



**TRAINING REPORT**

**OF**

**CORE JAVA**

**AT**

**“INFOWIZ Software Solution, Bathinda”**

**ON**

**“RAILWAY RESERVATION SYSTEM”**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE**

**OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**Computer Science and Engineering**

**Submitted By:**

**Name: Yash Kumar**

**Roll No.: 12201211**

---

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**PUNJABI UNIVERSITY**  
**PATIALA-147002**

# CERTIFICATE

Certificate of Training



**INFOWIZ**<sup>®</sup>  
A SOFTWARE SOLUTION



H.O.: SCO 118 - 120, Sector 34 - A, **CHANDIGARH**  
B.O.: First Floor, Crown Tower, 100 Ft. Road, **BATHINDA**  
Web.: [www.infowiz.co.in](http://www.infowiz.co.in) E-mail : [info@infowiz.co.in](mailto:info@infowiz.co.in)

## Certificate

No. INFOWIZ/602024/334

This is certified that Mr./Ms. YASH KUMAR S/D/o. Sh. VIDAY KUMAR  
of U.C.E, Pbi, Uni, Patiala has successfully undergone Training Course CORE JAVA  
From 10<sup>th</sup> JUNE 2024 to 25<sup>th</sup> JULY 2024. During the tenure of the above course, we found him/her  
a hardworking & innovative individual.  
We wish him/her a very bright and prosperous future.



Technical Head





Managing Director

Chandigarh : 0172 4567888, 98885 00888, 98886 00888      Bathinda : 0164 5007088, 90235 00888, 90236 00888

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
PUNJABI UNIVERSITY, PATIALA**

**CANDIDATE'S DECLARATION**

I “YASH KUMAR” hereby declare that I have undertaken Summer training at “INFOWIZ SOFTWARE SOLUTIONS,BATHINA” during a period from 10 JUNE 2024 to 25 JULY 2024 Partial fulfillment of requirements for the award of degree of B. Tech (Department of Computer Science &Engineering) at Punjabi University Patiala. The work which is being presented in the training report submitted to Department of Computer Science & Engineering) at Punjabi University, Patiala is an authentic record of training work

Signature of the Student

The summer training Viva–Voce Examination of\_\_\_\_\_has been

Held on\_\_\_\_\_and accepted.

Signature of Examiner

## **ABSTRACT**

This project report presents the development of a Railway Reservation System using Core Java. The system is designed to automate the processes involved in booking train tickets, managing train schedules, and handling customer data. The primary objective of the project is to create a user-friendly and efficient platform for railway reservations, minimizing the manual effort and errors associated with traditional booking systems.

The Railway Reservation System allows users to search for available trains, check seat availability, book tickets, and cancel reservations. The system also includes an administrative module for managing train schedules, fares, and passenger records. Core Java concepts such as object-oriented programming, file handling, exception handling, and collections have been extensively used to ensure robust functionality.

Key features of the project include dynamic train searching, real-time seat allocation, ticket cancellation, and the generation of PNR numbers for tracking. The system also maintains a record of all reservations and cancellations for audit and customer service purposes.

This project demonstrates the application of Java programming principles to solve real-world problems in the transportation domain. The system is designed with scalability and maintainability in mind, ensuring that it can be extended to include additional features such as payment integration and mobile accessibility in the future.

## ACKNOWLEDGEMENT

I would like to express my sincere appreciation to **Ms. Poonam** for their invaluable guidance and support throughout my summer training project on the Railway Reservation System. Their expert advice and constructive feedback were crucial in the successful development and completion of this project.

I am also grateful to **INFOWIZ Software Solution**, Bathinda for providing the opportunity to work on this project and for offering the resources and environment necessary for its success. The practical experience gained during this training has been immensely beneficial.

My thanks extend to my colleagues and fellow trainees for their collaboration and encouragement. Their insights and shared knowledge significantly contributed to the quality of the project.

Lastly, I would like to acknowledge the support and understanding of my family and friends throughout this period. Their encouragement was a constant source of motivation.

Name :- YASH KUMAR

**INFOWIZ** is leading strategic IT Company offering integrated IT solution. INFOWIZ is having rich experience managing global clients across various business verticals and align IT strategies to achieve business goals. The various accreditations that we achieved for every service, we offer reflect our commitment towards the quality assurance.

**INFOWIZ** is a 8 years young organization which has won the **NATIONAL AWARD** for **2 consecutive years 2014-2015 & 2015-16** for **BEST Industrial Training** from **Hon` able GOVERNOR of Punjab & Haryana Sh. Kaptan Singh Solanki**. He is also the **Chancellor of PTU & Punjabi University**. INFOWIZ is a member of Confederation of Indian Industry ( **CII membership number – N4654P** ) & also with an **ISO Certification**. We have a global foot prints in providing the off shore companies of US, UK, France, Ireland, Canada and Australia with quality and timely Web and SEO services.

INFOWIZ is an organization which is established in the field of Web Development (PHP & .NET), JAVA (Core as well as Advance), I-phone & Android Applications, Embedded systems (AVR, PIC & ARM),Automation, ROBOTICS, Networking (MCSE, CCNA & RHSE) & in Mechanical.

Our skilled team of professionals make sure that the product is developed as per the customer's needs and keeping the customer informed about the development of their project from time to time. We do not only emphasize on formulating an attractive solution to our clients but also believe in providing a workable solution. INFOWIZ offers research based Search Engine Marketing products that help achieve greater insights to customer's online business. Our Research & Development arm offers SEO tools for SEM professionals.

INFOWIZ also provides **Technical Support & Consultancy** to Software Companies like JIA Group New Zealand, Sagitech solutions Panchkula, Jarc Infotech Mohali, Infonet Solution, Delhi etc.

## **COURSES Offered :-**

### **For CSE/IT/MCA Professionals:-**

- 1) Web Development in PHP with LIVE Projects
- 2) Web Development in .NET with LIVE Projects
- 3) JAVA (Core as well as Advance ) with LIVE Projects
- 4) Android Applications with LIVE Projects
- 5) Web Designing (Photoshop, Coral Draw)

### **For ECE/EE/EIE/ME/CIVIL Professionals:-**

- 1) Robotics With Live Project
- 2) VLSI-VHDL with Live Project
- 3) Embedded System Design with Live Project
- 4) Microcontroller with Live Project
- 5) Microprocessor with Live Project

Our core strength is our timely, technically and cost effective project delivery. We also provide customers with designs as per their demands. INFOWIZ also provide **JOB Oriented Industrial.**

## LIST OF FIGURES

<b>Figure No.</b>	<b>Name</b>	<b>Page No.</b>
Figure 1.1	Features of JAVA	15
Figure 1.2	Intel Core Processor	21
Figure 1.3	RAM	21
Figure 1.4	Hard Disk	22
Figure 1.5	SDLC	23
Figure 1.5	Hierarchy Of Java AWT	30
Figure 1.5	Hierarchy Of Java Swing	33
Figure 3.1	Welcome Screen	52
Figure 3.2	Login Page	53
Figure 3.3	Home Page	54
Figure 3.4	Reservation Form	55
Figure 3.5	Passenger Form	56
Figure 3.6	Enquiry Form	57
Figure 3.7	Cancellation Form	58
Figure 3.8	Database	59



## LIST OF TABLES

<b>Table No.</b>	<b>Name</b>	<b>Page No.</b>
Table 1.1	Issues during technical Feasibility	25
Table 1.2	Potential Cost and Benefit of Software Project	26
Table 1.3	Issues during operational feasibility	27
Table 1.4	Components Of Java Swings	34-35

# Contents

<i>certificate</i> .....	2
<i>Candidate's Declaration</i> .....	3
<i>Abstract</i> .....	4
<i>Acknowledgement</i> .....	5
<i>Company Profile</i> .....	6
<i>List Of Figures</i> .....	8
<i>List Of Tables</i> .....	9
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>11</b>
1.1 INTRODUCTION TO PROJECT .....	11
1.2 INTRODUCTION TO JAVA .....	12
1.3 SYSTEM ANALYSIS .....	20
1.4 SOFTWARE TOOL USED FOR JAVA DEVELOPMENT:.....	27
1.5 JAVA AWT(Abstract Window Toolkit) .....	28
1.6 JAVA SWINGS.....	31
<b>CHAPTER 2 TRAINING WORK UNDERTAKEN</b> .....	<b>36</b>
2.1 SYSTEM DESIGN.....	36
2.2 INPUT DESIGN(Introduction) .....	39
2.3 OUTPUT DESIGN (Introduction) .....	41
2.4 Java: .....	42
<b>CHAPTER 3 RESULTS AND DISCUSSION</b> .....	<b>46</b>
3.1 TESTING .....	46
3.2 RESULTS .....	47
3.3 DISCUSSION .....	48
<b>CHAPTER 4 CONCLUSION AND FUTURE SCOPE</b> .....	<b>60</b>
4.1 FUTURE SCOPE .....	60
4.2 CONCLUSION .....	61
<b>REFERENCES</b> .....	<b>62</b>

# **CHAPTER 1 INTRODUCTION**

## **1.1 INTRODUCTION TO PROJECT**

Through this system, a practical approach has been taken under study to understand how online reservation system works. This system will include all the necessary fields which are required during online reservation system. This system will be easy to use and can be used by any person. The basic idea behind this project is to save data in a central database which can be accessed by any authorize person to get information and saves time and burden which are being faced by their customers.

### **Existing System:-**

Even though the online reservation system is available but the passengers have to fill all their necessary details using pen and paper which involves manual working. It's the main counter is not open due to late arrival or due to some reasons then in this case, passengers have to wait and in certain cases they have to wait in line, waiting for their chance to come. After wasting so much time, customers able to get their answers and sometimes they did not positive response.

### **Proposed System:-**

Through this system customers do not have to wait in line and they will able to get their answers in just a click. Through this system online form will be available by which passengers will able to fill their details along with their journey details. To book their seats, users will have to first search their trains as per their requirements and after getting correct train they will have to provide Train number. After pressing next button, it will show its status, whether it has been activated or in process. Online cancellation form will also be available and to cancel their reservation, they will have to provide their PNR number again to carry the next task.

### **Number of Modules:-**

1. Login Form.
2. Reservation Form.

3. Cancellation Form.
4. PNR Enquiry.
5. Exit.

## **1.2 INTRODUCTION TO JAVA**

- Java is a **programming language** and a **platform**.
- Java is a high level, robust, secured and object-oriented programming language.
- Java supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL)
- Java is open source Software.

### **JAVA HISTORY:**

- Java is a general-purpose, object-oriented programming language developed by Sun Microsystems of USA in 1991. Originally called Oak by James Gosling (one of the inventor of the language). Java was invented for the development of software for consumer electronic devices like TVs etc. The main aim had to make java simple, portable and reliable.
- Java Authors: James , Arthur Van , and others

### **Following data shows the year and beginning of Java**

#### **1990**

- Sun decided to developed software that could be used for electronic devices. And the project called as Green Project head by James Gosling.

**1991**

- Announcement of a new language named —Oak

**1992**

- The team verified the application of their new language to manage a list of home appliances using a hand held device.

**1993**

- The World Wide Web appeared on the Internet and transformed the text-based interface to a graphical rich environment.

**1994**

- The team developed a new Web browser called —Hot Java to locate and run Applets.

**1995**

- Oak was renamed to Java, as it did not survive —legal registration. Many companies such as Netscape and Microsoft announced their support for Java.

**1996**

- Java language is now famous for Internet programming as well as a general purpose OO language.

**1997**

- Sun releases Java Development Kit(JDK 1.1)

**1998**

- Sun releases Software Development Kit (SDK 1.2)

**1999**

- Sun releases Java 2 platform Standard Edition (J2SE) and Enterprise Edition(J2EE).

**2000**

- J2SE with SDK 1.3 was released.

**2002**

- J2SE with SDK 1.4 was released.

**2004**

- J2SE with JDK 5.0 was released.

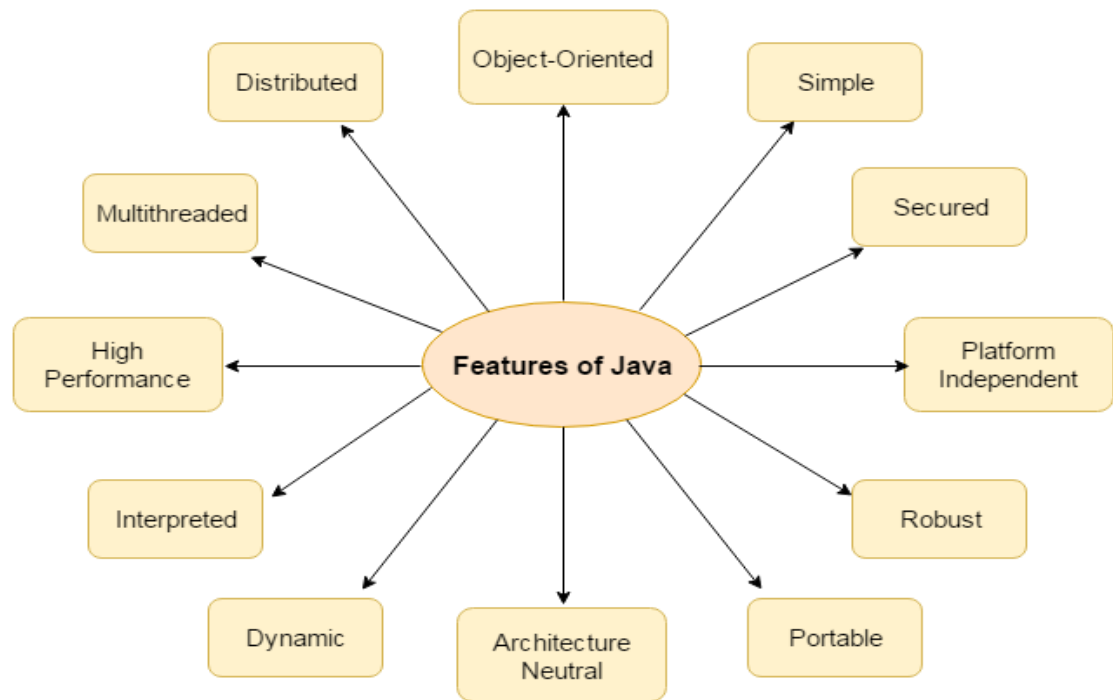
**USE JAVA:**

- Desktop Applications such as acrobat reader, media player, antivirus etc.
- Web Applications such as irctc.co.in, javatpoint.com etc.
- Enterprise Applications such as banking applications.
- Mobile
- Embedded System
- Smart Card
- Robotics
- Games etc.

**FEATURE OF JAVA:**

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Robust
- Architecture neutral
- Dynamic

- Interpreted
- High Performance
- Multithreaded
- Distributed



*Figure 1.1: Features of Java*

### **JAVA VIRTUAL MACHINE:**

As we know that all programming language compilers convert the source code to machine code. Same job done by Java Compiler to run a Java program, but the difference is that Java compiler convert the source code into Intermediate code is called as bytecode. This machine is called the Java Virtual machine and it exists only inside the computer memory.

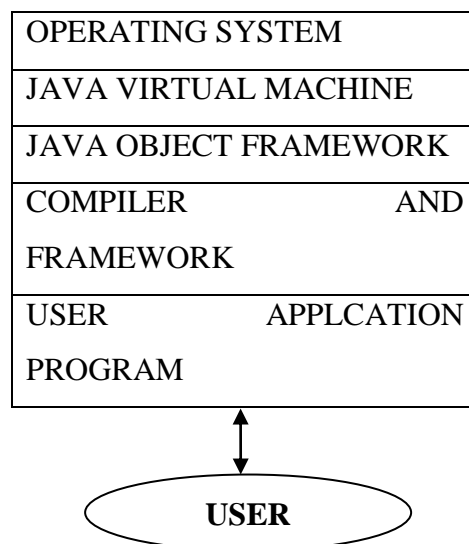
**Following figure shows the process of compilation:**



- The Virtual machine code is not machine specific. The machine specific code is generated. By Java interpreter by acting as an intermediary between the virtual machine and real machines shown below



- Java Object Framework act as the intermediary between the user programs and the virtual machine which in turn act as the intermediary between the operating system and the Java Object Framework.



### **JAVA ENVIRONMENT:**

- Java environment includes a number of development tools, classes and methods. The development tools are part of the system known as Java Development Kit (JDK) and the classes and methods are part of the Java Standard Library (JSL), also known as the Application Programming Interface (API).



- Java Development kit (JDK) – The JDK comes with a set of tools that are used for developing and running Java program. It includes:

1. Appletviewer ( It is used for viewing the applet)
2. Javac (It is a Java Compiler)
3. Java (It is a java interpreter)
4. Javap (Java diassembler,which convert byte code into program description)
5. Javah (It is for java C header files)
6. Javadoc (It is for creating HTML document)
7. Jdb (It is Java debugger)

- For compiling and running the program we have to use following commands:

**a) javac (Java compiler)**

- In java, we can use any text editor for writing program and then save that program with “.java” extension. Java compiler convert the source code or program in bytecode and interpreter convert “.java” file in “.class” file.

**Syntax:**

C:\javac filename.java

If my filename is “abc.java” then the syntax will be

C:\javac abc.java

**b) java(Java Interpreter)**

- As we learn that, we can use any text editor for writing program and then save that program with “.java” extension. Java compiler convert the source code or program in bytecode and interpreter convert “.java” file in “.class” file.

**Syntax:**

C:\java filename

If my filename is abc.java then the syntax will be

C:\java abc

**Simple Java Program:**

```
class FirstProgram
{
    public static void main(String args[])
    {
        System.out.println("This is my first program");
    }
}
```

- The file must be named “FirstProgram.java” to equivalent the class name containing the main method.
- Java is case sensitive. This program defines a class called “FirstProgram”.
- A class is an object oriented term. It is designed to perform a specific task. A Java class is defined by its class name, an open curly brace, a list of methods and fields, and a close curly brace.
- The name of the class is made of alphabetical characters and digits without spaces, the first character must be alphabetical.
- The line “public static void main (String [] args )” shows where the program will start running.
- The word main means that this is the main method –The JVM starts running any program by executing this method first.
- The main method in “FirstProgram.java” consists of a single statement
- System.out.println("This is my first program");
- The statement outputs the character between quotes to the console.

- Above explanation is about how to write program and now we have to learn where to write program and how to compile and run the program.
- For this reason, the next explanation is showing the steps:

1. Edit the program by the use of Notepad.
2. Save the program to the hard disk.
3. Compile the program with the javac command.(Java compiler)
4. If there are syntax errors, go back to Notepad and edit the program.
5. Run the program with the java command.(Java Interpreter)
6. If it does not run correctly, go back to Notepad and edit the program.
7. When it shows result then stop

## TYPES OF JAVA APPLICATION

- **Standalone Application :**It is also known as desktop application or window-based application.

An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

- **Web Application :**An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.
- **Enterprise Application :** An application that is distributed in nature, such as banking applications etc.
- It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.
- **Mobile Application :**An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

## **JAVA PLATFORMS / EDITIONS**

- **Java SE (Java Standard Edition)** :It is a java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection etc.
- **Java EE (Java Enterprise Edition)** : It is an enterprise platform which is mainly used to develop web and enterprise applications. It is built on the top of Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA etc.
- **Java ME (Java Micro Edition)** :It is a micro platform which is mainly used to develop mobile applications.
- **JavaFx** :It is used to develop rich internet applications. It uses light-weight user interface API.

## **1.3 SYSTEM ANALYSIS**

### **Software requirement analysis:**

The requirements gathering process was intensified and focused specially on software. To understand the nature of the problem to be built, one must understand the information domain for the software, as well as required function. Behavior, performance and interface requirements of both the system and the software were documented and received with the company.

### **Design:**

Software design is actually a multi – step process that mainly focuses on four distinct attributes of a program. Data structures, software architecture, interface representation and procedural detail. The design process translates requirements into a design document and becomes a part of the software configuration.

### **Code generation:**

In this process design is translated into machine – readable form. As the design process was performed in a detailed manner code generation was accomplished easily.

**Testing:**

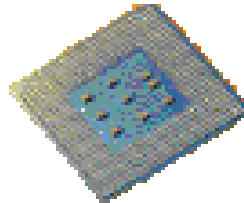
Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all functional externals are tested to uncover errors and ensure that defined input will produce actual result that agree with required results

**Software and Hardware Requirement Specifications**

The softwares which are used to develop this system are quite familiar and not new and complex to the user and are also easily available, and the software has been planned in such a way that it can be implemented on any operating system and can easily interact with any of the databases which have been provided in the software.

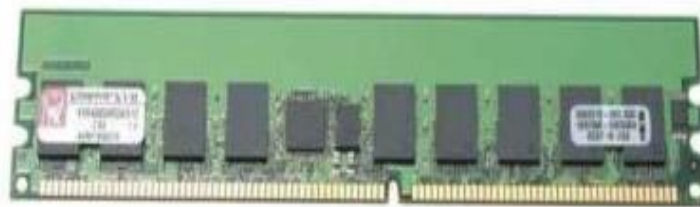
**HARDWARE:**

- INTEL CORE PROCESSOR



*Figure 1.2*

- 1GB RAM



*Figure 1.3*

- 80 Gb disk storage at the server end



*Figure 1.4*

## **SOFTWARE:**

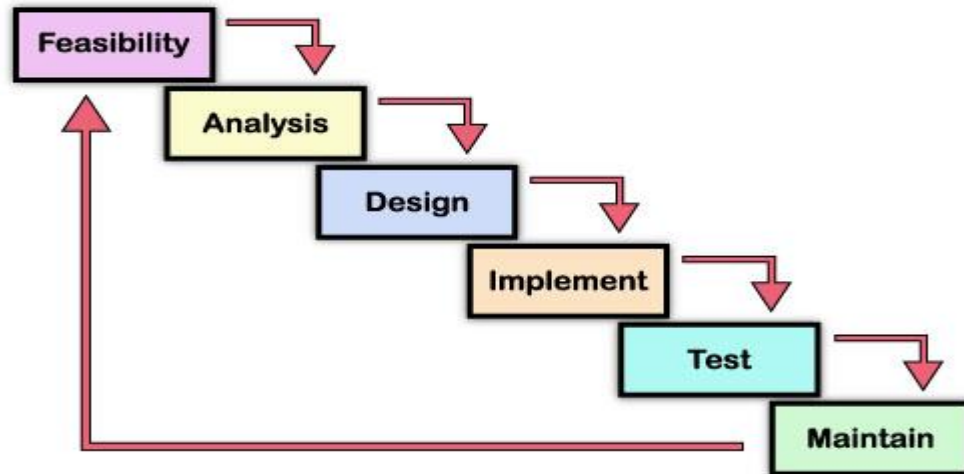
- Language: Java 21.0.2
- Database: Ms-SQL Server 2022 Express Edition, My-SQL Server 8.0
- O.S. : Platform-Independent (Windows, Linux, Solaris)
- JDK 21 - ECLIPSE
- Database - MySQL Database Server 5.0
- Designing Tool - SWING, AWT

Therefore it is quite clear that, that software is user-friendly. The language we used is interesting and its presentation is very attractive and interesting and also user will enjoy using this new system.

### **System Requirement Analysis**

Prior to the software development efforts in any type of system it is very essential to understand the requirements of the system and users. A complete specification of the software is the 1<sup>st</sup> step in the analysis of system. Requirements analysis provides the designer with the representation of function and procedures that can be translated into data, architecture and procedural design.

The goal of requirement analysis is to find out how current system is working and of there are any areas where improvement is necessary and possible.



*Figure 1.5 System development life cycle*

## **Requirement Analysis**

### **Identification of Need**

The present system, which is being implemented, is very complex. It is based on Command Line Interface that is very compound and hence it becomes very difficult for a normal user to interact with that software. Only those persons who have complete knowledge about the Java API and DOS can work on this. It becomes very difficult to deal with the same in absence of an expert. It can work with Oracle or mysql. The present system is not User-friendly. The manual system is very expensive and time consuming. The company was facing problems in keeping records of incoming and outgoing messages.

## **Conducting interviews**

We interviewed in some of the branches of the company with center head and the other employees to find out the facts behind the problem. As we interviewed in branches with different persons and we found some facts related to their problems. We put almost same questions. So it is difficult to give details of their interview. The question, which we put, was like as:

**The question, which we put with, the head of companies are as follows**

- What database are they using?

- How they manage the records?
- How you they manage in case of any crashes?
- How you they manage different cases of students?
- How much feedback do they get from their students and there records?
- Are you satisfied with the existing system?
- Is the existing system economical to you company?
- What is monthly expenditure of your company?

**The question, which we put with, the employees are as follows:**

- How you manage in the absence of an expert?
- What difficulties you face in managing the system if it hangs?
- Are you satisfied with the existing system?
- What they do when the number of incoming and outgoing messages is high?

## **Feasibility Study**

### **Technical Feasibility:**

Fundamentally, we are trying to answer the question “Can it actually be built?” To do this we investigated the technologies to be used on the project. For each technology alternative that we accessed we identified the advantages and disadvantages of it. By studying available resources and requirements, we concluded that minimum the application should be made user-friendlier.

The similar case exists with hardware. Most of the company is not ready to invest much more in purchasing new hardware peripherals. Therefore keeping their minimum available resources in mind we develop software, which do not require new software other than available software and the hardware requirement. The software, which we develop require following hardware and software.



Technology Issues	Market Issues
<ul style="list-style-type: none"> <li>• Performance</li> <li>• Ease of learning</li> <li>• Ease of deployment</li> <li>• Ease of support</li> <li>• Operational characteristics</li> <li>• Interoperability with other technologies</li> </ul>	<ul style="list-style-type: none"> <li>• Vendor viability</li> <li>• Alternate sources of the technology, if any</li> </ul>

*Table 1.1 some issues to consider when determining technical feasibility*

## **Economical Feasibility**

When accessing economical feasibility of an implementation alternative the basic question is “does the project make financial sense?” We did this by performing a cost/benefit analysis, which as its name suggests compares, the full/real costs of the application to its full/real financial benefits. The alternatives are evaluated on the basis of net cash flow, the amount by which the benefits exceed the cost, because the primary objective of all investments is to improve overall organizational performance. It was found that the company is unnecessarily spending money and in spite spending much money it is not getting satisfaction and truthful results. We also found that the company just wants to upgrade the existing system because they do not want to invest extra money to accept new system.

Keeping all these needs and demands of system in minimum budget we developed new software which will not only lower their budget but also not require much cost to accept it. It also does not need much maintenance. The new system will save the money, which they are investing on getting short codes, thus keeping all these good qualities is an economical feasibility.

Type	Potential Costs	Potential Benefits
Quantitative	<ul style="list-style-type: none"> <li>• Hardware/Software upgrades.</li> <li>• Training costs to train developers.</li> <li>• Expected operational costs.</li> </ul>	<ul style="list-style-type: none"> <li>• Reduced operating costs.</li> <li>• Increased revenue from additional sales of the organization services.</li> </ul>
Qualitative	<ul style="list-style-type: none"> <li>• Increased employee dissatisfaction from fear of change.</li> </ul>	<ul style="list-style-type: none"> <li>• Better response from customers</li> <li>• Positive public reception</li> </ul>

*Table 1.2 Potential costs and benefits of a software project*

## Operational Feasibility

Not only must an application make economical and technical sense, it must also make operational sense. The basic question that you are trying to answer is “Is it possible to maintain and support this application once it is in production?” Building an application is decidedly different than operating it, therefore one need to determine whether or not one can effectively operate and support it.

By carrying various types of interviews and questioning with center head of almost every center we found out their basic technical skills, their requirements, or whether they are ready to adopt new skills. We concluded that most of the companies wanted to adopt system which require not much more complex skills and are not ready to adopt an absolutely new environment, they wanted a system which is quite familiar with them and do not require extra skill to learn and operate it.

Keeping their demand and expectations in mind we developed a system which do not require any extra technical skill and training. It is developed using such environment, which are quite familiar to most of the people concerned with the system. The new system will prove easy to operate because it is developed in such a way so that it will prove user friendly. User will find it quite familiar and easy to operate because it is carrying necessary help guide and comment when required so that user does not have to face any difficulties, therefore it will prove feasible in respect to operate.

Operations Issues	Support Issues
<ul style="list-style-type: none"> <li>• What tools are needed to support operations?</li> <li>• What skills will operators need to be \trained in?</li> </ul>	<ul style="list-style-type: none"> <li>• What documentation will users be given?</li> <li>• How will change be managed?</li> </ul>

*Table 1.3 Issues to consider when determining the operational feasibility of a project.*

## 1.4 SOFTWARE TOOL USED FOR JAVA DEVELOPMENT:

### ECLIPSE IDE for JAVA DEVELOPERS

Eclipse IDE is a popular integrated development environment (IDE) used by Java developers. It's known for its powerful features and extensibility.

### Getting Started with Eclipse for Java Development

#### 1. Download and Install Eclipse:

- Go to the Eclipse Downloads page and download the “Eclipse IDE for Java Developers” package.

#### 2. Set Up a Java Project:

- Open Eclipse and create a new Java project by navigating to `File > New > Java Project`.
- Enter the project name and configure settings as needed.

#### 3. Write and Run Java Code:

- Right-click on `src` in the Project Explorer, then `New > Class` to create a new Java class.
- Write your code and save the file.
- To run your code, right-click on the file with the `main` method and select `Run As > Java Application`.

#### 4. Use Git Integration:

- To connect to a Git repository, go to `File > Import > Git > Projects from Git`, and follow the wizard to clone a repository.

#### 5. Debugging:

- Set breakpoints by double-clicking on the margin next to the line numbers.
- Start debugging by right-clicking on your Java class with the `main` method and selecting `Debug As > Java Application`.

### Tips for Effective Use

- **Customize Your Workspace:** You can arrange views and perspectives according to your preference.
- **Use Keyboard Shortcuts:** Familiarize yourself with Eclipse's keyboard shortcuts to boost productivity.
- **Explore Plugins:** Check out the Eclipse Marketplace for plugins that can add functionality tailored to your needs.

Eclipse is a versatile and feature-rich IDE that can support a wide range of Java development needs, from simple projects to complex enterprise applications.

## 1.5 JAVA AWT(Abstract Window Toolkit)

Java AWT or Abstract Window Toolkit is an API used for developing GUI(Graphic User Interfaces) or Window-Based Applications in Java. Java AWT is part of the Java Foundation Classes (JFC) that provides a way to build platform-independent graphical applications.

In this AWT tutorial, you will learn the basics of the AWT, including how to create windows, buttons, labels, and text fields. We will also learn how to add event listeners to components so that they can respond to user input.

By the end of this tutorial, you will have a good understanding of the AWT and be able to create simple GUIs in Java.

## Java AWT Basics

Java AWT (Abstract Window Toolkit) is an API used to create **Graphical User Interface (GUI)** or Windows-based Java programs and Java AWT components are platform-dependent, which means they are shown in accordance with the operating system's view. AWT is heavyweight, which means that its components consume resources from the underlying operating system (OS). The java.awt package contains AWT API classes such as *TextField*, *Label*, *TextArea*, *RadioButton*, *CheckBox*, *Choice*, *List*, and so on.

### Points about Java AWT components

- i.Components of AWT are heavy and platform dependent
- ii. AWT has less control as the result can differ because of components are platform dependent.

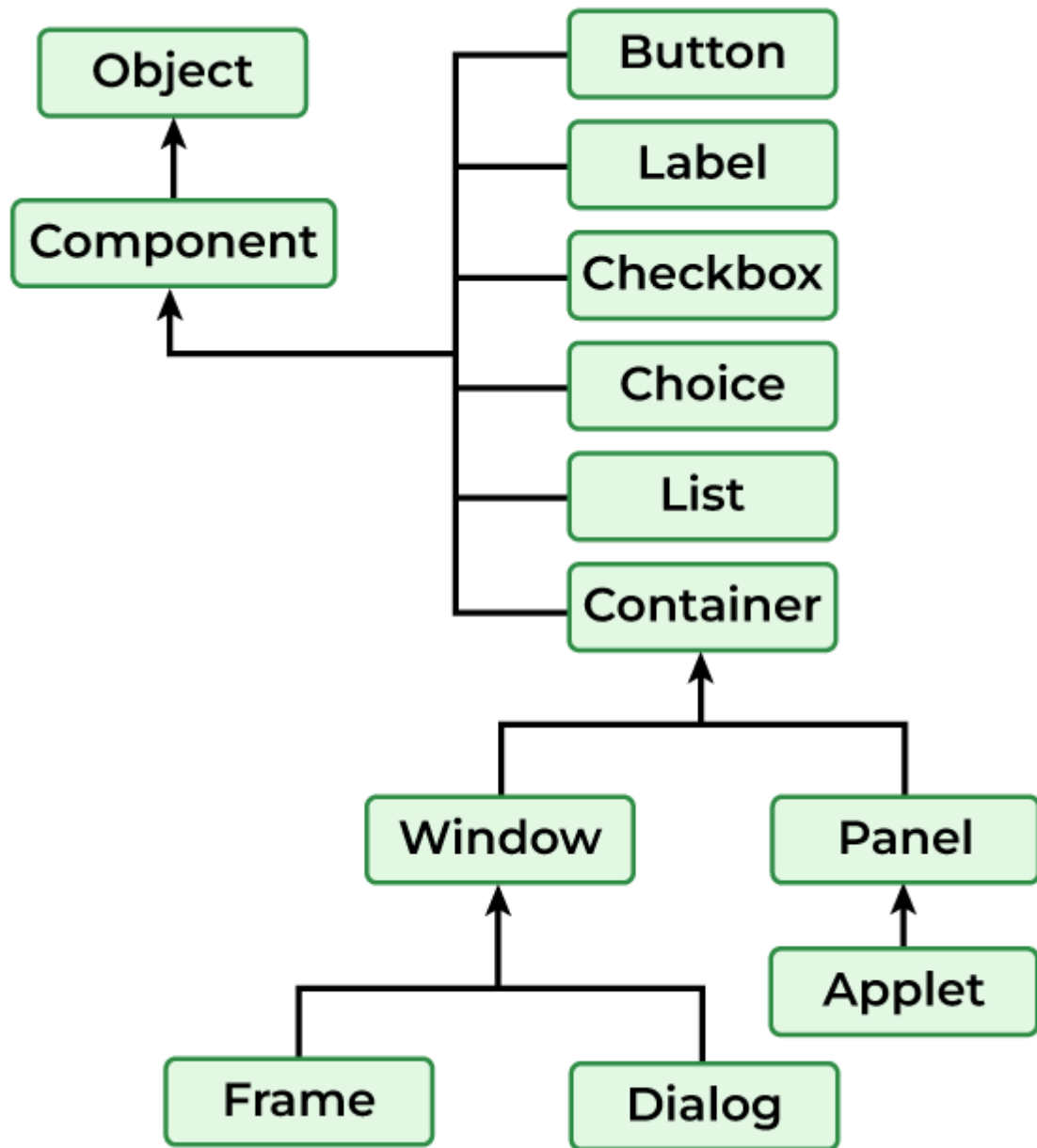
### Why AWT is Platform Independent?

The Java AWT utilizes the native platform subroutine to create API components such as *TextField*, *CheckBox*, and buttons. This results in a different visual format for these components on different platforms such as Windows, MAC OS, and Unix. The reason for this is that each platform has a distinct view of its native components. AWT directly calls this native subroutine to create the components, resulting in an AWT application resembling a Windows application on Windows OS, and a Mac application on the MAC OS. In simpler terms, the AWT application's appearance adapts to the platform it is running on.

AWT is platform independent even after the AWT components are platform dependent because of the points mentioned below:

- 1. JVM (Java Virtual Machine):** As Java Virtual Machine is platform dependent
- 2. Abstract APIs:** AWT provides an abstract layer for GUI. Java applications interact with AWT through Abstract API which are platform independent. Abstract API allows Java to isolate platform-specific details, making code portable across different systems.
- 3. Platform-Independent Libraries:** The Libraries of AWT are written in Java which they are totally platform-independent. Because of this, it ensures that AWT functionality remains consistent across different environments.

## Java AWT Hierarchy



*Figure 1.6 Hierarchy of Java AWT*

- **Components:** AWT provides various components such as buttons, labels, text fields, checkboxes, etc used for creating GUI elements for Java Applications.
- **Containers:** AWT provides containers like panels, frames, and dialogues to organize and group components in the Application.

- **Layout Managers:** Layout Managers are responsible for arranging data in the containers some of the layout managers are BorderLayout, FlowLayout, etc.
- **Event Handling:** AWT allows the user to handle the events like mouse clicks, key presses, etc. using event listeners and adapters.
- **Graphics and Drawing:** It is the feature of AWT that helps to draw shapes, insert images and write text in the components of a Java Application.

## 1.6 JAVA SWINGS

**Java Swing tutorial** is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

### Features Of Swing Class

- Pluggable look and feel.
- Uses MVC architecture.
- Lightweight Components
- Platform Independent
- Advanced features such as JTable, JTabbedPane, JScrollPane, etc.
- Java is a platform-independent language and runs on any client machine, the GUI look and feel, owned and delivered by a platform-specific O/S, simply does not affect an application's GUI constructed using Swing components.
- **Lightweight Components:** Starting with the JDK 1.1, its AWT-supported lightweight component development. For a component to qualify as lightweight, it must not depend on

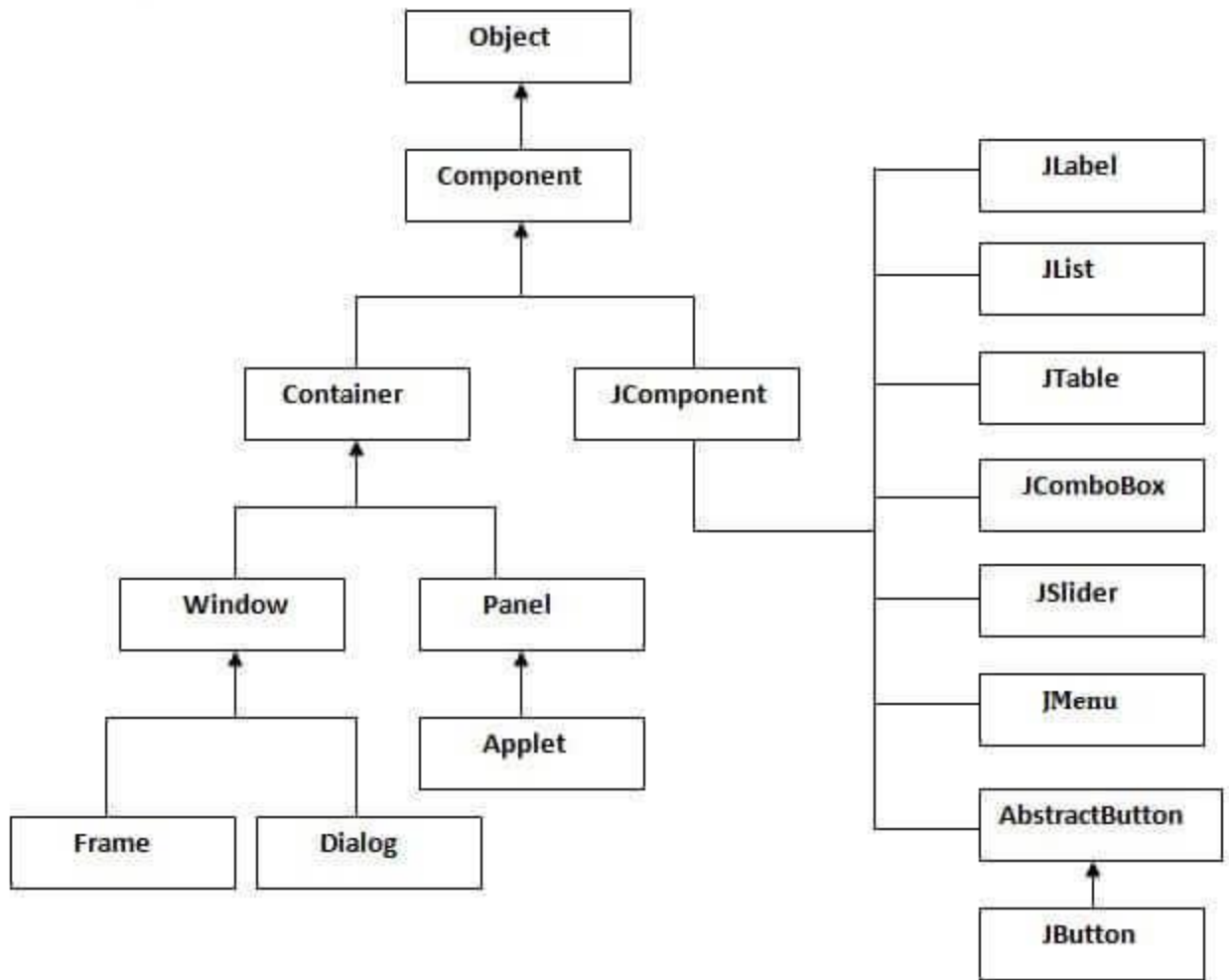
any non-Java (O/s based) system classes. Swing components have their own view supported by Java's look and feel classes.

- **Pluggable Look and Feel:** This feature enable the user to switch the look and feel of Swing components without restarting an application. The Swing library supports components' look and feels that remain the same across all platforms wherever the program runs. The Swing library provides an API that gives real flexibility in determining the look and feel of the GUI of an application
- **Highly customizable** – Swing controls can be customized in a very easy way as visual appearance is independent of internal representation.
- **Rich controls**– Swing provides a rich set of advanced controls like Tree TabbedPane, slider, colorpicker, and table controls.



## Hierarchy of Java Swing classes

The hierarchy of java swing API is given below.



*Figure 1.7 Hierarchy of Java Swing*

Class	Description
Component	A Component is the Abstract base class for about the non-menu user-interface controls of Java SWING. Components are representing an object with a graphical representation.
Container	A Container is a component that can container Java SWING Components
JComponent	A JComponent is a base class for all swing UI Components In order to use a swing component that inherits from JComponent, the component must be in a containment hierarchy whose root is a top-level Java Swing container.
JLabel	A JLabel is an object component for placing text in a container.
JButton	This class creates a labeled button.
JColorChooser	A JColorChooser provides a pane of controls designed to allow the user to manipulate and select a color.
JCheckBox	A JCheckBox is a graphical (GUI) component that can be in either an on-(true) or off-(false) state.
JRadioButton	The JRadioButton class is a graphical (GUI) component that can be in either an on-(true) or off-(false) state. in the group
JList	A JList component represents the user with the scrolling list of text items.
JComboBox	A JComboBox component is Presents the User with a show up Menu of choices.
JTextField	A JTextField object is a text component that will allow for the editing of a single line of text.

Class	Description
JPasswordField	A JPasswordField object it is a text component specialized for password entry.
JTextArea	A JTextArea object is a text component that allows for the editing of multiple lines of text.
ImageIcon	A ImageIcon control is an implementation of the Icon interface that paints Icons from Images
JScrollbar	A JScrollbar control represents a scroll bar component in order to enable users to Select from range values.
JOptionPane	JOptionPane provides set of standard dialog boxes that prompt users for a value or Something.
JFileChooser	A JFileChooser it Controls represents a dialog window from which the user can select a file.
JProgressBar	As the task progresses towards completion, the progress bar displays the tasks percentage on its completion.
JSlider	A JSlider this class is letting the user graphically (GUI) select by using a value by sliding a knob within a bounded interval.
JSpinner	A JSpinner this class is a single line input where the field that lets the user select by using a number or an object value from an ordered sequence.

*Table 1.4 Components of Swing Class the task's percentage*

## **CHAPTER 2 TRAINING WORK UNDERTAKEN**

### **2.1 SYSTEM DESIGN**

#### **INTRODUCTION -:**

The systems objectives outlined during the feasibility study serve as the basic from which the work of system design is initiated. Much of the activities involved at this stage is of technical nature requiring a certain degree of experience in designing systems, sound knowledge of computer related technology and through understanding of computers available in the market and the various facilities provided by the vendors. Nevertheless, a system cannot be designed in isolation without the active involvement of the user. The user has a vital role to play at this stage too. As we know that data collected during feasibility study will be utilized systematically during the system design. It should, however be kept in mind that detailed study of the existing system is not necessarily over with the completion of the feasibility study. Depending on the plan of feasibility study, the level of detailed study will vary and the system design stage will also vary in the amount of investigation that still needs to be done. This investigation is generally an urgent activity during the system. Sometimes, but rarely, this investigation may form a separate stage between feasibility study and computer system design. Designing a new system is a creative process, which calls for logical as well as lateral thinking. The logical approach involves systematic moves towards the end product keeping in mind the capabilities of the personnel and the equipment at each decision making step. Lateral thought implies encompassing of ideas beyond the usual functions and equipment. This is to ensure that no efforts are being made to fit previous solutions into new situations.

#### **System Design Considerations**

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, But still much of design work depends on knowledge and experience of the designer. When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user or limitations of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions there of since

the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system designing phase:

### **Design Objectives**

The primary objective of the design of course, is to deliver the requirements as specified in the feasibility reports. In general the following design objectives should be kept in mind.

#### **Practicality**

The system must be stable and can be operated by people with average.

#### **Efficiency**

This involves accuracy, timeliness and comprehensiveness to the system output.

#### **Cost**

It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.

#### **Flexibility**

The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.

#### **Security**

This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse. System design involves first logical design and then physical construction of the system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, database and procedures. The physical construction, which follows the logical design, produces actual program software, files and a working system.

### **Major System Design Activities**

Several development activities are carried out during structured design. They are database design, implementation planning, system test preparation, system interface specification, and user documentation.

## **Database design**

This activity deals with the design of the physical database. A key is to determine how the access paths are to be implemented.

Program design: In conjunction with database design is a decision on the programming language to be used and the flowcharting, coding, and debugging procedure prior to conversion. The operating system limits the programming languages that will run on the system. System and program test preparation. Each aspect of the system has a separate test requirement. System testing is done after all programming and testing completed the test on system and program test requirements become a part of design specifications a prerequisite to implementation.

In contrast to the system testing is acceptance testing, which puts the system through a procedure design to convince the user that the proposed system will meet the stated requirements. Acceptance testing is technically similar to system testing but politically it is different.

## **Design Process**

The computer system design process is an exercise of specifying how, the system will work. It is an iterative process, which is based on what the system will be do as shown in the feasibility report. Mainly, following five parts have been included in the system design process

### **Output Design**

The starting point of the design process is the proper knowledge of system requirements which will normally be converted in terms of output.

### **Input Design**

Once the output requirements have been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs. The basic documents in which these data are available need to be identified. If necessary, these documents may have to be revised or new documents may have to be introduced.

## **File Design**

Once the input data is captured in the system, these may to be preserved either for a short or long period. These data will generally be stored in files in a logical manner. The designer will have to devise the techniques of storing and retrieving data from these files.

## **Procedure Design**

This step involves specifications of how processing will be performed. In this, there are two aspects:

## **2.2 INPUT DESIGN(Introduction)**

Once the analysis and design of the system has been done, it would be necessary to identify the data that are required to be processed to produce the outputs. Input is one of the most expensive phases of the operation of a computerized system and creates sometimes a major problem. Different type of problem with a system can usually be traced back to faulty input design method needless to say, therefore, that the input data are the lifeblood of a system and have to be analyzed and designed with utmost care and consideration. Input design features can ensure the reliability of the system and generate correct reports form the accurate data. The input design also determines whether the user can interact efficiently with the system.

## **Elements of Input Data**

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing.

## **Input Data**

The goal of designing input data is to make data entry as easy, logical and error free from errors as possible. In entering data, operators need to know the following:

- The allocated space for each field.
- Field sequence, which much match that in the source document.
- The format in which data fields are entered for example, filling out the date field is required through the edited format mm/dd/yy.

- When we approach input data design, we design the source document. Let us elaborate on each step.

### **Source Documents**

Source data are captured initially on original paper or a source document. For example, a cheque written against an account is a source document. When it reaches the bank, it is encoded with special magnetic ink character recognition so that a reader that is part of the information system of the bank can process it. Therefore, source documents initiate a processing cycle as soon as they are entered into the system. Source documents may be entered into the system from punch cards, from diskettes, or even directly through the keyboard.

A source document should be logical and easy to understand. Each area in the form should be clearly identified and should specify for the user what to write and where to write it. A source document may or may not be retained in the proposed system. Thus, each source document may be evaluated in terms of its continued use in the proposed system, the extent of modification for the proposed system & Replacement by an alternative source document.

### **Input Design Guidelines**

The design of input play very significant role in getting the correct output. It covers all phases of input from creation of initial data (original recording) to actual entering the data to the system for processing. The input design is the link that ties the information system into the world of its users. Some features of design may vary depending on whether the system is batch-oriented or on-line. Here, we will discuss the various objectives of input design. They focus on:

- Controlling amount of input
- Avoiding delay
- Avoiding errors in data
- Avoiding extra steps
- Keeping the process simple



## **2.3 OUTPUT DESIGN (Introduction)**

Presenting the data processed by a computer-based information system in an attractive and usable form has become very essential these days' success and acceptance of a system to some extent depends on good presentation. Therefore, system analyst must know fully how to design output report in an attractive way. Many new output devices are being introduced in the market because of recent development in computer technology. System analyst must be aware of these new technologies and try to use these new output devices if possible. Currently, excellent graphic displays are widely available. Speech output systems are also fast emerging.

There are three main reasons why outputs from the computer are required. They are:

- For communication to the persons concerned.
- For re-input to the computer for being connected with other data and further processing.
- For permanent storage.

### **Types of Output:**

Outputs of a system can take different forms. The most common are reports, displays on screen, printed forms etc. the outputs also vary in terms of their contents, type of stationery. Frequency and timing etc. besides, due consideration also need to be given as to who will use the output and for what purpose. All these points must be kept in mind while designing outputs so that the objectives of the system are met in the best possible way.

Outputs of a data-processing system can be placed into two categories:

- Application Output
- Operating Output

### **Application Output**

These are the outputs desired out of the system to meet its objectives. These are of three types:

- Output as a basis for decision-making. This type of output is generally required by management for decision-making purposes.

- Output as a requirement to meet a functional objective. Invoices, Excise Gate Pass, Purchase Orders are the examples of such output.
- Statutory outputs: All organization is required to produce a certain amount of reports and forms as required by law.

### **Operating Output**

These outputs are mainly generated for use of EDP staff and give various indications as to how the system operates. System logs, error messages, status indicators etc. are the examples of such output. These types of output are not concerned for the users.

## **USED TECHNOLOGY**

### **2.4 Java:**

- In 1990, Sun Microsystems Inc. (US) was conceived a project to develop software for customer electronics devices that could be controlled by a remote. This project was called Stealth Project but later its name was changed to Green Project. In 1991, Bill Joy, James Gosling, and several others met in Aspen, Colorado to discuss this project. James Gosling was to identify the proper programming language for the project. Gosling thought C and C++ could be used to develop the project. But the problem they faced with them is that they were system dependent languages and hence could not be used on various processors, which the electronic devices might use. So started developing a new language, which was completely system independent. This language was initially called Oak. Since this name was registered by some other company, later it was changed to Java.
- Why the name Java? James Gosling and his team members were consuming a lot of coffee while developing this language. They felt that they were able to develop a better language because of the good quality coffee they consumed. So the coffee had its own role in developing this language and good quality coffee was exported to the entire world from a place called 'Java Island'. Hence they fixed the name of the place for the language as Java. And the symbol for Java language is coffee cup and saucer.

- On Jan 23<sup>rd</sup> 1996, JDK 1.0 version was released. Today more than 4 million developers use Java and more than 1.75 billion devices run Java. Thus Java was pervaded the world. Java is software that enables users to run program on any type of computer system. Java is suitable for internet because of its independency. It eliminates a lot of security problems for data on internet. It is purely object-oriented language. This means Java program use classes and objects. Because of its lot of features we create our project report of Hospital Management System in this language.

## **Technological Specifications**

### **Polymorphism:**

- Polymorphism came fr the two Greek words ‘poly’ meaning many and morphos meaning forms. The ability to exist in different forms is called ‘polymorphism’. In Java, a variable, an object, or a method can exist in different forms, thus performing various tasks depending on the context. Because same variable or method can perform different tasks, the programmer has the advantage of writing flexible code.

### **Abstract class:**

- An abstract class is a class that contains 0 or more abstract methods. It can contain instances variables and concrete methods in addition to abstract methods. It should be declared by the keyword ‘abstract’.

### **Interfaces:**

- An interface contains only abstract methods which are all incomplete methods. So it is not possible to create an object to an interface. In this case, we can create separate classes where we can implement all the methods of the interface. These classes are called implementation classes.

### **Packages:**

- It is necessary in software development to create several classes and interfaces. After creating these classes and interfaces, it is better if they are divided into some groups depending on their relationship. Thus, the classes and interfaces which handle similar or same task are put into the same directory. This directory or file is also called a package.

### **Exception Handling:**

- When there is an exception, the user data may be corrupted. This should be tackled by the programmer by carefully designing the program. For example, if your application unsuccessfully tries to log into a database, it should not display an error message that includes the user name it is using.

## **2.5 MYSQL DATABASE**

### **MYSQL**

MYSQL is comprehensive, integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. MYSQL allows companies to gain greater insight from their business information and achieve faster results for a competitive advantage.

The driving force behind MYSQL is the process of integration. With MSSQL 2005 the database developer experiences a paradigm shift. He can now locate his code with reference to its functionality; he can access data in its native formats or build complex systems that are server driven.

### **ABOUT MYSQL**

Among the many application development and end user products available within the MYSQL family there is a common ability to access the database. Whether directly or indirectly this is achieved through the Structured Query Language (SQL). These query languages have been useful in developing the software in an efficient way. The Query language has Data Definition Language (DDL), Data Manipulation Language (DML) and Data Control Language (DCL).

**Data definition language** helps in defining various database objects such as Tables and views. This has been useful in defining all the relational tables of the project. It has also been used in creating many views providing for information hiding and data security. Data Definition Language (DDL) also supports alteration of the existing tables.

**Data Manipulation Language** (DML) is the most important of the query Language. This has features for inserting records into the tables, Deleting the existing records from the tables, Updating the existing records in the tables and retrieving the data from one or more tables as and when required.

**Data Control Language** (DCL) is used for controlling data and having access to the databases. This part of the language provides extensive security features in order to safe guard the users information from both unauthorized access and intentional damage. Granting and revoking privileges provide security. Oracle users have names and passwords and own tables, views and other database objects.

These facilities have eased the job of developing the software to meet the organizational requirements. Many similar programming logic have been coded as functions and procedures and have been used in many areas. For Example retrieving the records, Calculation of various parameters of Pay etc. This has been a big advantage in developing the package.

The facility of exceptions also helped in developing a user-friendly package. Exceptions are raised and a suitable actions are taken when errors occur such as ‘when no data found’, ‘Too many rows’ etc.

The facility of Cursors where used in retrieving records one by one in an orderly way. This helps in management in viewing the records one after other to get the required information about each employee. This usage of Cursors eliminates many errors such as ‘Too many rows’ etc. Each record can be retrieved and studied for any amount of time before retrieving the next record. Cursors retrieve the records in ascending order of the values of the primary Key.

## **CHAPTER 3 RESULTS AND DISCUSSION**

### **3.1 TESTING**

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. The tester may analysts, programmer or a specialist trained for software testing, is actually trying to make the program fail. Analysts know that an effective testing program does not guarantee system reliability. Therefore reliability must be designed into the system.

#### **Unit Testing**

In unit testing we have to test the programs making up the system. For this reason unit testing is sometimes called as the Program testing. The software units in a system are modules and routines that are assembled and integrated to perform a specific function.

Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within the module alone. Unit testing can be performed from the bottom up, starting with the lowest level modules and proceeding one at a time. Unit testing is done for each module in Online\_Examination. This ensures that the value we enter match with the data type and within the specified limits.

#### **Integration Testing**

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing for conducting tests to uncover errors associated within the

interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus in the integration testing step, all the errors are corrected for the next testing steps. In Online\_Examination each module is integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

## **Validation Testing**

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been recovered and corrected and a final series of a software tests-validation tests begin.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

In validation testing if user wants to enter the numeric value he can only enter the numeric value not the text value. For e.g.: in phone number field user can only enter numeric value to it. The system is user friendly with user guide and messages to explain further procedures. An attempt has been made to perfect the process by incorporating validation at each level.

## **3.2 RESULTS**

The Railway Reservation System developed in Java has achieved the primary objectives set for the project. The system was able to effectively handle user registrations, train searches, ticket bookings, and cancellations. Furthermore, it facilitated administrative functions, allowing for the management of train schedules and generating reports. The following key results were observed:

### **1. Functional Capabilities:**

- The system allowed users to register, log in, and search for trains based on source, destination, and travel dates. It successfully displayed seat availability and enabled users to book or cancel tickets.
- A unique PNR was generated for each successful booking, which users could later use to view or cancel their reservations.
- The admin panel provided an interface for managing train schedules, updating train details, and monitoring system performance.

### **2. Database Integration:**

- The system efficiently stored and retrieved data from the database using Java JDBC. Real-time updates to seat availability, bookings, and cancellations were successfully implemented, ensuring the consistency of the system's data.
- The database design ensured data integrity, with normalized tables and relationships that prevented redundancy and errors in bookings or cancellations.

### **3. System Performance:**

- The system handled multiple operations (booking, cancellation, and train management) without significant performance delays. For a moderate number of users, it provided a responsive experience, both in terms of train search and booking confirmation.

### **4. Security and Data Integrity:**

- The use of prepared statements effectively prevented SQL injection attacks, ensuring secure interactions with the database. Basic user authentication protected personal data, though more advanced security measures could be added in future versions.

### **5. Limitations:**

- The console-based interface was functional but not as user-friendly as a GUI. The project was successful in handling the core processes but would benefit from further enhancements such as a graphical interface, real-time updates, and payment integration.

Overall, the Railway Reservation System demonstrated the successful application of object-oriented principles and database management in Java, meeting the primary functional and technical requirements for a reservation system.

## **3.3 DISCUSSION**

The development of the Railway Reservation System in Java has been a significant learning experience, offering insights into both technical and functional aspects of software development. This project aimed to streamline the booking, cancellation, and management of train reservations, addressing key issues such as user experience, efficiency, and data integrity. Below are key points discussed based on the overall development process:



## 1. Technology Stack and Design Considerations

Java was chosen for this project due to its platform independence, strong community support, and robust libraries for building secure, efficient, and maintainable applications. Using object-oriented programming principles in Java, such as encapsulation, inheritance, and polymorphism, ensured modularity and reuse of code. Moreover, Java's JDBC API was used for database connectivity, ensuring smooth interactions with the relational database, such as MySQL or SQLite, chosen to store user and train information.

In terms of design, the system was modeled using a layered architecture: presentation, business logic, and data access. This separation of concerns facilitated easier debugging, maintenance, and scalability, especially for future enhancements like adding payment integration or real-time seat availability.

## 2. Key Features Implemented

The system implements several key features typically expected from a railway reservation system:

- **User Registration and Authentication:** Secure user login with options for new users to register, ensuring personalized services such as booking history.
- **Train Search and Availability:** Users can search for trains based on source, destination, and travel dates. The system displays available seats, taking into account real-time updates from the database.
- **Ticket Booking and Cancellation:** Booking a ticket automatically reduces seat availability and generates a unique PNR for tracking purposes. The cancellation module ensures smooth refunds and updates to seat availability.
- **Admin Module:** An admin section allows for the management of train schedules, adding or updating train details, and generating reports.

## 3. Challenges Faced

The primary challenge during development was ensuring data consistency, especially when dealing with concurrent reservations. Handling transactions and preventing race conditions required

careful design of the database queries and locking mechanisms. Using transactions in the database ensured that no two users could book the same seat simultaneously.

Additionally, managing the UI/UX within a console-based Java application posed challenges. While the system is functional, user experience could be further improved through the use of a graphical interface such as JavaFX or by integrating the system with web technologies.

#### **4. Database Design and Integrity**

Database management was critical for the system's success. Normalization techniques were applied to avoid redundancy and ensure data integrity. Tables were designed for storing user information, train schedules, reservations, and cancellations, with proper indexing to optimize query performance. Ensuring referential integrity between tables (e.g., linking reservations to user IDs and train schedules) was crucial in maintaining a reliable database.

The use of prepared statements in Java helped prevent SQL injection, enhancing security during interactions with the database. Further, the decision to use foreign key constraints ensured that orphaned records (such as reservations without corresponding user records) could not exist.

#### **5. Performance Considerations**

Performance was a key focus, particularly when retrieving train schedules and availability data. Efficient query design and indexing improved performance under high traffic. However, as the system scales, additional measures such as caching, database sharding, or even shifting to a distributed database may be necessary to handle larger datasets and increased concurrency.

On the client-side, loading times and response feedback were kept to a minimum, although performance may degrade as the number of users increases. Future work could include stress testing to measure the system's performance under load and implement improvements accordingly.

#### **6. Security**

Security was another important aspect of this project. While the current implementation handles user authentication and avoids common vulnerabilities like SQL injection, future iterations should consider more robust authentication mechanisms, such as two-factor authentication or OAuth

integration. Encryption of sensitive data, such as passwords and personal information, is crucial to ensure data security.

## **7. Areas for Improvement and Future Enhancements**

While the Railway Reservation System fulfills the basic functional requirements, several areas for improvement were identified:

- **Graphical User Interface (GUI):** Moving from a console-based to a GUI-based system using JavaFX or Swing would provide a more user-friendly interface.
- **Real-time Data Synchronization:** Implementing real-time updates for train schedules and seat availability (e.g., using WebSocket or polling) would improve the user experience.
- **Payment Integration:** Integrating secure payment gateways would enhance the system, allowing users to complete the booking process entirely online.
- **Mobile App Integration:** With the increasing use of mobile devices, developing a corresponding mobile application or responsive web interface could make the system more accessible.

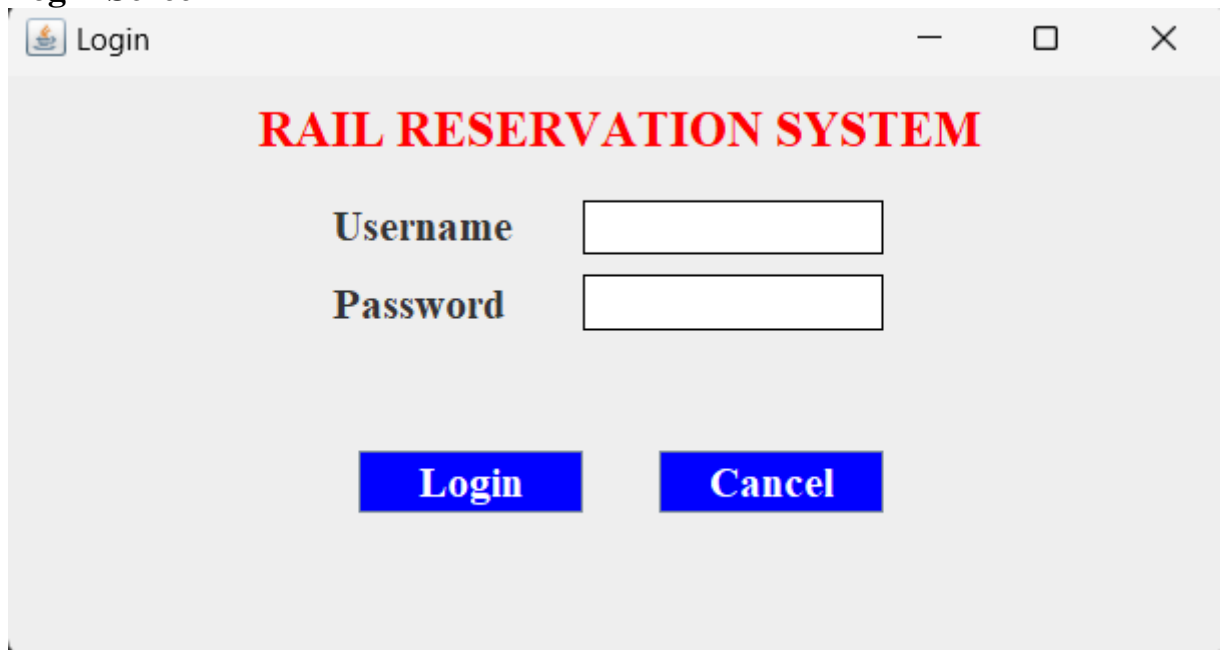
# Graphical User Interface

## Welcome Screen



*Figure 3.1 Welcome Screen*

## Login Screen



The image shows a login window titled "Login" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area has a light gray background. At the top, the text "RAIL RESERVATION SYSTEM" is displayed in a large, bold, red serif font. Below this, the labels "Username" and "Password" are shown in a bold black serif font, each followed by a white rectangular input field. At the bottom, there are two blue rectangular buttons with white text: "Login" and "Cancel".

**RAIL RESERVATION SYSTEM**

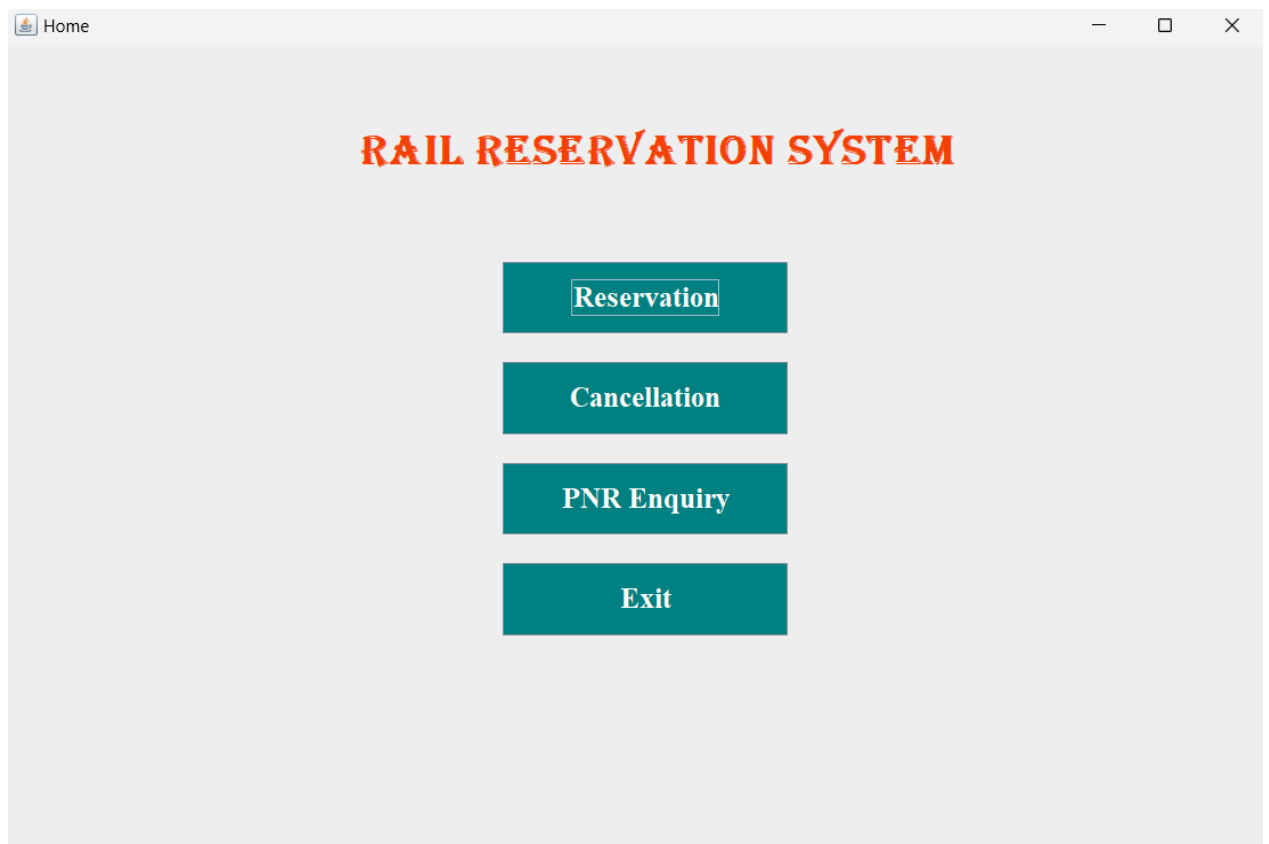
**Username**

**Password**

**Login** **Cancel**

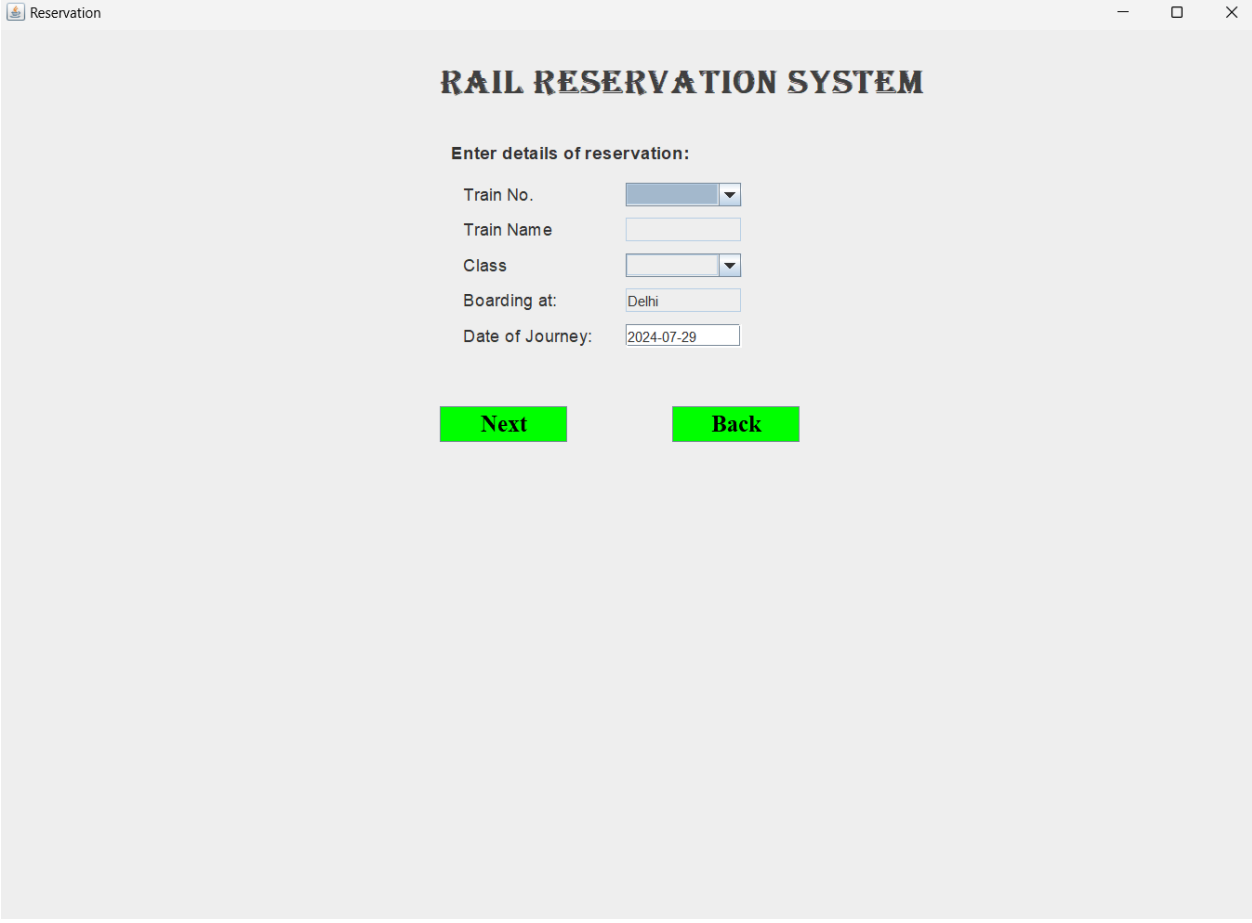
*Figure 3.2 Login Screen*

## Main Screen



*Figure 3.3 Main Screen*

## Reservation Form



The screenshot shows a web application window titled "Reservation" with standard window controls (minimize, maximize, close). The main content area has a light gray background and features the title "RAIL RESERVATION SYSTEM" in bold, black, uppercase letters. Below the title, the instruction "Enter details of reservation:" is displayed. The form contains six input fields arranged vertically: "Train No." (a dropdown menu), "Train Name" (a text box), "Class" (a dropdown menu), "Boarding at:" (a text box containing "Delhi"), and "Date of Journey:" (a text box containing "2024-07-29"). At the bottom of the form, there are two green buttons: "Next" on the left and "Back" on the right.

**RAIL RESERVATION SYSTEM**

Enter details of reservation:

Train No.

Train Name

Class

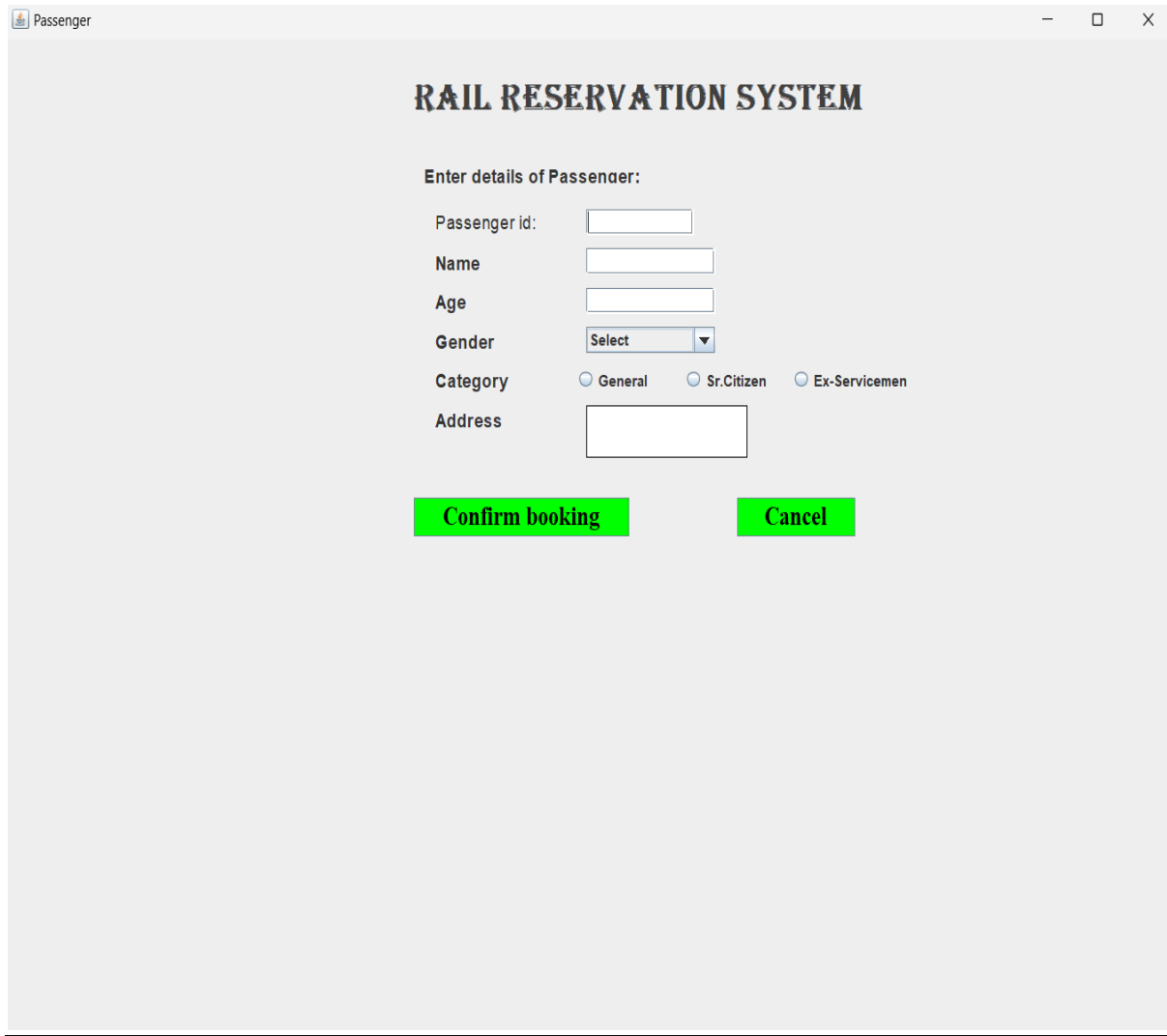
Boarding at:

Date of Journey:

**Next** **Back**

*Figure 3.4 Reservation Form*

## Passenger Form



The screenshot shows a window titled "Passenger" with standard window controls (minimize, maximize, close). The main content area has a title "RAIL RESERVATION SYSTEM" in bold. Below the title, the instruction "Enter details of Passenger:" is followed by a series of input fields: "Passenger id:" with a text box, "Name" with a text box, "Age" with a text box, "Gender" with a dropdown menu showing "Select", "Category" with three radio buttons labeled "General", "Sr.Citizen", and "Ex-Servicemen", and "Address" with a text box. At the bottom of the form are two green buttons: "Confirm booking" and "Cancel".

**RAIL RESERVATION SYSTEM**

Enter details of Passenger:

Passenger id:

Name

Age

Gender

Category ☐ General ☐ Sr.Citizen ☐ Ex-Servicemen

Address

**Confirm booking** **Cancel**

*Figure 3.5 Passenger Form*



## Enquiry Form

PNR Enquiry

# RAIL RESERVATION SYSTEM

Enter your PNR No.

[Search](#) [Home](#)

**PNR Details**

**Passenger Details**

Figure 3.6 Enquiry Form

# Cancellation Form

Cancellation

RAIL RESERVATION SYSTEM

PNR No.

Train No.

Train Name

Class

Date of Journey

From

To

Boarding At

Passenger id

Search

Delete

Home

Figure 3.7 Cancellation Form

## Database Layout

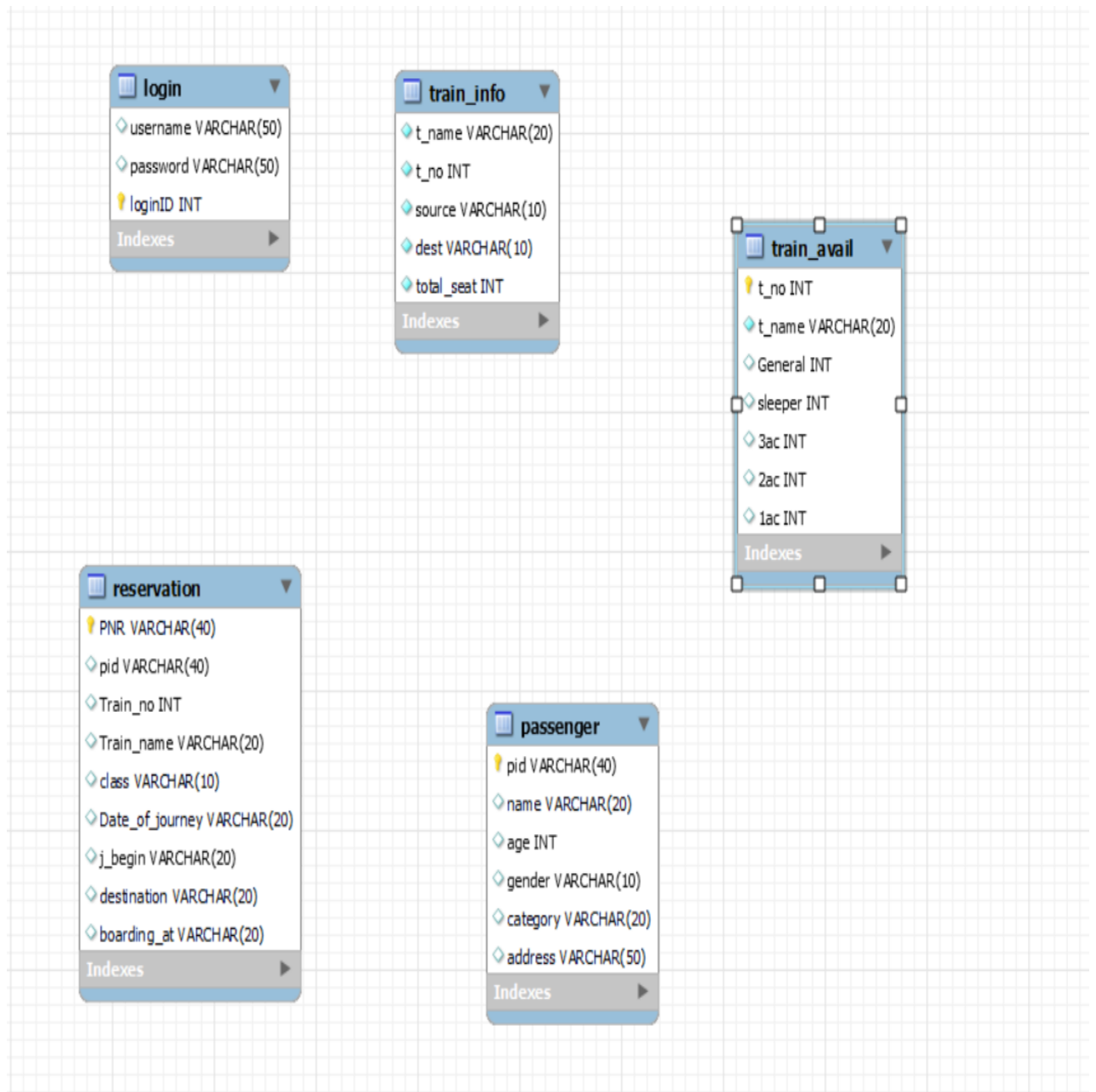


Figure 3.8 Database

## CHAPTER 4 CONCLUSION AND FUTURE SCOPE

### 4.1 SCOPE OF THE PROJECT

#### FUTURE SCOPE:-

The future scope of a **Java project on Railway Reservation System** can be quite broad and promising, given the technological advancements in software development, big data, AI, and IoT. In terms of scalability, the system could be designed to accommodate a large volume of users, handle high traffic, and ensure seamless performance even during peak hours.

Here are several areas where such a project can be enhanced or extended:

#### 1. Integration with Mobile Platforms

- **Mobile App Development:** Extend the system to include Android/iOS mobile applications, offering passengers a seamless way to book tickets, check schedules, and manage reservations on the go.
- **Push Notifications:** Implement real-time notifications for train schedules, delays, cancellations, and booking confirmations directly to the user's phone.

#### 2. AI-Powered Predictive Analytics

- **Demand Prediction:** Use AI/ML algorithms to predict future booking trends, which can help in optimizing seat allocation, dynamic pricing, and ensuring efficient train operations.
- **Customer Behavior Analytics:** Use machine learning to study user behavior, personalize travel suggestions, and provide targeted offers or discounts.

#### 3. Real-Time Train Tracking

- **GPS Integration:** Use GPS and IoT to integrate real-time train tracking, giving passengers real-time updates on train locations, estimated arrival times, and platform numbers.
- **Smart Station Information:** Provide real-time information about nearby stations, train timings, and other facilities to passengers based on their location.

## 4.2 CONCLUSION

The **development of the Railway Reservation System** in Java has successfully tackled the essential functionalities of train booking, ticket cancellation, and schedule management. By utilizing Java's object-oriented programming principles, the system was constructed with a focus on **modularity, scalability, and maintainability**, ensuring that it can adapt to future needs with minimal disruption.

### Core Architectural Design

The system was built on a **layered architecture**, seamlessly integrating the **presentation layer, business logic layer, and data access layer**. This approach not only provided a clear separation of concerns but also created a **solid foundation** for any future enhancements, allowing the system to evolve efficiently over time. Each layer is crafted to independently handle its responsibilities, improving the overall **robustness and flexibility** of the application.

### Key Functionalities

The Railway Reservation System incorporates a variety of **user-centric features**:

1. **Train Search**: Facilitating the search for trains based on user-specified criteria like destination, date, and train type.
2. **Ticket Booking**: Allowing users to seamlessly book tickets, with real-time feedback on seat availability and other necessary information.

### Database Connectivity & Security

For data handling and storage, **Java Database Connectivity (JDBC)** was employed, ensuring efficient and **real-time database interactions**. This ensures that user data and train availability are consistently updated and managed with precision. Additionally, **data consistency and security** were key considerations throughout the system's design:

- **Prepared statements** were utilized to prevent **SQL injection attacks**, safeguarding the system from malicious inputs.

# REFERENCES

## CITATION REFERENCES

### Books/Sites Referred

1. Herbert Schildt, Java: A Beginner's Guide, 7th ed. New York, USA: McGraw-Hill Education, 2018.
2. Orla Higgins, Flex on Java, 1st ed. Birmingham, UK: Packt Publishing, 2008.
3. [www.roseindia.net](http://www.roseindia.net)
4. [www.java2s.com](http://www.java2s.com)
5. [www.javatpoint.com](http://www.javatpoint.com)
6. [www.youtube.com](http://www.youtube.com)