DIGITAL NOTICE BOARD SYSTEM

A PROJECT REPORT

Submitted by

PRADEEP M (2302811724321081)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

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

SAMAYAPURAM – 621 112

DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **" DIGITAL NOTICE BOARD SYSTEM"** is the bonafide work of **PRADEEP(2302811724321081)** who carried out theproject work during the academic year 2024 - 2025 under my supervision.

/

Signature

Dr. T. AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,

K. Ramakrishnan College of Engineering,

Samayapuram, Trichy -621 112.

Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,

K. Ramakrishnan College of Engineering,

Samayapuram, Trichy -621 112.

SSubmitted for the viva-voce examination held on 3.12.24

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on " **DIGITAL NOTICE BOARD SYSTEM**" is the result of original work done by me and best of my knowledge, similar work has not been submitted to "**ANNA UNIVERSITY CHENNAI**" for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**

Signature

PRADEEP M

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

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

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.,** for having provided 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 our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,** Principal, whogave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D**., Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E**., Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her 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 providingvaluable information during the course.

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

VISION OF THE INSTITUTION

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

MISSION OF THE INSTITUTION

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

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

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 and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, 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, legal and cultural issues and the consequent responsibilities 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:** Apply ethical principles and commit to professional ethics and responsibilities and 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-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The "Digital Notice Board" is a simple Java-based graphical user interface (GUI) application that demonstrates the functionality of a notice board where users can display custom messages dynamically. Developed using the AWT (Abstract Window Toolkit), the program incorporates key concepts of event-driven programming and GUI design.

The application features a user-friendly interface comprising a text input field, a "Submit" button, and a label that serves as the notice display area. Users can type a message into the input field and click the "Submit" button to display their message on the notice board in real time. The displayed message is instantly updated, and the input field is cleared for subsequent entries.

The program also includes essential window management functionality, such as responding to the window close event, ensuring a seamless user experience. With its clean, efficient code structure, the "Digital Notice Board" serves as a foundational project for understanding interactive GUI applications in Java.

TABLE OF CONTENTS

CHAPTER 1

1.1 INTRODUCTION

The "Digital Notice Board" is a simple Java-based application that allows users to input messages and display them dynamically on a digital notice board interface. The application comprises three main components: a text field for message input, a submit button to trigger updates, and a label that serves as the notice board display area.When a user enters a message and clicks the "Submit" button, the label updates todisplay the new message, and the input field is cleared for subsequent entries. Theprogram handles user actions in real- time through event-driven programming andincludes window management to ensure smooth functionality.

1.2 OBJECTIVE

The objective of the "Digital Notice Board" project is to develop a user-friendly application that replicates the functionality of a traditional notice board using Java's Abstract Window Toolkit (AWT). The project aims to provide an interactive platform for dynamically updating messages, demonstrate real-time handling of user inputs, and apply principles of event-driven programming. It also seeks to enhance understanding of GUI design and window management while offering a practical implementation of Java programming concepts.

CHAPTER 2

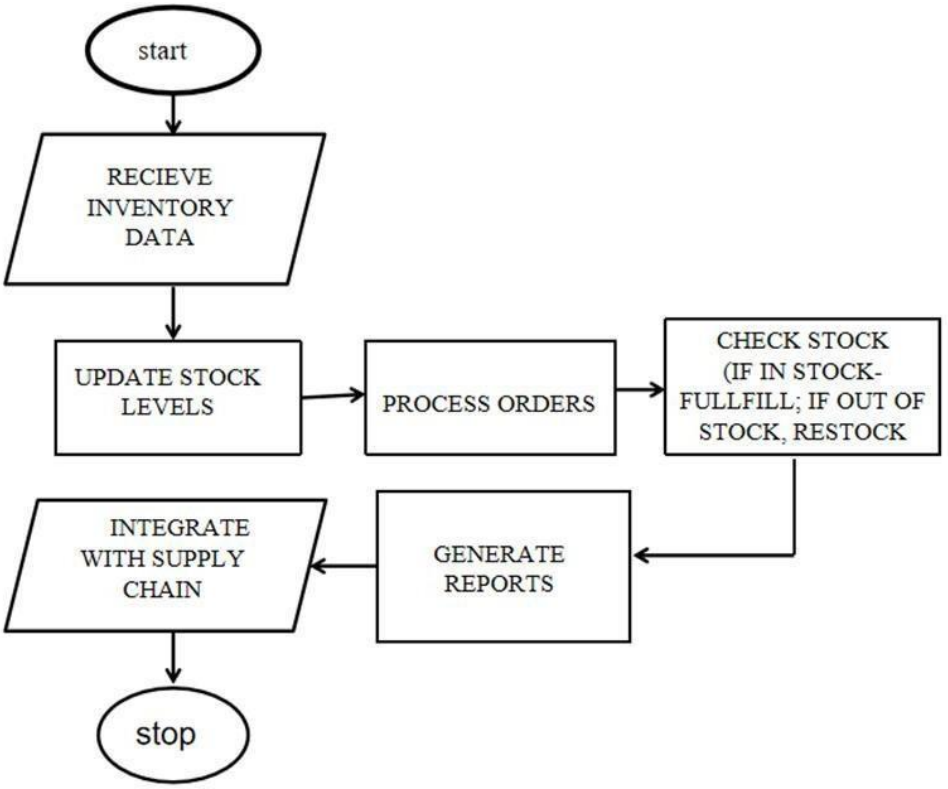
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed work for the "Digital Notice Board" project involves creating a user-friendly, Java-based application that simulates a digital version of a traditional notice board. The project will begin with requirement gathering, where the primary needs, such as user input, dynamic message updates, and an interactive GUI, will be identified. In the system design phase, a simple layout will be created using Java's Abstract Window Toolkit (AWT), incorporating components like a TextField for user input, a Button for submitting messages, and a Label for displaying the message.The next phase, implementation, will focus on coding the logic for event handling, specifically using the ActionListener interface to process user input and dynamically update the message displayed on the notice board. Additionally, window management will be implemented to handle the proper closure of the application.

Afterward, the project will undergo testing and debugging to ensure the smooth functioning of all features, including verifying that the message updates correctly, the button works as expected, and the window can be closed without errors. The successful implementation of these steps will result in a fully functional digital noticeboard that allows real- time message posting in a simple and interactive manner.

2.2 BLOCK DIAGRAM



CHAPTER 3
JAVA PROGRAMMING CONCEPTS

3.1 EVENT HANDLING

The program uses the ActionListener interface to handle user interactions, suchas button clicks, and the actionPerformed() method to define response acti0ons.

3.2 AWT (ABSTRACT WINDOW TOOLKIT)

It employs AWT components like TextField, Button, and Label to create an interactive graphical user interface.

3.3 LAYOUT MANAGEMENT

The FlowLayout manager is used to arrange components in a sequential andvisually organized manner within the application window.

3.4 WINDOW MANAGEMENT

The WindowAdapter class handles the window-closing event, ensuring the application closes gracefully when the user exits.

CHAPTER 4
MODULE DESCRIPTION

4.1 USER INPUTMODULE

The User Input Module is the first step in allowing users to interact with the digital notice board application. It enables users to enter messages using a TextField component, which acts as an input area where the user can type the text they wish to display on the notice board. The module ensures that the input is captured and stored for further processing. After the user types their message, they can submit it by clicking the "Submit" button. Once the submit action is triggered, the entered text is sent to the event-handling system for processing. This system validates the input and prepares it for display on the notice board. The module plays a crucial role in making the interface interactive, allowing users to add new content to the digital notice board in a straightforward and user-friendly manner. The design of this module focuses on creating an intuitive user experience, with clear input fields and simple controls to ensure seamless entry of messages. By isolating user input from other processing functions, the module ensures that the user's actions are effectively captured and passed on to the display system for presentation.

4.2 DISPLAY MODULE

The Display Module is responsible for presenting the messages entered by users on thedigital notice board. Once the user submits their message, this module is tasked with displaying the message in the main content area of the application. The Label component in the GUI serves as the primary element for this display, where the latest user message is dynamically updated. After each new submission, the Display Module refreshes the content area by updating the text of the Label to reflect the newly inputted message. It is crucial for this module to provide real-time updates, ensuring that users see their messages immediately after submission. Additionally, the module handles the input field reset, clearing the text input area so users can submit new messages without confusion. By ensuring that only one message is displayed at a time, it prevents clutter and keeps the notice board organized. This module also manages the dynamic nature of the application, ensuring smooth interaction between the user input and the visual

representation on the board. Overall, the Display Module plays a central role in updating and displaying content on the notice board and maintaining a clean, user-friendly interface.

4.3 WINDOW MANAGEMENT MODULE

The Window Management Module oversees the opening, closing, and proper disposal of the application window. This module is responsible for handling the lifecycle of the application window, ensuring that it opens correctly when the program is launched and closes smoothly when the user finishes interacting with it. The module uses the WindowAdapter class to listen for window events, such as when the user attempts to close the window. Upon detecting such an event, the module triggers appropriate actions to terminate the application gracefully, ensuring that no resources are left dangling and that the program exits without errors. It also handles any resizing or repositioning of the window during runtime, ensuring the GUI components are correctly displayed, regardless of the window's size. This is particularly important in maintaining the consistency of the layout and user experience, especially if the window is resized or adjusted. Furthermore, the Window Management Module plays a key role in managing any potential issues related to window visibility or layout adjustments. By effectively managing the window's life cycle, this module ensures the application operates smoothly and is both functional and responsive to user interaction

4.4 CONTENT MANAGEMENT MODULE

Enables creation, editing, and scheduling of notices. It ensures that all displayed content is relevant, updated, and formatted appropriately.

4.5 NOTIFICATION MODULE

Sends alerts or updates via email, SMS, or in-app notifications to ensure users stay informed even when away from the display**.**

CHAPTER 5

CONCLUSION

In conclusion, the Digital Notice Board application successfully demonstrates the use of Java Swing components to create a functional and interactive user interface. The application features a simple yet effective setup, where users can input messages using a JTextField, and the submitted messages are displayed dynamically on a digital notice board via a JLabel. The inclusion of a JButton allows the user to submit the message, triggering an ActionListener that processes the event and updates the display in real-time. This functionality makes the application a straightforward tool for digital communication, similar to a traditional notice board, but with the added benefits of interactivity and real-time updates.

The program's design highlights the power of basic Java Swing components like text fields, buttons, and labels for building a graphical user interface. It also demonstrates essential event-handling techniques, such as using ActionListener to respond to user actions.Although the application is simple, it provides a foundation for further development.There is potential to improve the design by adding more advanced features like input validation, message formatting, multi-user capabilities, and a more polished user interface. These enhancements would make the application more user-friendly and adaptable for real-world use, such as in offices or educational settings.

Furthermore, the Digital Notice Board application offers a practical example of how event- driven programming can be implemented in Java. By using the ActionListener to trigger specific actions when the user interacts with the application, the system efficiently handles user inputs and provides immediate feedback. This approach also enables easy scalability, allowing additional features tobe added, such as the ability to display multiple messages or even integrate with a

database for storing notices.

REFERENCES

Books:

- Head First Java by Kathy Sierra and Bert Bates.
- Java: The Complete Reference" by Herbert Schildt.
- Java Programming and Software Engineering Fundamentals by Duke University.

Websites:

- Java Documentation – https://docs.oracle.com/en/java/
- W3Schools Java Tutorial – https://www.w3schools.com/java/
- GeeksforGeeks Java Tutorials – https://www.geeksforgeeks.org/java/

YouTube Links:

- Java Programming for Beginners (YouTube Playlist by Programming with Mosh) Java Tutorial for Beginners - Programming with Mosh
- Java Inventory System Tutorial
  Build an Inventory System in Java
- Java GUI Tutorial
  Java Swing GUI Tutorial for Beginners

Other Resources:

- Stack Overflow – https://stackoverflow.com/questions/tagged/java
- Oracle Java Tutorials – https://docs.oracle.com/javase/tutorial/

APPENDIX

APPENDIX-A SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DigitalNoticeBoard {
    public static void main(String[] args) {
        JFrame frame = new JFrame("🌟 Digital Notice Board 🌟");
        frame.setSize(800, 600);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel noticePanel = new JPanel();
        noticePanel.setLayout(new BoxLayout(noticePanel, BoxLayout.Y_AXIS));
        JScrollPane scrollPane = new JScrollPane(noticePanel);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);

        JTextArea noticeInput = new JTextArea(3, 50);
        noticeInput.setFont(new Font("Arial", Font.PLAIN, 14));
        noticeInput.setBackground(new Color(255, 250, 205));
        noticeInput.setForeground(new Color(0, 0, 0));
        noticeInput.setLineWrap(true);
        noticeInput.setWrapStyleWord(true);

        JButton addButton = new JButton("➕ Add Notice");
        addButton.setFont(new Font("Arial", Font.BOLD, 14));
        addButton.setBackground(new Color(72, 201, 176));
        addButton.setForeground(Color.WHITE);
        addButton.setFocusPainted(false);
```

```
        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String noticeText = noticeInput.getText().trim();
                if (!noticeText.isEmpty()) {
                    JLabel noticeLabel = new JLabel("<html><p style='width: 500px;'>" +
noticeText + "</p></html>");
                    noticeLabel.setFont(new Font("Comic Sans MS", Font.PLAIN, 16));
                    noticeLabel.setBackground(new Color(255, 239, 200));
                    noticeLabel.setForeground(new Color(50, 50, 50));
                    noticeLabel.setOpaque(true);
                    noticeLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
                    noticeLabel.setPreferredSize(new Dimension(600, 40));
                    noticeLabel.setBorder(BorderFactory.createLineBorder(new Color(255, 165, 0),
2));

                    JPanel borderedPanel = new JPanel(new BorderLayout());
                    borderedPanel.setBackground(new Color(255, 250, 205));
                    borderedPanel.setPreferredSize(new Dimension(600, 50));
                    borderedPanel.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
Color.DARK_GRAY));
                    borderedPanel.add(noticeLabel, BorderLayout.CENTER);

                    noticePanel.add(borderedPanel);
                    noticePanel.revalidate();
                    noticePanel.repaint();
                    noticeInput.setText("");
                } else {
                    JOptionPane.showMessageDialog(frame, "Notice cannot be empty!", "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        });
```

```
        JPanel inputPanel = new JPanel(new BorderLayout());
        JLabel inputLabel = new JLabel(" 📝 Add Notice:");
        inputLabel.setFont(new Font("Arial", Font.BOLD, 14));
        inputLabel.setForeground(Color.WHITE);
        inputPanel.setBackground(new Color(34, 45, 65));
        inputPanel.add(inputLabel, BorderLayout.WEST);
        inputPanel.add(noticeInput, BorderLayout.CENTER);
        inputPanel.add(addButton, BorderLayout.EAST);

        frame.setLayout(new BorderLayout());
        frame.add(scrollPane, BorderLayout.CENTER);
        frame.add(inputPanel, BorderLayout.SOUTH);
        frame.getContentPane().setBackground(new Color(34, 45, 65));

        frame.setVisible(true);
    }
}
```

APPENDIX-B SCREEN SHOT