

RAILWAY MANAGEMENT SYSTEM

Project report submitted to the Amrita Vishwa Vidyapeetham in partial fulfilment of the
requirement for the Degree of
B.Tech. Computer Science and Engineering
Artificial Intelligence



Submitted by:

Abhinav Pandey (AM.EN.U4AIE21088)
Abhishek. A. (AM.EN.U4AIE21003)
Anfas Hassan V (AM.EN.U4AIE21014)
Aravind MJ (AM.EN.U4AIE21017)
Rithesh R (AM.EN.U4AIE21054)
D.S.S. Sandeep Chandra (AM.EN.U4AIE21079)

JULY 2022

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING AMRITA
VISHWA VIDYAPEETHAM
(Estd. U/S 3 of the UGC Act 1956)
Amritapuri Campus Kollam -690525**



BONAFIDE CERTIFICATE

Your Guides

Coordinator name

Project Guide

Project Coordinator

Reviewer

Chairperson

Dept. of Computer Science & Engineering

Place: Amritapuri

Date: 11 July 2022

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AMRITA VISHWA
VIDYAPEETHAM**

**(Estd. U/S 3 of the UGC Act 1956)
Amritapuri Campus Kollam -690525**

DECLARATION

We, Abhinav Pandey, Abishek A., Anfas Hassan, Aravind MJ, Rithesh R, D.S.S.

Sandeep Chandra hereby declare that this project entitled **Railway Management System** is a record of the original work done by us under the guidance of **Smt Akshara P Byju** Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, that this work has not formed the basis for any degree/diploma/associations/fellowship or similar awards to any candidate in any university to the best of our knowledge.

Place: Amritapuri Date: 12 July 2022

Signature of the student

Signature of the Project Guide

ACKNOWLEDGMENTS

Humble pranams at the lotus feet of Amma, Sri Mata
Amritanandamayī devi

We thank Akshara P Byju, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, for her timely advice and constructive suggestions. We thank the faculty of the CSE Department for giving us an opportunity to explore our capabilities and widen our knowledge. We also thank our team members for their cooperation and hard work put into this project. We also thank our batch mates for their support and our parents for their support and encouragement.

ABSTRACT

The Indian Railways (IR) carries about 5.5 lakhs passengers in reserved accommodation every day. The Computerised Passenger Reservation System (PRS) facilitates the booking and cancellation of tickets from any of the 4000 terminals (i.e. PRS booking window all over the countries). These tickets can be booked or cancelled for journeys commencing in any part of India and ending in any other part, with travel time as long as 72hours and distance up to several thousand kilometres. In the given project we will be developing a desktop app which will help users to find train details, book and exact rates of their tickets to the desired destination. It is a combined app for both railway officials and passengers. The railway officials can control the flow of trains in the station and display them on screen. It is a complete railway management app that uses many data structures such as arrays, linked list and queue.

CONTENTS

S.N.	List of Figures	Page no
1	Introduction	7
2	Feasibility Study	9
3	System Requirement	15
4	System Environment	17
5	Conclusion	18
6	Source Code	18

INTRODUCTION

Every Indian can proudly say that the Indian Railway is the world's second largest railway system. If going by figures it covers 6853 stations and 63028 kilometres of track distance, carrying a load of 37840 passengers and around 500 tons of freight annually. With this huge magnitude our railway system faced a problem of a complete and fast reservation system and was formed around the objective of providing reserved reservation to the train travellers.

We are introducing the reserved ticketing in terminals and interactive live report of the train arrival response systems on touch screens. In the given project we will be developing a desktop app which will help users to find train details, book and exact rates of their tickets to the desired destination. It is a combined app for both railway officials and passengers. The railway officials can control the flow of trains in the station and display them on screen. It is a complete railway management app that uses many data structures such as arrays, linked list and queue.

1.1 Product Scope:

- Access to train timings
- Online ticket booking
- Manage incoming or late trains

1.2 Objectives:

- The main aim of this system is to manage the passenger ticket booking as well as the arrangement of train arrival timings.
- The train timings as we see on the screen in a railway station will be seen on the app in the passenger portal.
- On the other hand the railway officials get the train timings and display it on the screen.

- If one train is late the official will manage the timings and change it on the screen.
- The final product is a system that has two portals for passengers and railway officials, one shared screen between two and use of data structures such as linked lists, queue, arrays, priority queue, etc.

FEASIBILITY STUDY

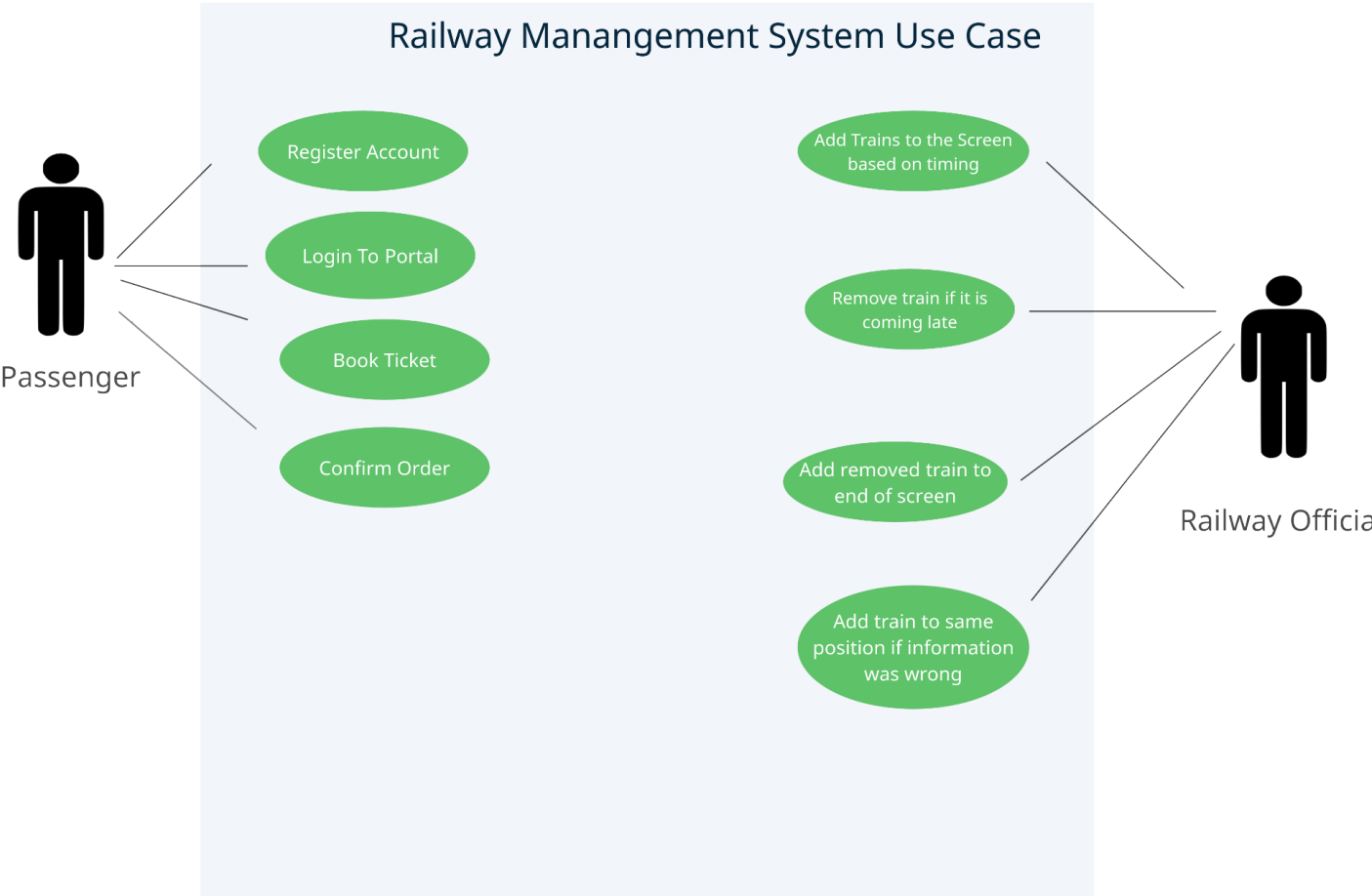
Product Features:

- Passenger login/register
(Username, Password)
- Train booking
 1. Form
 - ❖ Name
 - ❖ Date
 - ❖ From/to
 - ❖ No of tickets
 - ❖ Train Name
 - ❖ Price
 2. Train Screen
(See the train timings)
- Officials login
(Username, Password)
 - ❖ Take train and timing as input
 - ❖ Input in Train queue
 - ❖ Remove from queue if train is late and add in new queue
 - ❖ Add to main train queue if the late train is now on time

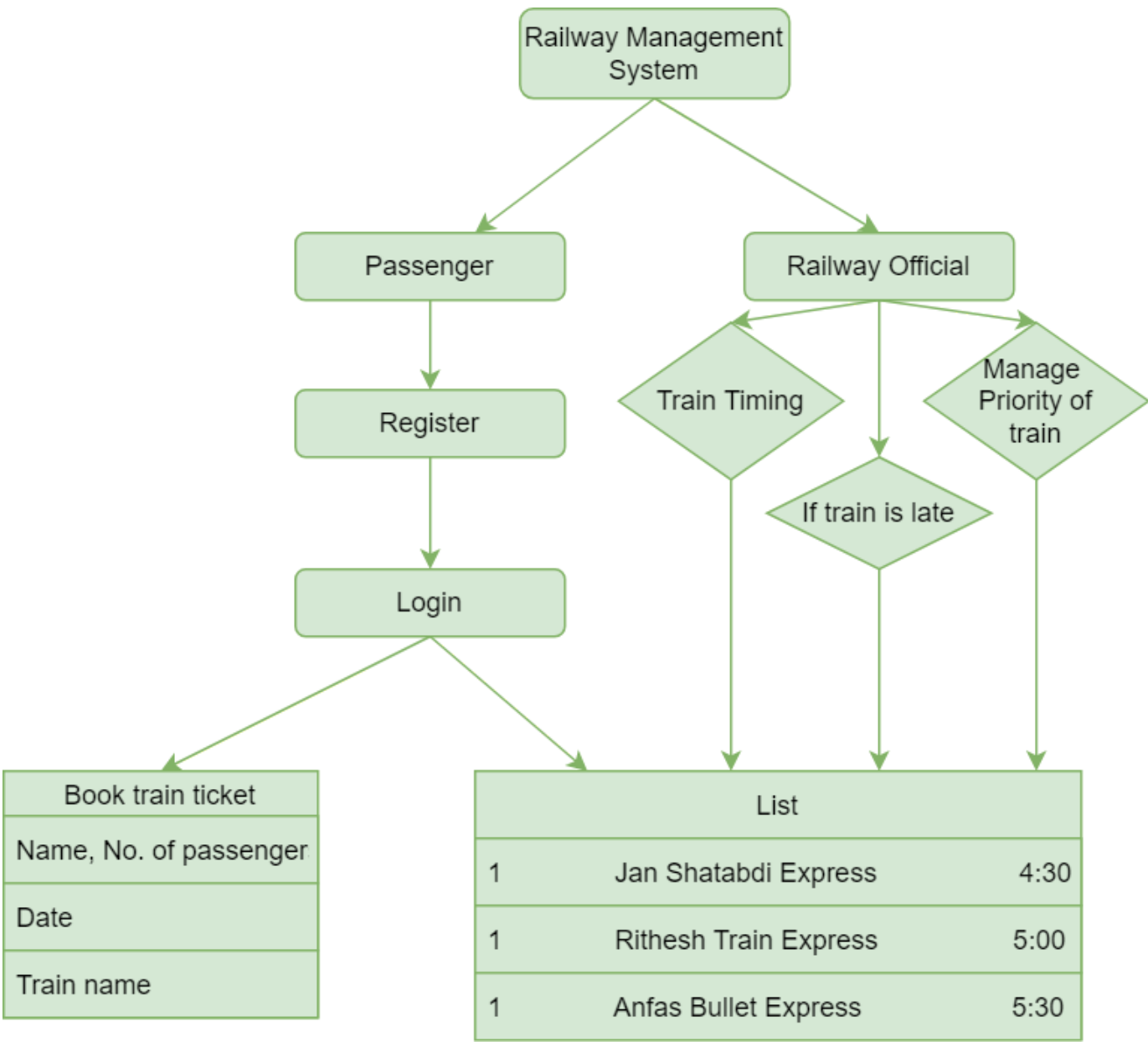
Use cases:

Actors	Use Case
Passenger	Can register, book a train ticket and get the ticket at last. They can also see the trains which are currently coming to the train station.
Railway Official	Can get the information about incoming trains. They can remove the train from the train screen if the train is late and put it back to the screen if the information was wrong.

Use case diagram:



Design and Implementation:



User-Interface:

(OFFICIAL PORTAL)

MANAGEMENT

ADD TRAIN

NAME

TIME

...

REMOVE

ADD

LATE QUEUE

1

2

3

DELAY

ADD BACK

TRAIN SCREEN

1 Jan-Shatabdi-Express 2:40

2 Abhinav-Orient-Express 2:40

3 Jan-Shatabdi-Express 3:20

4 Abishek-Coastline 3:30

5 Rithesh-Central-Express 3:40

(TRAIN SCREEN)

S.N.	TRAIN	TIME
1	Jan-Shatabdi-Express	2:40
2	Abhinav-Orient-Express	2:40
3	Jan-Shatabdi-Express	3:20
4	Abishek-Coastline	3:30
5	Rithesh-Central-Express	3:40

(LOGIN PAGE)

LOGIN

USERNAME

PASSWORD

LOGIN

If you don't have an account

Register Now

(REGISTER PAGE)

REGISTER

NAME

USERNAME

PASSWORD

REGISTER

(TICKET BOOKING)

BOOK A TICKET

<

FROM

...

TO

...

DATE

NAME

NO OF TICKETS

0

TRAIN NAME

...

PRICE

BOOK

SYSTEM REQUIREMENT

Data Structures Used:

Linked Lists: (Train Screen)

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers.

We have implemented linked lists in our program in the main train screen. Every train is a line one after another, we represent every train as a node in the linked list.

We have used different methods like:

- deleteFromBeginning - When a train leaves the station
- insertFromBack - When a train is coming to the station
- deleteFromMiddle - When a train is coming late
- insertAtMiddle - If a train was removed from the list but the information was wrong we can add back at its same position

Queue: (Store late coming trains)

A Queue is a linear structure which follows a particular order in which the operations are performed. We can only insert from the back and remove from the top.

We have implemented a queue to store late coming trains. If a train is coming late, we remove it from the main screen and put it in the queue. If another train follows it, that train will also be added to the queue. And when it is coming to the station, we remove it from the queue and insert it in the linked list.

We have use different methods like:

- Enqueue

- Dequeue

Arrays: (Login and register information)

An array is a collection of items stored at contiguous memory locations. We have used 2D-arrays for storing the username and password of both the passengers and officials.

SYSTEM ENVIRONMENT

(INTERFACE)

- Passenger needs to register with the system before using the portal
- Passenger need to login to with registered id to access the portal
- Passenger should input all the details like name, date, train to book a ticket
- Railway officials need the input the train and timing in the train screen
- Railway officials get the information if a train is late and remove from the screen
- Railway officials can put the train back in its original timing if information is wrong

(SOFTWARE)

- JDK 8 or higher
- IDE for java
- Java Swing GUI Library
- Windows Builder Editor

(HARDWARE)

- Hard Disk 20GB
- Memory 2GB of RAM
- Processor Intel® i5
- CPU 2GHz

CONCLUSION

6.1 Summary of Achievements

The following achievements were made during the project:

- Learned about implementing data structures in real life project
- Learned to handle csv files
- Learned to manage and use Swing library
- Learning about the different phases of software development and the software engineering processes involved in the development of software

6.2 Main difficulties encountered

- Handling csv files
- Converting stored files into data structures and passing it
- Time constraint

6.3 Limitations of the Project

- Print ticket option is not provided here
- Database could have been used

SOURCE CODE

Github link:

<https://github.com/Abhinavxox/Railway-Management-System>