DIGITAL NOTICE BOARD SYSTEM

A PROJECT REPORT

Submitted by

A SACHIN KUMAR (2303811710421132)

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 – 621 112

NOVEMBER- 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 **A SACHIN KUMAR (2303811710421132)** 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

SIGNATURE

Mrs.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

CGB1201-JAVA PROGRAMMING
Mr.MALARMANNAN A, M.E.,
SUPERVISOR
ASSISTANTP ROFESSOR

SIGNATURE

Mr. A. Malarmannan, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 06-12-2024

CGB1201-JAVA PROGRAMMING
Mr. M. SARAVANAN, M.E.,
INTERNAL EXAMINER
ASSISTANTPROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Mr.R. KARTHIK, M.E.,
EXTERNAL EXAMINER
ASSISTANT PROFESSOR
8138-SCE, TRICHY.

06/12/2024

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“Digital Notice Board 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.**

Signature

A SACHIN KUMAR

Place: Samayapuram
Date: 06-12-2024

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 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 opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **MR. A. MALARMANNAN, 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 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 OF THE INSTITUTION

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

MISSION OF THE INSTITUTION

- 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 talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF 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.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

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

2. PEO2: Employability Skills and Research

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

3. PEO3: Ethics and Values

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

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

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

PSO 2: Quality Software

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

PSO 3: Innovation Ideas

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

PROGRAM OUTCOMES (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 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.

ABSTRACT

The **Digital Notice Board System** is an innovative approach to modernize the way announcements, notices, and updates are disseminated. Designed using **Java**, this system leverages the power of digital communication to replace traditional paper-based notice boards with a dynamic and efficient alternative.

The system allows administrators to easily create, update, and manage notices through a user-friendly interface. Notices are displayed on digital screens in real-time, ensuring timely communication. The application incorporates features like **category-based notices**, **scheduled updates**, and **remote accessibility**, enabling administrators to broadcast relevant information effectively.

The system employs **client-server architecture** where the server handles notice creation and distribution, while the client devices (digital boards) fetch and display the data seamlessly. Additional functionalities such as **user authentication**, **priority-based notice sorting**, and **multimedia support** (text, images, videos) enhance its versatility.

This project is particularly beneficial for institutions, offices, and public spaces where maintaining an up-to-date and eco-friendly communication system is crucial. The use of Java ensures platform independence, scalability, and robustness, making it a reliable solution for real-time information sharing.

By minimizing paper waste and reducing manual labor, this system not only supports environmental sustainability but also enhances the overall efficiency of communication processes.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Digital Notice Board system is a software application designed to replace traditional notice boards with a dynamic digital solution. It allows authorized users to post, update, and delete notices, ensuring efficient and organized communication. Notices are displayed on a digital interface, making it accessible to the intended audience in real-time. The system provides user-friendly functionality and enhances the notice management process.	**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**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

- **Efficient Information Sharing:** Provide a platform for real-time dissemination of notices and updates to reduce manual efforts and ensure timely communication.
- **Eco-Friendly Solution:** Minimize the use of paper by replacing traditional notice boards with a digital and environmentally sustainable alternative.
- **User-Friendly Management:** Enable administrators to easily create, schedule, and manage notices through an intuitive and accessible interface.
- **Engaging Content Display:** Support multimedia formats (text, images, videos) to make notices more engaging and visually appealing for viewers.

1.2 Overview

The **Digital Notice Board System** is a modernized solution designed to replace traditional notice boards with a digital, interactive platform for disseminating information. Developed in **Java**, the system offers a dynamic and eco-friendly way to display notices, announcements, and updates in real time.

This system is intended for use in various environments, such as educational institutions, offices, and public spaces, where frequent communication is essential. The platform provides administrators with tools to create, edit, and schedule notices easily through a centralized interface. Notices can include text, images, and videos, enhancing their visual appeal and effectiveness.

The system operates on a **client-server architecture**, where the server manages the notice database and pushes updates to connected digital screens (clients). Security features like user authentication ensure that only authorized personnel can manage notices.

By reducing dependency on paper and manual labor, the Digital Notice Board System promotes sustainability while increasing the efficiency and speed of information sharing. Its modular design ensures flexibility, scalability, and ease of use, making it an essential tool for modern communication needs.

1.3 Java Programming Concepts

1)Object-Oriented Programming (OOP)

- Use of classes and objects to represent components like admin, notices, and display devices.
- Encapsulation, inheritance, and polymorphism to ensure modular and reusable code.

2)Classes and Objects: To represent notices and manage their properties.

3)ArrayList: Used to dynamically store and manage a list of notices.

4Control Structures: Loops and conditional statements to handle user inputs and system functionality.

5)Scanner Class: For reading user inputs.

6)Encapsulation: To manage notice data securely through object properties.

7)Method Overriding (toString): To display notice details in a readable format.

CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

1. Notice Management Module

- Handles CRUD (Create, Read, Update, Delete) operations for notices.
- Ensures each notice is uniquely identified using an auto-incremented ID.

2. User Interaction Module

- Provides a menu-driven interface for users to post, update, delete, or view notices.
- Uses the Scanner class for input handling.

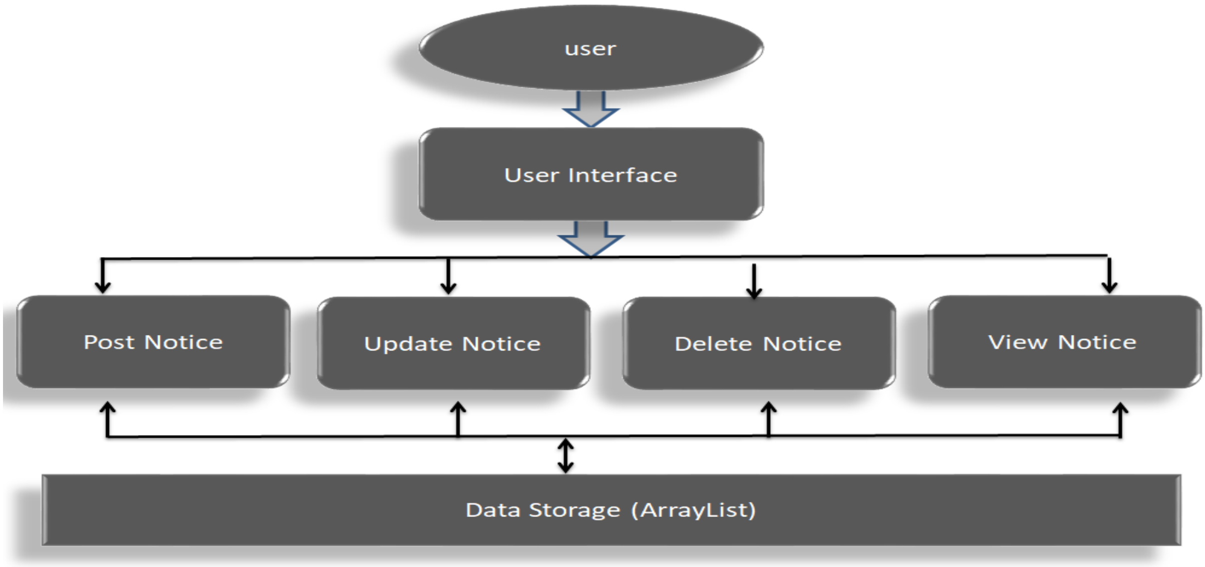
3. Data Storage and Retrieval Module

- Utilizes ArrayList to store notices dynamically in memory.
- Facilitates efficient retrieval of notices by their unique IDs.

4. Display Module

- Outputs all existing notices for public viewing.
- Displays user-friendly messages for system operations (e.g., successful updates or errors).

2.2 Block Diagram



CHAPTER 3
MODULE DESCRIPTION

3.1 Module 1 Notice Management Module

- **CRUD Operations:** This module provides functionality to create, read, update, and delete notices. These operations are the core actions needed to manage the content displayed on the notice board.
- **Unique Identification:** Each notice is assigned a unique ID using an auto-incrementing mechanism. This ensures notices can be efficiently tracked and retrieved without ambiguity.
- **Notice Integrity:** Ensures data integrity by validating inputs (e.g., checking for empty content or invalid updates) before processing the notices.

3.2 Module 2 User Interaction Module

- **Interactive Interface:** Offers a menu-driven interface where users can navigate options to post, update, delete, or view notices. This ensures ease of use for both technical and non-technical users.
- **Input Handling:** Employs the Scanner class to handle user inputs effectively, ensuring that options and notice details are captured accurately.

3.3 Module 3 Data Storage and Retrieval Module

- **Dynamic Storage:** Uses an ArrayList to dynamically store notices in memory, allowing for flexible and scalable notice management as the system grows.
- **Efficient Retrieval:** Ensures quick access to notices using their unique IDs, making updates and deletions seamless and precise.
- **Data Organization:** Maintains a structured collection of notices, making it easy to iterate through the list for viewing or modifying specific entries.

3.4Module 4 Display Module

- **Notice Output:** Displays all stored notices in a clear and readable format, ensuring that the audience can easily understand the information presented.
- **User-Friendly Messages:** Shows contextual messages for operations such as successful updates, invalid inputs, or errors, guiding users through the system’s functionality.

CHAPTER 4
CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The **Digital Notice Board System** is a modern, efficient, and eco-friendly solution to traditional notice boards, leveraging the power of Java to provide dynamic and real-time communication. By integrating features such as notice management, user interaction, and dynamic data storage, the system ensures seamless handling of notices and their efficient display to the intended audience.

With its user-friendly interface, robust backend, and focus on sustainability, the system addresses the challenges of manual notice handling, reducing paper usage and enhancing communication efficiency. The use of object-oriented design and modular architecture ensures scalability and adaptability for future enhancements.

In conclusion, the **Digital Notice Board System** not only improves the way information is disseminated but also contributes to a sustainable and technologically advanced approach to communication in various environments such as schools, offices, and public spaces. It lays the foundation for further innovation, such as mobile integration and analytics, making it a vital tool for modern information systems.

4.2 FUTURE SCOPE

The future scope of a digital notice board in Java includes integration with IoT for smart updates, cloud services for remote management, and multi-platform accessibility through mobile and web interfaces. AI can personalize notices, while enhanced security ensures safe data transmission. Features like multimedia support, automation, and scheduling improve functionality, and analytics provide insights for better management. Integration with external systems and support for multiple languages further expands its usability, making it ideal for educational institutions, corporations, and smart city projects.

APPENDIX A

(SOURCE CODE)

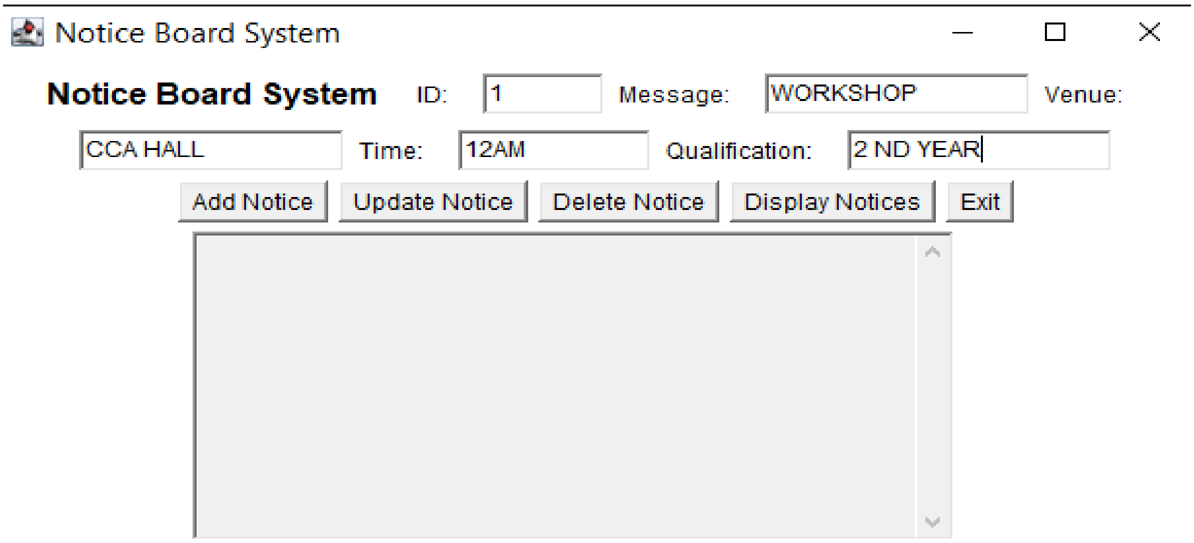
```
import java.awt.*;
import java.awt.event.*;
import java.util.HashMap;

class Notice {
    private String message;
    private String venue;
    private String time;
    private String qualification;

    public Notice(String message, String venue, String time, String qualification) {
        this.message = message;
        this.venue = venue;
        this.time = time;
        this.qualification = qualification;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getVenue() {
        return venue;
    }

    public void setVenue(String venue) {
        this.venue = venue;
    }

    public String getTime() {
        return time;
    }

    public void setTime(String time) {
        this.time = time;
    }

    public String getQualification() {
        return qualification;
    }

    public void setQualification(String qualification) {
        this.qualification = qualification;
    }

    @Override
    public String toString() {
        return "Message: " + message + ", Venue: " + venue + ", Time: " + time + ", Qualification: 
" + qualification;
    }
}

public class NoticeBoardSystemAWT extends Frame {
    private HashMap<Integer, Notice> notices = new HashMap<>();
    private int noticeIdCounter = 1;
    private TextArea displayArea;
    private TextField idField, messageField, venueField, timeField, qualificationField;
    private Label messageLabel;

    public NoticeBoardSystemAWT() {
        // Layout and Title
        setLayout(new FlowLayout());
        Label title = new Label("Notice Board System");
        title.setFont(new Font("Arial", Font.BOLD, 16));
        add(title);

        // Input Fields
        add(new Label("ID:"));
        idField = new TextField(5);
        add(idField);

        add(new Label("Message:"));
        messageField = new TextField(15);
        add(messageField);

        add(new Label("Venue:"));
        venueField = new TextField(15);
        add(venueField);

        add(new Label("Time:"));
        timeField = new TextField(10);
        add(timeField);

        add(new Label("Qualification:"));
        qualificationField = new TextField(15);
        add(qualificationField);

        // Buttons
        Button addButton = new Button("Add Notice");
        Button updateButton = new Button("Update Notice");
        Button deleteButton = new Button("Delete Notice");
        Button displayButton = new Button("Display Notices");
        Button exitButton = new Button("Exit");
        add(addButton);
        add(updateButton);
        add(deleteButton);
        add(displayButton);
        add(exitButton);

        // Display Area
        displayArea = new TextArea(10, 50);
        displayArea.setEditable(false);
        add(displayArea);

        // Message Label
        messageLabel = new Label(" ");
        messageLabel.setForeground(Color.RED);
        add(messageLabel);

        // Button Actions
        addButton.addActionListener(e -> addNotice());
        updateButton.addActionListener(e -> updateNotice());
        deleteButton.addActionListener(e -> deleteNotice());
        displayButton.addActionListener(e -> displayNotices());
        exitButton.addActionListener(e -> System.exit(0));

        // Frame Settings
        setSize(600, 400);
        setTitle("Notice Board System");
        setVisible(true);

        // Close Window Action
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    private void addNotice() {
        try {
            String message = messageField.getText();
            String venue = venueField.getText();
            String time = timeField.getText();
            String qualification = qualificationField.getText();

            if (message.isEmpty() || venue.isEmpty() || time.isEmpty() || qualification.isEmpty())
{
                messageLabel.setText("Error: All fields are required!");
                return;
            }
            notices.put(noticeIdCounter++, new Notice(message, venue, time, qualification));
            messageLabel.setText("Notice added successfully!");
            clearFields();
        } catch (Exception ex) {
            messageLabel.setText("Error: Invalid input!");
        }
    }

    private void updateNotice() {
        try {
            int id = Integer.parseInt(idField.getText());
            if (!notices.containsKey(id)) {
                messageLabel.setText("Notice not found!");
                return;
            }

            Notice notice = notices.get(id);
            notice.setMessage(messageField.getText());
            notice.setVenue(venueField.getText());
            notice.setTime(timeField.getText());
            notice.setQualification(qualificationField.getText());

            messageLabel.setText("Notice updated successfully!");
            clearFields();
        } catch (Exception ex) {
            messageLabel.setText("Error: Invalid input!");
        }
    }

    private void deleteNotice() {
        try {
            int id = Integer.parseInt(idField.getText());
            if (notices.remove(id) != null) {
                messageLabel.setText("Notice deleted successfully!");
            } else {
                messageLabel.setText("Notice not found!");
            }
            clearFields();
        } catch (Exception ex) {
            messageLabel.setText("Error: Invalid input!");
        }
    }

    private void displayNotices() {
        if (notices.isEmpty()) {
            displayArea.setText("No notices available.");
        } else {
            StringBuilder builder = new StringBuilder();
            notices.forEach((id, notice) -> builder.append("ID: ").append(id).append(",
").append(notice).append("\n"));
```

```
            displayArea.setText(builder.toString());
        }
    }

    private void clearFields() {
        idField.setText("");
        messageField.setText("");
        venueField.setText("");
        timeField.setText("");
        qualificationField.setText("");
    }

    public static void main(String[] args) {
        LoginFrame loginFrame = new LoginFrame();
    }
}

class LoginFrame extends Frame {
    private TextField usernameField, passwordField;
    private Label messageLabel;
    public LoginFrame() {
        setLayout(new FlowLayout());
        setTitle("Login");
        // Username
        add(new Label("Username:"));
        usernameField = new TextField(15);
        add(usernameField);
        // Password
        add(new Label("Password:"));
        passwordField = new TextField(15);
        passwordField.setEchoChar('*');
        add(passwordField);
        // Buttons
        Button loginButton = new Button("Login");
        Button exitButton = new Button("Exit");
        add(loginButton);
        add(exitButton);
        // Message Label
        messageLabel = new Label(" ");
        messageLabel.setForeground(Color.RED);
        add(messageLabel);
        loginButton.addActionListener(e -> validateLogin());
        exitButton.addActionListener(e -> System.exit(0));
        setSize(300, 200);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
    private void validateLogin() {
        String username = usernameField.getText();
        String password = passwordField.getText();

        // Simple authentication (hardcoded)
        if ("admin".equals(username) && "password".equals(password)) {
            messageLabel.setText("Login successful!");
            dispose(); // Close the login window
            new NoticeBoardSystemAWT(); // Open the Notice Board System
        } else {
            messageLabel.setText("Invalid credentials!");
        }
    }
}
```

APPENDIX B

(SCREENSHOTS)

1) GIVING INPUTS



Notice Board System

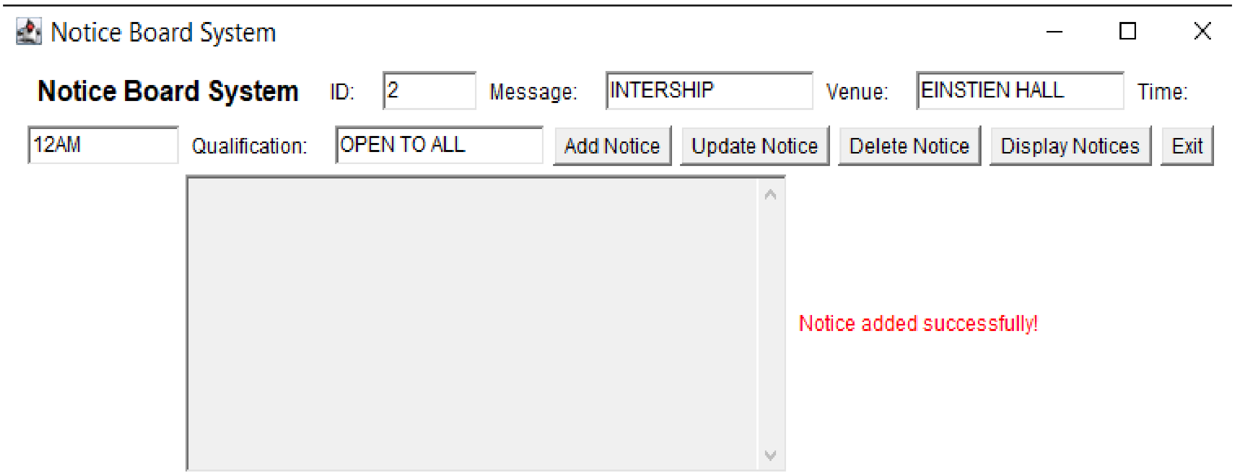
ID: 1

Message: WORKSHOP

Venue: CCA HALL

Time: 12AM

Qualification: 2 ND YEAR



Notice Board System

ID: 2

Message: INTERSHIP

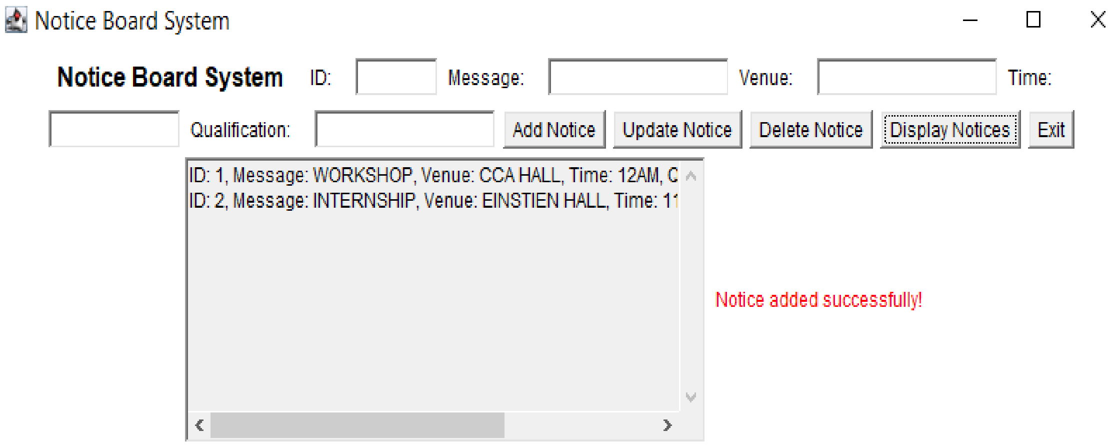
Venue: EINSTIEN HALL

Time: 12AM

Qualification: OPEN TO ALL

Notice added successfully!

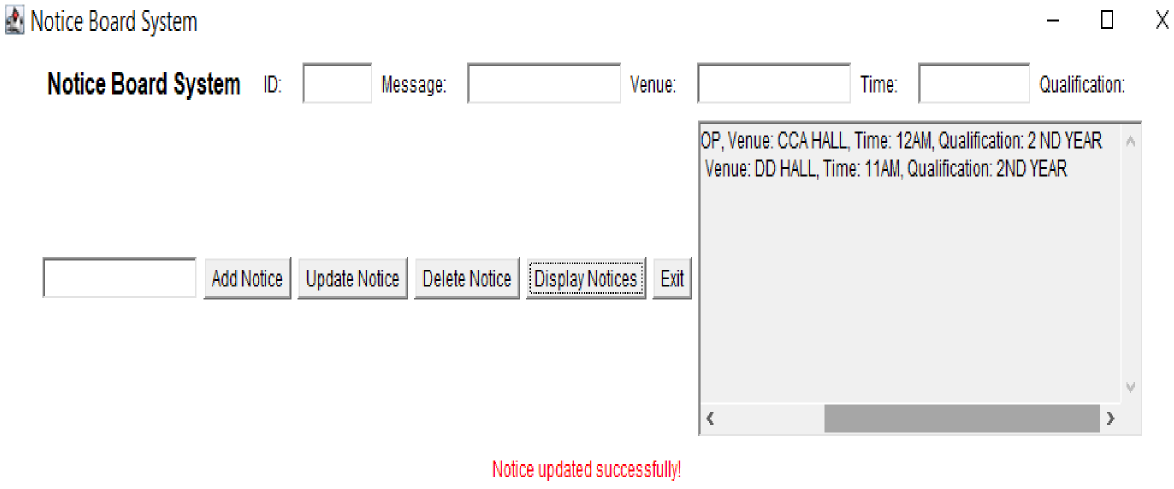
2)BEFORE UPDATE



```
ID: 1, Message: WORKSHOP, Venue: CCA HALL, Time: 12AM, C
ID: 2, Message: INTERNSHIP, Venue: EINSTIEN HALL, Time: 11
```

Notice added successfully!

3)AFTER UPDATE

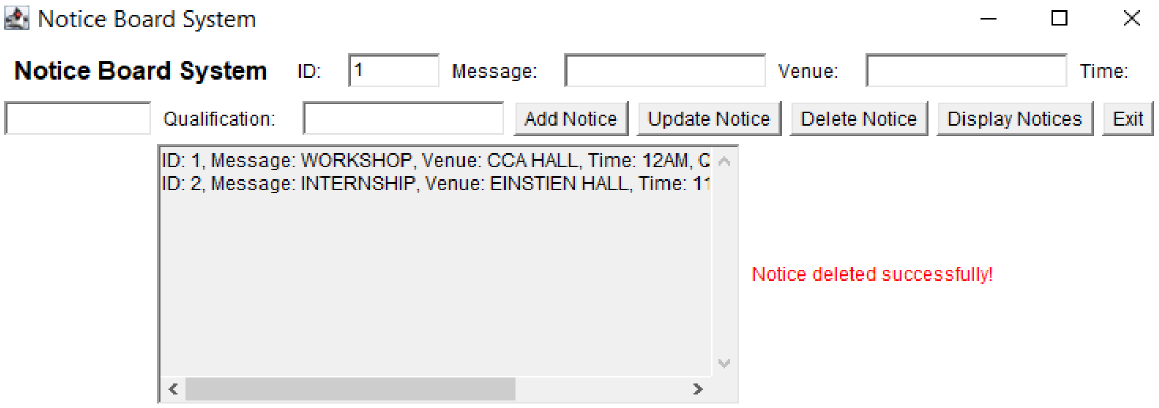


```
OP, Venue: CCA HALL, Time: 12AM, Qualification: 2 ND YEAR
 Venue: DD HALL, Time: 11AM, Qualification: 2ND YEAR
```

Notice updated successfully!

4)DELETING NOTICES

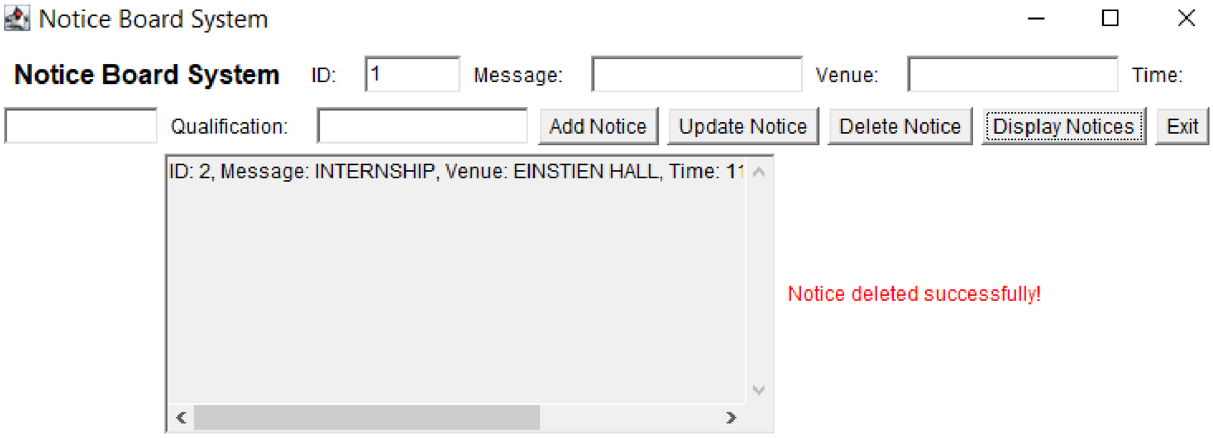
(input of id number to delete)



1

```
ID: 1, Message: WORKSHOP, Venue: CCA HALL, Time: 12AM, C
ID: 2, Message: INTERNSHIP, Venue: EINSTIEN HALL, Time: 11
```

Notice deleted successfully!



1

```
ID: 2, Message: INTERNSHIP, Venue: EINSTIEN HALL, Time: 11
```

Notice deleted successfully!

(result)

REFERENCES

1. Bhardwaj, Gaurav, and Gunjan Sahu. "IOT based Smart Notice Board." International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 6, 2020. IOT based Smart Notice Board .
2. Sharma, Dharmendra Kumar, and Vineet Tiwari. "Small and medium range wireless electronic notice board using Bluetooth and ZigBee." IEEE, 2015. Digital Notice Board (A Mobile Application)
3. Khandagale, A.S., et al. "Design E - Notice Board using Effective web Technologies for Educational Organizations." International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), vol. 9, no. 3, 2020.35_Design_JP (1).PDF