HELPDESK MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

GRISH NARAYANAN S- (2303811710421048)

In partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution , affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621112.

NOVEMBER - 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **“HELPDESK MANAGEMENT SYSTEM”** is the bonafide work of **GRISH NARAYANAN S - (2303811710421048)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING
Dr.A.DELPHIN CAROLINA RANI, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

[illegible]

CGB1201-JAVA PROGRAMMING
Mr. M. SARAVANAN, M.E.,
SUPERVISOR
ASSISTANT PROFESSOR

[illegible]

SIGNATURE	**SIGNATURE**
Dr.A.Delphin Carolina Rani ,M.E.,Ph.D.,	Mr. M. Saravanan, M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANTPROFESSOR
Department of CSE	Department of CSE
K.Ramakrishnan College of Technology (Autonomous)	K.Ramakrishnan College of Technology (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the viva-voce examination held on 02.12.2024

CGB1201-JAVA PROGRAMMING
Mr.MAHARMANNAN A, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

[illegible]

CGB1201-JAVA PROGRAMMING
Dr.R.SEETHAMILSELVI, M.E., Ph.D.,
EXTERNAL EXAMINER
PROFESSOR
8138-SCE, TRICHY.

[illegible]

INTERNAL EXAMINER EXTERNALEXAMINER

DECLARATION

I declare that the project report on **“HELPDESK MANAGEMENT SYSTEM”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

Grish

Signature

GRISH NARAYANAN S

Place: Samayapuram

Date: 02.12.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology(Autonomous)**",for providing us with the opportunity to do this project.

I am glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave me the opportunity to frame the project with full satisfaction.

I wholeheartedly thank **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encouragement in pursuing this project.

Iexpressourdeepexpressionandsinceregratitudetoourprojectsupervisor **Mr.M.SARAVANAN,M.E.,**Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OFTHE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSIONOFTHEINSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talentand enhancing the competency of students to transform them as all-round personality respecting moral and ethical values.

VISIONOF DEPARTMENT

To be a center of eminence in creating competent software professional swith research and innovative skills.

MISSIONOF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2:Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAMEDUCATIONALOBJECTIVES

PEO1:Domain Knowledge

To produce graduates who have a strong foundation of knowledge and skills in the field of Computer Science and Engineering.

PEO2:Employability Skills and Research

To produce graduates who are employable in industries/publicsector/research organizations or work as an entrepreneur.

PEO3:Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAMSPECIFICOUTCOMES(PSOs)

PSO1:Domain Knowledge

Toanalyze,design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO2:Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO3: Innovation Ideas

To adapt to emerging Information and Communication Technologies(ICT)toinnovate ideas and solutions to existing/novel problems.

PROGRAM OUT COMES(POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems anddesignsystemcomponentsorprocessesthatmeetthespecifiedneedswithappropriate considerationforthepublichealthandsafety,andthecultural,societal,andenvironmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and researchmethodsincludingdesignofexperiments,analysisandinterpretationofdata,and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legaland culturalissues and theconsequentresponsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:**Applyethicalprinciplesandcommittoprofessionalethicsandresponsibilitiesand norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-longlearning:**Recognizetheneedfor,andhavethepreparationandabilitytoengage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The HelpDesk Management System is a Java application designed to efficiently manage support tickets using a graphical user interface built with Swing. The system comprises two main classes: `HelpDeskSystem` and `Ticket`. The `HelpDeskSystem` class extends `JFrame` and creates the main window with components such as `JTextArea` for displaying tickets, `JTextField` for user input, and buttons to create, view, and close tickets. User interactions are handled by implementing the `ActionListener` interface, which triggers actions like ticket creation, viewing, and closure. The `Ticket` class represents individual tickets with a description and status (open or closed). Tickets are stored dynamically in an `ArrayList`, enabling efficient management and updates. This system provides a user-friendly interface for handling support tickets, ensuring organized and effective resolution of issues.

ABSTRACTWITHPOs AND PSOsMAPPING

CO5:BUILD JAVAAPPLICATIONS FOR SOLVINGREAL-TIMEPROBLEMS.

ABSTRACT	Pos MAPPED	PSOs MAPPED
The HelpDesk Management System is a Java application designed to efficiently manage support tickets using a graphical user interface built with Swing. The system comprises two main classes: `HelpDeskSystem` and `Ticket`. The `HelpDeskSystem` class extends `JFrame` and creates the main window with components such as `JTextArea` for displaying tickets, `JTextField` for user input, and buttons to create, view, and close tickets. User interactions are handled by implementing the `ActionListener` interface, which triggers actions like ticket creation, viewing, and closure. The `Ticket` class represents individual tickets with a description and status (open or closed). Tickets are stored dynamically in an `ArrayList`, enabling efficient management and updates. This system provides a user-friendly interface for handling support tickets, ensuring organized and effective resolution of issues.	**PO1-3** **PO2-3** **PO3-3** **PO4-3** **PO5-3** **PO6-3** **PO7-3** **PO8-3** **PO9-3** **PO10-3** **PO11-3** **PO12-3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note:1-Low,2-Medium,3-High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of the HelpDesk Management System is to create a simple yet effective graphical user interface (GUI) application for managing help desk tickets. The system allows users to create, view, and close tickets, facilitating efficient tracking and resolution of support requests.

1.2 Overview

The HelpDesk Management System is built using Java and Swing for the GUI components. It maintains a list of tickets, each represented by a description and a status (open or closed). The system provides functionalities to:

i. Create new tickets.
ii. View the list of all tickets.
iii. Close existing tickets.

Users interact with the system through a GUI that includes text input fields and buttons to perform the aforementioned actions. The system uses an array list to store the tickets and updates the display dynamically as actions are performed.

1.3 Java Programming Concepts

The HelpDesk Management System applies fundamental concepts of Object-Oriented Programming (OOP) and specific Java-related techniques. It features two main classes: `HelpDeskSystem`, responsible for the GUI and overall functionality, and `Ticket`, which represents individual tickets. The system is built using Java Swing components such as `JFrame`, `JTextArea`, `JTextField`, `JButton`, `JPanel`, and `JScrollPane`. It manages event handling through action listeners attached to buttons, ensuring smooth user interactions. Additionally, an `ArrayList` is used to dynamically store and manage tickets. The system loosely follows the Model-View-Controller (MVC) pattern: `Ticket` acts as the model, while `HelpDeskSystem` serves as both the view and controller, handling the interaction and logic of the application. This design ensures a clear separation of concerns and promotes efficient handling of user requests within the system.

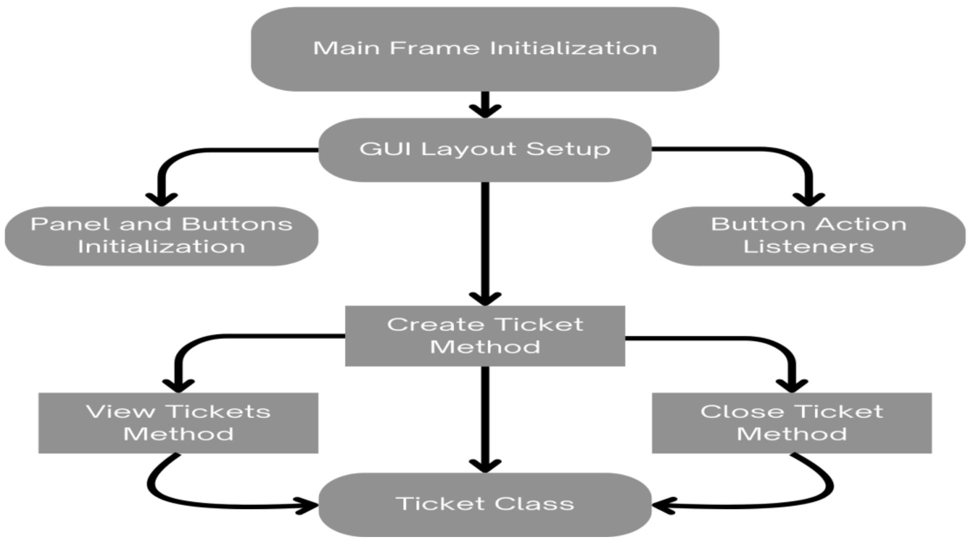
CHAPTER 2
PROJECT METHODOLOGY

2.1Proposed Work

The proposed HelpDesk Management System aims to provide a user-friendly application to efficiently manage help desk tickets. This system allows users to create, view, and close tickets through a graphical user interface (GUI) built using Java Swing. The application consists of two primary classes: `HelpDeskSystem` and `Ticket`. The `HelpDeskSystem` class extends `JFrame` to create the main window, incorporating components like `JTextArea`, `JTextField`, and `JButton` to facilitate user interactions. An `ArrayList` is employed to store ticket objects dynamically, enabling easy management of ticket data. Users can input ticket descriptions, which are added to the list and displayed in a non-editable text area. The system also provides functionalities to view all tickets and close specific ones, updating their status to "closed" and ensuring efficient tracking and resolution of support requests.

2.2 Block Diagram

The block diagram of the HelpDesk Management System is as follows: The main components include the `HelpDeskSystem` class, which acts as the primary controller and view, handling user input and displaying ticket information. The `Ticket` class represents individual tickets, encapsulating the description and status of each ticket. The `ArrayList<Ticket>` serves as the model, storing the ticket objects.



CHAPTER 3
MODULE DESCRIPTION

3.1 User Interface (UI)

This module handles the creation and layout of the user interface components. It uses variousSwing components like `JFrame`, `JTextArea`, `JTextField`, `JButton`, `JPanel`, and `JScrollPane`to build an intuitive interface where users can create, view, and close help desk tickets.

3.2 Ticket Management

This module manages the lifecycle of tickets, including creation, viewing, and closing. The`HelpDeskSystem` class maintains an `ArrayList` of `Ticket` objects, providing methods for adding newtickets, displaying all tickets, and updating the status of tickets as closed.

3.3 Event Handling

This module includes the implementation of event listeners for button actions. It uses the `ActionListener` interface to respond to user actions such as clicking the "Create Ticket", "View the user’s request.

3.4Data Storage

Tickets are stored in an `ArrayList` within the `HelpDeskSystem` class. This list dynamically maintains the tickets and allows for operations such as adding new tickets, closing existing ones, and displaying the current list of tickets with their statuses.

3.5Ticket Class

The `Ticket` class represents individual tickets with properties for description and status (open or closed). It includes methods to get the description, check if a ticket is closed, close a ticket, and convert the ticket information to a string format for display purposes.

CHAPTER 4
CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The HelpDesk Management System is an effective application designed to manage support tickets through a user-friendly interface built with Java Swing. It consists of two main classes: `HelpDeskSystem`, which handles the graphical user interface and ticket management, and `Ticket`, which represents individual tickets with descriptions and statuses. The system uses an `ArrayList` to store and manage tickets dynamically. Key functionalities include creating new tickets, viewing all tickets, and closing existing ones. User interactions are facilitated through buttons, and actions are handled by implementing the `ActionListener` interface. This structure ensures efficient and organized handling of help desk tickets, enhancing the overall support process.

4.2 FUTURE SCOPE

The HelpDesk Management System can be further enhanced by integrating features such as user authentication to restrict access, priority tagging for tickets to address critical issues promptly, and the ability to assign tickets to specific support agents. Additionally, incorporating a database to store tickets permanently and enabling email notifications for ticket updates can significantly improve the system's efficiency and user experience. Machine learning algorithms could also be integrated to predict and prioritize recurring issues, further optimizing the help desk operations.

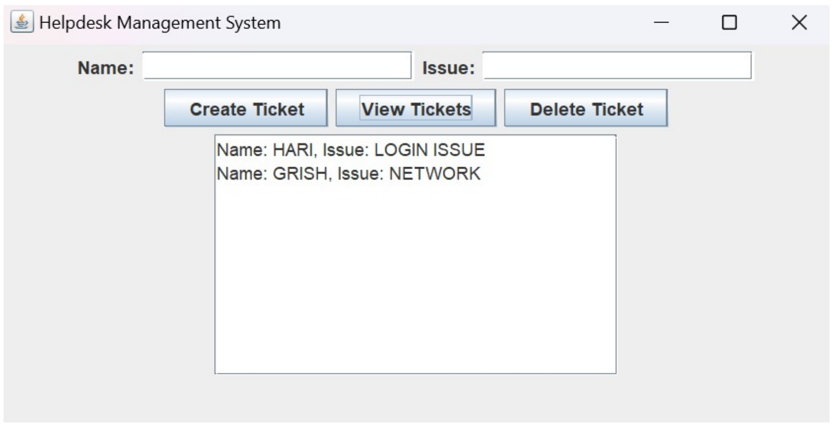
APPENDIX A
(SOURCE CODE)

```
# HelpDesk Management System

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class HelpDeskSystem extends JFrame {
    private ArrayList<Ticket> tickets;
    private JTextAreaticketArea;
    private JTextFieldticketInput;

    public HelpDeskSystem() {
        tickets = new ArrayList<>();
setTitle("HelpDesk Management System");
setSize(400, 300);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setLayout(new BorderLayout());

ticketArea = new JTextArea();
ticketArea.setEditable(false);
add(new JScrollPane(ticketArea), BorderLayout.CENTER);

JPanel panel = new JPanel();
ticketInput = new JTextField(20);
JButtoncreateButton = new JButton("Create Ticket");
JButtonviewButton = new JButton("View Tickets");
JButtoncloseButton = new JButton("Close Ticket");

createButton.addActionListener(new ActionListener() {
```

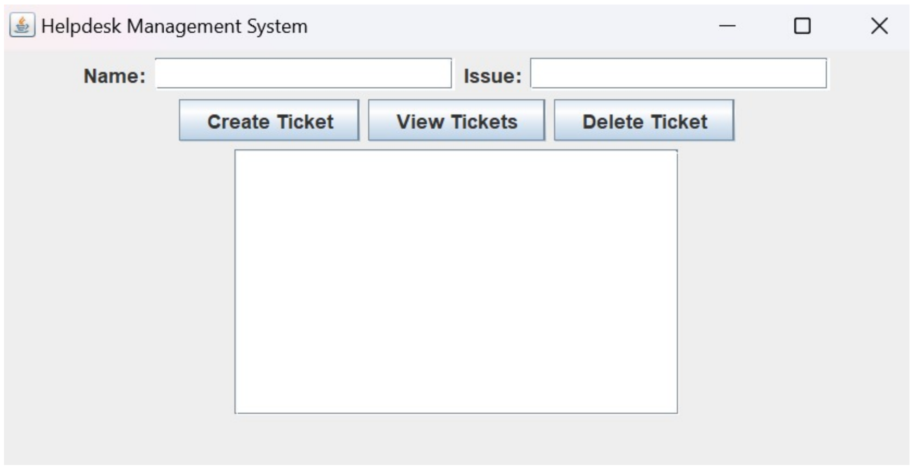
```
            public void actionPerformed(ActionEvent e) {
createTicket();
            }
        });

viewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
viewTickets();
            }
        });

closeButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
closeTicket();
            }
        });

panel.add(ticketInput);
panel.add(createButton);
panel.add(viewButton);
panel.add(closeButton);
add(panel, BorderLayout.SOUTH);
    }

    private void createTicket() {
        String ticketDescription = ticketInput.getText();
        if (!ticketDescription.isEmpty()) {
tickets.add(new Ticket(ticketDescription));
ticketInput.setText("");
JOptionPane.showMessageDialog(this, "Ticket Created!");
        } else {
JOptionPane.showMessageDialog(this, "Please enter a ticket description.");
        }
    }
    private void viewTickets() {
        StringBuilder sb = new StringBuilder();
        for (Ticket ticket : tickets) {
sb.append(ticket.toString()).append("\n");
        }
ticketArea.setText(sb.toString());
    }

    private void closeTicket() {
        String ticketDescription = ticketInput.getText();
        for (Ticket ticket : tickets) {
            if (ticket.getDescription().equals(ticketDescription) && !ticket.isClosed()) {
ticket.close();
ticketInput.setText("");
JOptionPane.showMessageDialog(this, "Ticket Closed!");
                return;
            }
        }
JOptionPane.showMessageDialog(this, "Ticket not found or already closed.");
    }

    public static void main(String[] args) {
SwingUtilities.invokeLater(() -> {
HelpDeskSystem system = new HelpDeskSystem();
system.setVisible(true);
        });
    }
}

class Ticket {
    private String description;
    private boolean closed;
    public Ticket(String description) {
this.description = description;
this.closed = false;
    }

    public String getDescription() {
        return description;
    }

    public booleanisClosed() {
        return closed;
    }

    public void close() {
        closed = true;
    }

    @Override
    public String toString() {
        return description + (closed ? " [CLOSED]" : " [OPEN]");
    }
}
```

APPENDIX B

(SCREENSHOTS)



```
Name: HARI, Issue: LOGIN ISSUE
Name: GRISH, Issue: NETWORK
```



REFERENCES

Java Concepts:

Java Swing Documentation: The official documentation for Java Swing components can be a great resource for understanding how to build GUI applications.
- [Java Swing Documentation](https://docs.oracle.com/javase/tutorial/uiswing/)

Event Handling in Java: Understanding how to handle events in Java Swing is crucial for building interactive applications.
- [Handling Events in Java](https://docs.oracle.com/javase/tutorial/uiswing/events/)

Java ArrayList: Learning about the ArrayList class and its methods for managing dynamic arrays in Java.
- [Java ArrayListDocumentation](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html)

Java GUI Programming: Tutorials and guides on creating GUI applications in Java.
- [Java GUI Programming Tutorials](https://www.tutorialspoint.com/java/java_swing.htm)

Links for coding part:

- https://zzzcode.ai/
- https://www.hesk.com/
- https://github.com/