the project user friendly. Various functions and structures are used to make a complete use of this language.

The customers are required to register on the server for getting access to the database and query result retrieval. Upon registration, each user has an account that is essentially the ‘view level’ for the customer. The account contains comprehensive information of the user entered during registration and permits the customer to get access to his/her past reservations, enquire about travel fare and availability of seats, make fresh reservations, and update his account details. Each passenger is allotted a unique PNR no. through which one can access his/her account.

The railway administrator is another member involved in the transactions. The administrator is required to login using a master password, once authenticated as an administrator, one has access and right of modification to all the information stored in the database. This includes the account information of the customers, attributes and statistics of stations, description of the train stoppages and physical description of coaches, all the reservations that have been made. The railway administrator has the right to modify any information stored at the server database.

This project is dedicated to model the existing railway reservation system that aims at development of Railway Reservation System that facilitates the railway customer to manage their reservations and the railway administrator to modify the backend database in a user-friendly manner. The customer and the railway administrator are two parties that interact with the database, who have different ‘view level schemas’ to the database information. The software provides a comprehensive set of features to enhance the operational limits.

Now one can easily plan the journey comfortably as the process is efficient and fast with being easy to access. The efficiency of the railway will increase result of computerization.

**(a) PURPOSE:**

The purpose of this software is to describe the Railway Reservation System which provides the rail timing details, reservation, enquiry, billing and cancellation on various types of reservation namely: -

* Confirmed reservation for available seat
* Reservation against cancellation
* PNR generation

**(b) Scope:**

The scope of this project is limited to an engineering college. This project aims to maintain passenger and train information.

Other scopes of this project are as follows:

* Freight Revenue enhancement
* Passenger Revenue enhancement
* Improved and optimised service

**(c)** **Overview:**

This project aims at development of a Railway Reservation System that facilitates the Railway customers to manage their reservations and the Railway administrators to modify the backend databases in a User-Friendly manner.

This project includes the following functions:

1) Create new database

2) Add new Record

3) Modify

4) Display record

5) Ticket reservation

6) Ticket Modification

7) Ticket Cancellation

8) Ticket printing

1. **GENERAL DESCRIPTION:**

**(a)PRODUCT PERSPECTIVE:**

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

**(b)SYSTEM INTERFACE:**

* Keyboard
* Mouse

**(c) HARDWARE INTERFACE:**

* System Model : Lenovo Ideapad 330
* Processor : Intel Core i5 8th gen 1.6 GHz
* RAM : 4 GB
* Hard disk : 500 GB

**(d) SOFTWARE INTERFACE:**

* Operating System : Windows 10 Pro
* Compiler : C++

**(e)COMMUNICATION INTERFACE:**

* Indian Railway’s website [www.indianrail.gov.in](http://www.indianrail.gov.in) PRS enquires regarding Berth availability, Passenger Status, Fare, Train Schedule etc.
* National Train Enquiry System (NTES) website [www.trainenquiry.com](http://www.trainenquiry.com) gives dynamic information about the running status of any train and its expected arrival/departure at given station.
* Mobile telephone based SMS enquiry service. A new phone based facility for rail users’ viz., Country wide extension of Universal Rail Enquiry number “139” through setting up of Interactive Voice Response System (IVRS)

**(f)PRODUCT FUNCTIONS:**

This describes notes about specific products.

* 1. TRAIN DETAILS:

Customers may view the train number, train name, train timing (arrival and departure), arrival and departure station of the train and number of seats required.

* 1. RESERVATION:

After checking the number of seats available, the customers reserve the tickets.

* 1. BILLING:

After reserving the required number of tickets, the customer paid the fare.

* 1. CANCELLATION:

If the customer wants to cancel the ticket, he can cancel it using PNR number.

**(a)EXTERNAL INTERFACES:**

* Booking Terminals
* Enquiry Terminals

**(b)PERFORMANCE REQUIREMENTS:**

It’s available during all 24 hours.

Variety of compartments based on comfort:

* AC
* Non-AC
* General

Types of concerns and complexities:

* 10 types of trains
* 3 types of classes

**(c)SOFTWARE SYSTEM ATTRIBUTES:**

* Reliable
* Available
* Secure
* Portable

1. **NON-FUNCTIONAL REQUIREMENTS:**

In systems engineering and requirements engineering, non-functional requirements are requirements that specify criteria that can be used to judge the operation of system, rather than specific behaviors. Non-functional requirements are often called qualities of a system.

Qualities, i.e. non-functional requirements can be divided into 2 main categories:

1. Execution qualities such as security and usability are observable at run time.
2. Evolution qualifies, such as extensibility and scalability embody in the static structure of the software system.

The Non-Functional requirements of our project are:

* **UTILITY**
  + 1. The system must be easy to use by both admin and users such that they do not need to read an extensive number of manuals.
    2. The system must be quickly accessible by both admin and users.

iii)The system must be intuitive and simple in the way it displays all relevant data and relationships.

* **RELIABILITY**
  + 1. The system must give accurate inventory status to the user continuously. Any inaccuracies are taken care by the regular confirming of the actual levels with the levels displayed in the system.
    2. The system must provide a password enabled login to the user to avoid any foreign entity changing the data in the system.
    3. The system should provide the user updates on completion of requested processes and if the requested processes fail, it should provide the users the reasons for the failure.
    4. The system should not update the data in any database for any failed processes.
    5. The system must not lag, because the workers using it don’t have down time to wait for it to complete an action.
    6. All the functions of the system must be available to the user every time the system is turned on.

(vi) The calculations performed by the system must comply according to the norms set by the user and should not vary unless explicitly changed by the user.

* **SUPPORTABILITY**
  + 1. The software is designed such that it works even on systems having the minimum configuration.
    2. The system is adaptable even if additional plugins or modules are added at a later point.
    3. The data can be exported to the admin so as to make the system more portable.
* **INTERFACING**
  + 1. The system must offer an easy and simple way of viewing the current inventory.
    2. The system must be able to display the destination packages.
* **STANDARDS**
  + 1. The coding standards and naming conventions will be as per the American standards.
* **DESIGN CONSTRAINT**
  + 1. The design constraints are that the browser at each place may not follow similar screen resolutions. This can lead to the website not having the impact it is planned to have.
* **STANDARD DEVELOPMENT TOOLS**
  + 1. The System shall be built using a standard web page development tool that confirms to either IBM’s CUA standards or Microsoft’s GUI standards.
* **PORTABILITY**

1. The application should be portable on any windows-based system.
2. **FUNCTIONAL REQUIREMENTS:**

In software engineering, a functional requirement defines a function of a software-system or component. A function is described as a set of inputs, the behavior and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that show how a use case to be fulfilled. Typically, a requirements analyst generates functional requirements after building use cases. However, this may have exceptions since software development is an iterative process and sometime certain requirements are conceived prior to the definition of the use case. Both artifacts (use cases documents and requirements documents) complement each other in a bidirectional process. A typical functional requirement will contain a unique name and number, a brief summary, and a rationale. This information is used to help the reader understand why the requirement is needed, and to track the requirement through the development of the system.

 The core of the requirement is the description of the required behavior, which must be a clear and readable description of the required behavior. This behavior may come from organizational or business rule, or it may be discovered through elicitation sessions with users, stakeholders and other experts within the organization. Software requirements must be clear, correct unambiguous, specific and verifiable.

* + - * 1. **RESERVATION OF TICKET:**
* REQUEST TIME TABLE:

Passenger requests database to display railway timetable.

* DISPLAY TIMETABLE:

Database displays timetable to the customer.

* REQUEST TO RESERVE TICKET:

Passenger requests the clerk to reserve his/her ticket.

* INPUT DETAILS:

Clerk asks customer to enter details for the reservation of ticket.

* CALCULATE FARE:

Clerk calculates the total fare of the journey according to the number of passengers and tells the customer.

* RESERVE TICKET:

Ticket is reserved and customer pays the fare.

* REQUEST TO PRINT:

Customer requests to print the ticket.

* PRINT TICKET:

Ticket is being printed and handled to the customer.

* + - * 1. **CANCELLATION OF TICKET:**
* REQUEST TO CANCEL TICKET:

Customer requests to cancel his/her ticket for which he/she has to give the PNR number.

* CANCEL TICKET:

Once the PNR number is received, the ticket is cancelled.

1. **DOCUMENT APPROVAL:**

The bill passed on any proposals related to railway management needs approval of Ministry of Railway Department.

**E-R DIAGRAM**

**DATA FLOW DIAGRAM 0 LEVEL DFD**

**RESERVE**

**ENQUIRY OFFICER**

**ENQUIRY**

**RESERVATION OFFICER**

**CUSTOMER**

ENQUIRY

OFFICER

Answer to enquiry

Enquiry

Request for enquiry

Payment

Request for Form

Application Form

Rejection

Conformation

Payment Received

Submit Form

Confirmation

Rejection

Check for Status

Request for Reservation

Issue Form

RESERVATION

OFFICER

COUSTMER

Ticket Printing

Train Timing and availability

ReservationRecord

Ticket to Costumer

Check for Status

Rejection

Conformation

Payment Receive

Issue Ticket

Check Status

Update

Record

Answer to Enquiry

Request for enquiry

Payment

COSTUMER

RESERVATION

OFFICER

ENQUIRY

OFFICER

Confirm

Enquiry

Ticket Received

Submit Form

Issue Form

Request for Form

Conformation

Rejection

Application Form

**DATA FLOW DIAGRAM LEVEL 1**

**CLASS DIAGRAM**

|  |
| --- |
| **Customer** |
| - age : integer array  - tr\_no : integer  - tr\_nm : string  - name : string array  - sex : char array  - noseat : integer  - fare : integer  - mode : integer  - place\_s : string  - place\_d : integer  - timea : integer  - timed : integer |
| + input( )  + disp ( )  + cancel() |
| Class to input details of customers |

|  |
| --- |
| **Train** |
| - t\_no : integer  - t\_name : string  - place\_a : string  - place\_d : string  - timea : integer  - timed : integer  - noac : integer  - nonac : integer  - nog : integer  - fareac : integer  - farenac : integer  - fareg : integer |
| + input( )  + modify( )  + disp( )  + disptt( )  + rett\_no( )  + ret\_sorce( )  + ret\_dest( ) |
| Class to input and modify train details |

|  |
| --- |
| **Admin** |
|  |
| + newdatabase( )  + update( )  + updatedata( )  + modify( ) |
| Class to add and update database of train |

|  |
| --- |
| **Ticket** |
|  |
| + enquiry( )  + reserve( )  + cancel( )  + print()  + modify( ) |
| Class to print, reserve and modify tickets |

**SEQUENCE DIAGRAM**

10. Cancel Ticket

2. Display timetable

9. Request to cancel ticket

7. Request to print ticket

8. Print Ticket

6. Reserve ticket

5. Calculate fare

4. Input details

3. Request to reserve ()

1. Request timetable

**TICKET**

**CLERK**

**RAILWAY DATABASE**

**PASSENGER**

**DATA DICTIONARY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **ALIAS** | **USE** | **CONTENT** | **ADDITIONAL INFORMATION** |
| PNR Number | None | Enquiry  Reservation  Cancellation | PNR=pnrg() | None |

**Cost Estimation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **COST ESTIMATION** | | | | |
|  | **PRODUCT** | **PURPOSE** | **UNIT COST (₹)** | **NO. OF UNITS** | **TOTAL COST (₹)** |
|  |  |  |  |  |  |
| **SOFTWARE** | **WINDOWS 10** | OPERATING SYSTEM | 16835.00 | 4 | 67340.00 |
| **OFFICE 365** | OFFICE TOOLS | 8999.00 | 1 | 8999.00 |
| **PROJECT PROFESSION** | PROJECT MANAGMENT TOOL | 89000.00 | 4 | 356000.00 |
| **LINUX HOSTING** | SERVER | 12559.00 | 1 | 12559.00 |
| **WEBSITE BACKUP** | BACKUP | 3924.00 | 1 | 3924.00 |
| **WEBSITE SECURITY ESSENTIAL** | SECURITY | 12845.35 | 1 | 12845.35 |
| **CLion** | IDE | 42064.50 | 4 | 168257.80 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **HARDWARE** | **LENOVO IDEAPAD 330** | SYSTEM | 70890.00 | 4 | 283560.00 |
|  |  |  |  |  |  |
| **EMPLOYEE WAGE** | **WAGE FOR 4 MONTHS** | SALARY | 82500 | 4 | 330000 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | **TOTAL AMOUNT** |  |  |  | 1243485.15 |

**REFERENCES AND BIBLIOGRAPHY**

* **BOOKS USED:**
* Object Oriented Programming C++ (E. Balaguruswami)
* Introducing C++ (Sumita Arora)
* Software Engineering (Pressman)
* **SITES USED :**
* [www.scribd.com](http://www.scribd.com)
* [www.irctc.com](http://www.irctc.com)
* [www.indianrail.com](http://www.indianrail.com)
* [www.wikipedia.org](http://www.wikipedia.org)
* [www.yatra.com](http://www.yatra.com)
* [www.trainenquiry.com](http://www.trainenquiry.com)

**SRS**

**On**

**Railway Reservation System**

**Submitted by**

* Adarsh Chaudhary 12000216059

**SRS**

**On**

**Railway Reservation System**

**Submitted by**

* Ashutosh Jha 12000216046

**SRS**

**On**

**Railway Reservation System**

**Submitted by**

* Ayushi Jha 12000216044

**SRS**

**On**

**Railway Reservation System**

**Submitted by**

* Disha Sinha 12000216040