ALUMNI MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

SUBIKSEN V S (2303811710421160)

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 **“ALUMNI MANAGEMENT SYSTEM”** is the bonafide work of **SUBIKSEN V S (2303811710421160)** 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 – 06.12.24

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

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 **“ALUMNI 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.**

.

Signature

Subi

SUBIKSEN V S

Place: Samayapuram

Date: 6.12.24

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

1.Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.

2.Be an institute with world class research facilities

3.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 Alumni Management System is a comprehensive platform designed to streamline communication and foster meaningful interactions between educational institutions and their alumni. By centralizing alumni-related activities, the system ensures that institutions can maintain long-term relationships with their graduates. Its primary purpose is to facilitate seamless engagement through the management of alumni information, event organization, and career services. This helps both alumni and institutions. benefit from continued networking and professional growth opportunities.

One of the standout features of the system is its ability to manage critical alumni data, organize various alumni-centric events, and provide job opportunity listings. Built using Java and AWT, the system leverages object-oriented programming principles to ensure a modular and scalable design. The graphical user interface (GUI) is intuitive and user-friendly, providing centralized options that allow users to quickly navigate through various functionalities, thereby improving overall user experience.

To enhance system robustness, custom exception handling is implemented, ensuring that operations remain user-friendly even when errors occur. The system also includes an update mechanism that allows alumni to keep their contact information current, ensuring the data remains relevant. Furthermore, the system is designed with future enhancements in mind, offering the flexibility to integrated features .

ABSTRACT WITH POs AND PSOs MAPPING

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

ABSTRACT	POs MAPPED	PSOs MAPPED
The **Alumni Management System** is a robust application designed to streamline alumni engagement and foster long-term institutional relationships. This system aligns with PSO 1 by leveraging foundational Computer Science principles to analyze, design, and implement efficient solutions for alumni data management, event coordination, and job postings. PSO 2 is realized through adherence to software engineering best practices, ensuring high-quality, scalable, and reliable software. In line with PSO 3, the system integrates emerging ICT tools, such as search and analytics, to drive innovation in connecting alumni and institutions. The project achieves key POs by incorporating advanced problem analysis, modern tool usage, ethical standards, teamwork, and sustainability, all while fostering continuous learning and effective communication.	**PO1-3** **PO2-3** **PO3-3** **PO5-3** **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 Alumni Management System is to provide an efficient and user-friendly platform to manage alumni information, organize events, create and track job opportunities, and analyze data for institutional development. By incorporating modules such as alumni management, event management, job opportunity tracking, user interface with search and analytics, the system aims to foster better communication and engagement between institutions and their alumni. Additionally, the system seeks to improve networking opportunities for alumni, support career growth, and enable easy access to alumni-related events and job postings. Through seamless integration of these features, the system enhances the overall alumni experience and strengthens the connection between alumni and their alma mater. The primary goal is to offer an effective, sustainable solution for alumni relations, contributing to the institution's long-term success and growth.

1.2 Overview

The Alumni Management System is a comprehensive platform designed to connect educational institutions and their alumni. It facilitates efficient management of alumni information, organization, and job opportunity postings. The system features five core modules: Alumni Management, Event Management, Job Opportunity Tracking, User Interface with Input Validation, and Search & Analytics. These modules work collaboratively to ensure seamless operations, promote alumni engagement, and support professional networking. With a user-friendly interface and robust data handling capabilities, the system empowers institutions to foster long-term relationships with their alumni while supporting institutional and career development initiatives. Additionally, it helps in maintaining a database of alumni with accurate information, providing a centralized point for networking opportunities. The system is designed to offer flexibility, enabling institutions to modify or update the information based on evolving needs, thus enhancing the management of alumni relations. Furthermore, with its advanced search and analytics features, the system delivers actionable insights to guide decision-making and improve alumni engagement strategies.

1.3 Java Programming Concepts

Object-Oriented Programming (OOP)

- **Encapsulation:** Each module (e.g., Alumni,Event, JobOpportunity) is enclosed in a separate class with private fields and public methods.
- **Inheritance:** Extending shared functionalities, though not directly visible, can be implemented for common features across entities.

Collection Framework

- **ArrayList:** Used for managing dynamic lists of alumni , events , and job opportunities and events

Event Handling

- **ActionListener:** Used for handling button clicks to perform specific actions while the interaction.
- **WindowAdapter:** It is used to capture the window event in closing the window on the screen.

Graphical User Interface (GUI)

- **AWT (Abstract Window Toolkit):** Utilized for building the components of the user interface.
- **Layouts:** Use of FlowLayout and manual positioning to align elements on the frame of the window.

Core Java Fundamentals

- **Basic I/O operations :** for capturing information from the user to display it on the dialogs box.
- **String manipulation :** for processing user input and displaying results on screen of the window.

CHAPTER 2
PROJECT METHODOLOGY

2.1 Proposed Work

The **Alumni Management System** serves as a comprehensive platform to maintain a centralized database of alumni, storing essential details such as names, graduation years, professions, and contact information, with powerful search functionality for easy access based on criteria like profession or graduation year. It efficiently manages alumni events, providing features to organize, view, and manage event details, including names, dates, and descriptions. The system facilitates seamless posting and browsing of job opportunities, offering key information such as job titles, company names, locations, and descriptions. With a user-friendly interface, it ensures smooth interaction through intuitive GUI elements like buttons and prompts, while robust input validation maintains data accuracy. Security measures safeguard sensitive alumni data, ensuring privacy and integrity in every operation.

2.2 Block Diagram

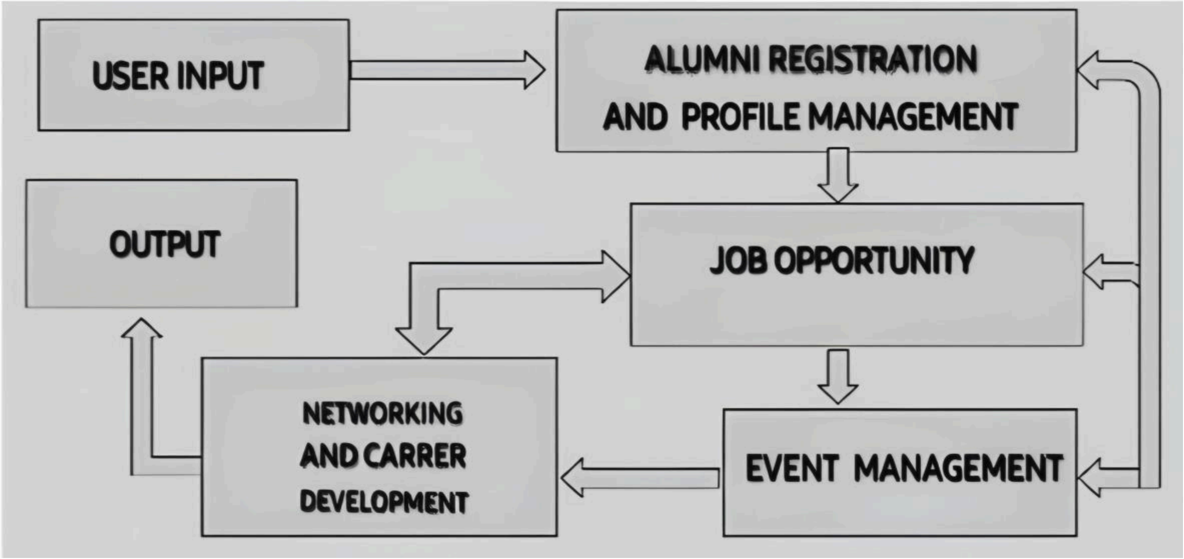

USER INPUT
ALUMNI REGISTRATION AND PROFILE MANAGEMENT
OUTPUT
JOB OPPORTUNITY
NETWORKING AND CARRER DEVELOPMENT
EVENT MANAGEMENT

Fig 2.2 Block Diagram of Alumni Management System

CHAPTER 3
MODULE DESCRIPTION

1.1 Alumni Management Module

This module is responsible for managing alumni information, such as names, graduation years, contact details, and professions. It provides functionality to add, update, search, and display alumni details, ensuring efficient record-keeping and easy retrieval.

1.2 Event Management Module

This module focuses on organizing and managing alumni-related events, including meetings, reunions, and seminars. Users can add, edit, and view event details like event name, date, and description. It ensures better alumni engagement and networking.

1.3 Job Opportunity Module

Dedicated to posting and managing job opportunities for alumni or students. It allows users to add job details like title, company, location, and description, while also providing an option to browse through available opportunities.

1.4 User Interface & Input Validation Module

This module ensures a user-friendly graphical interface for seamless interaction with the system's features. Input validation is integrated to prevent invalid data entries and maintain the system's reliability and accuracy.

1.5 Search and Analytics Module

This module provides advanced search functionalities to locate specific alumni, events, or job opportunities based on various criteria like profession, graduation year, or company. It includes analytical tools to generate insights, such as the number of alumni in specific industries, popular event types, or trends in job postings, helping stakeholders make informed decisions.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The Alumni Management System successfully integrates key modules like Alumni Management, Event Management, Job Opportunities, User Interface & Input Validation, and Search & Analytics to streamline operations for educational institutions. By leveraging Java's robust features and implementing a user-friendly interface, the system enhances communication, facilitates event coordination, and provides job-related updates for alumni. This project demonstrates the application of software engineering principles, delivering a scalable and efficient solution. Future improvements could include incorporating advanced analytics and cloud integration to further enhance system capabilities.

4.2 FUTURE SCOPE

The Alumni Management System has significant potential for future enhancements. It can integrate with cloud platforms for real-time data access and scalability, while a mobile application can improve accessibility and user engagement. Advanced analytics and predictive tools can offer insights into alumni trends, aiding in strategic decisions. Social media integration will enhance networking and event promotions. AI-driven chatbots can provide instant support, and personalized dashboards can cater to individual user needs. Expanding to a global alumni network and incorporating enhanced security measures will improve connectivity and data protection. Gamification can boost engagement, while automated notifications streamline communication for events and job opportunities.

APPENDIX A

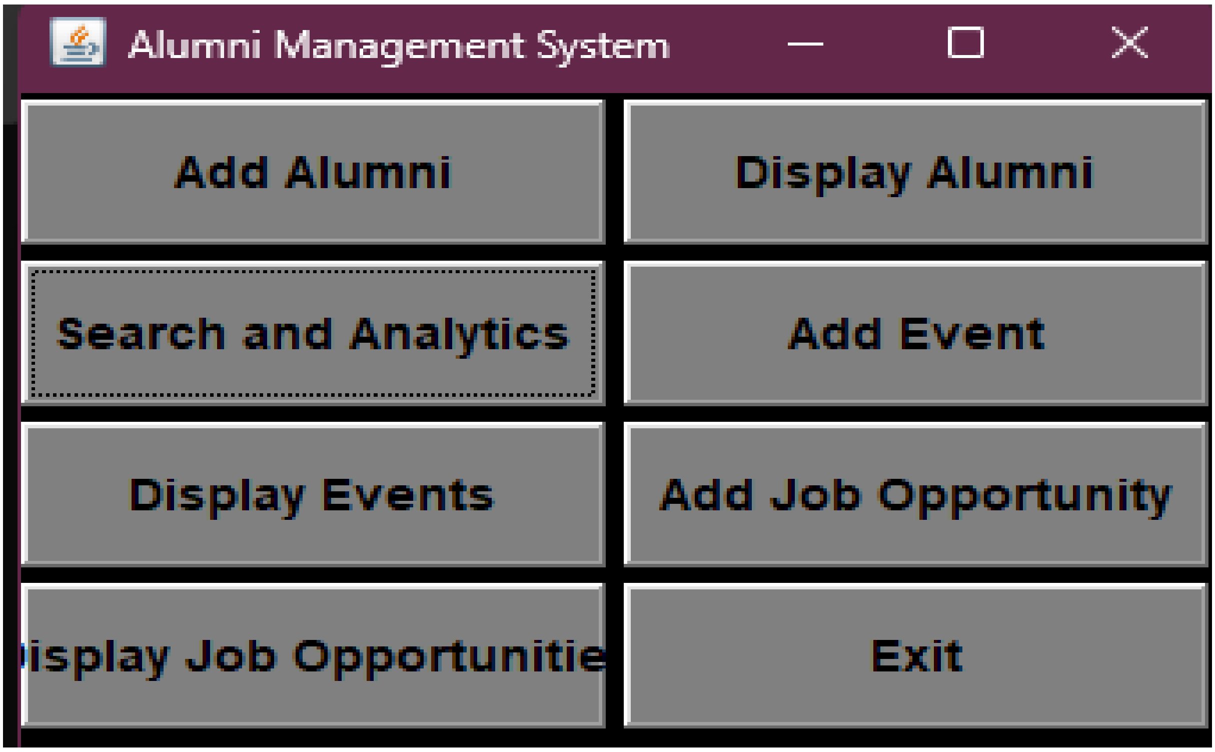
(Project Source Code)

```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;

public class AlumniManagementSystem extends Frame {

    private List<Alumni> alumniList = new ArrayList<>();
    private List<Event> eventList = new ArrayList<>();
    private List<JobOpportunity> jobOpportunities = new ArrayList<>();

    public AlumniManagementSystem() {
        setLayout(new GridLayout(6, 1, 5, 5)); // Align options centrally in a grid
        setBackground(Color.BLACK);

        // Buttons for modules
        Button addAlumniButton = createButton("Add Alumni");
        Button displayAlumniButton = createButton("Display Alumni");
        Button searchAnalyticsButton = createButton("Search and Analytics");
        Button addEventButton = createButton("Add Event");
        Button displayEventsButton = createButton("Display Events");
        Button addJobOpportunityButton = createButton("Add Job Opportunity");
        Button displayJobsButton = createButton("Display Job Opportunities");
        Button exitButton = createButton("Exit");

        // Adding buttons to the frame
        add(addAlumniButton);
        add(displayAlumniButton);
        add(searchAnalyticsButton);
        add(addEventButton);
        add(displayEventsButton);
        add(addJobOpportunityButton);
        add(displayJobsButton);
        add(exitButton);

        // Button listeners
        addAlumniButton.addActionListener(e -> addAlumni());
        displayAlumniButton.addActionListener(e -> displayAlumni());
        searchAnalyticsButton.addActionListener(e -> searchAndAnalytics());
        addEventButton.addActionListener(e -> addEvent());
        displayEventsButton.addActionListener(e -> displayEvents());
        addJobOpportunityButton.addActionListener(e -> addJobOpportunity());
        displayJobsButton.addActionListener(e -> displayJobOpportunities());
        exitButton.addActionListener(e -> System.exit(0));

        setTitle("Alumni Management System");
        setSize(500, 500);
        setVisible(true);

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

    private Button createButton(String label) {
        Button button = new Button(label);
        button.setBackground(Color.GRAY);
        button.setForeground(Color.BLACK);
        button.setFont(new Font("Arial", Font.BOLD, 14));
        return button;
    }

    // Alumni Management Module
    private void addAlumni() {
        String name = prompt("Enter Alumni Name:");
        int year = Integer.parseInt(prompt("Enter Graduation Year:"));
        String contact = prompt("Enter Contact Number:");
        String profession = prompt("Enter Profession:");
        alumniList.add(new Alumni(name, year, contact, profession));
        showMessage("Alumni added successfully.");
    }

    private void displayAlumni() {
        if (alumniList.isEmpty()) {
            showMessage("No Alumni found.");
        } else {
            StringBuilder sb = new StringBuilder("Alumni List:\n");
            for (Alumni alumni : alumniList) {
                sb.append(alumni).append("\n");
            }
            showMessage(sb.toString());
        }
    }

    // Event Management Module
    private void addEvent() {
        String name = prompt("Enter Event Name:");
        String date = prompt("Enter Event Date:");
        String description = prompt("Enter Event Description:");
        eventList.add(new Event(name, date, description));
        showMessage("Event added successfully.");
    }

    private void displayEvents() {
        if (eventList.isEmpty()) {
            showMessage("No Events found.");
        } else {
            StringBuilder sb = new StringBuilder("Event List:\n");
            for (Event event : eventList) {
                sb.append(event).append("\n");
            }
            showMessage(sb.toString());
        }
    }

    // Job Opportunity Module
    private void addJobOpportunity() {
        String title = prompt("Enter Job Title:");
        String company = prompt("Enter Company Name:");
        String location = prompt("Enter Job Location:");
        String description = prompt("Enter Job Description:");
        jobOpportunities.add(new JobOpportunity(title, company, location, description));
        showMessage("Job Opportunity added successfully.");
    }

    private void displayJobOpportunities() {
        if (jobOpportunities.isEmpty()) {
            showMessage("No Job Opportunities found.");
        } else {
            StringBuilder sb = new StringBuilder("Job Opportunities:\n");
            for (JobOpportunity job : jobOpportunities) {
                sb.append(job).append("\n");
            }
            showMessage(sb.toString());
        }
    }

    // Search and Analytics Module
    private void searchAndAnalytics() {
        String profession = prompt("Enter Profession to Search:");
        boolean found = false;
        StringBuilder result = new StringBuilder("Search Results:\n");
        for (Alumni alumni : alumniList) {
            if (alumni.getProfession().equalsIgnoreCase(profession)) {
                result.append(alumni).append("\n");
                found = true;
            }
        }
```

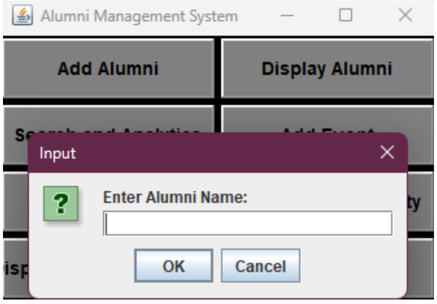
```
        if (!found) {
            result.append("No alumni found for the given profession.");
        }
        showMessage(result.toString());
    }

    // Helper methods
    private String prompt(String message) {
        return JOptionPane.showInputDialog(this, message);
    }

    private void showMessage(String message) {
        JOptionPane.showMessageDialog(this, message);
    }

    // Inner classes for entities
    private class Alumni {
        private String name;
        private int graduationYear;
        private String contactNumber;
        private String profession;

        public Alumni(String name, int graduationYear, String contactNumber, String profession)
{
            this.name = name;
            this.graduationYear = graduationYear;
            this.contactNumber = contactNumber;
            this.profession = profession;
        }

        public String getProfession() {
            return profession;
        }

        @Override
        public String toString() {
            return "Name: " + name + ", Year: " + graduationYear + ", Contact: " + contactNumber
+ ", Profession: " + profession;
        }
    }

    private class Event {
        private String eventName;
        private String date;
        private String description;

        public Event(String eventName, String date, String description) {
            this.eventName = eventName;
            this.date = date;
            this.description = description;
        }

        @Override
        public String toString() {
            return "Event: " + eventName + ", Date: " + date + ", Description: " + description;
        }
    }

    private class JobOpportunity {
        private String title;
        private String company;
        private String location;
        private String description;

        public JobOpportunity(String title, String company, String location, String description) {
            this.title = title;
            this.company = company;
            this.location = location;
            this.description = description;
        }

        @Override
        public String toString() {
            return "Job Title: " + title + ", Company: " + company + ", Location: " + location + ",
Description: " + description;
        }
    }

    public static void main(String[] args) {
        new AlumniManagementSystem();
    }
}
```

APPENDIX B

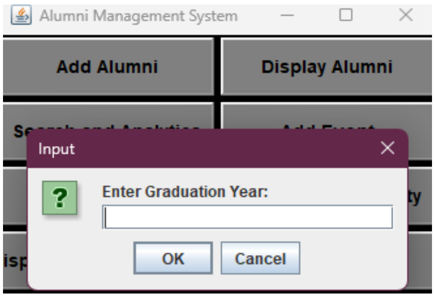
SCREENSHOT



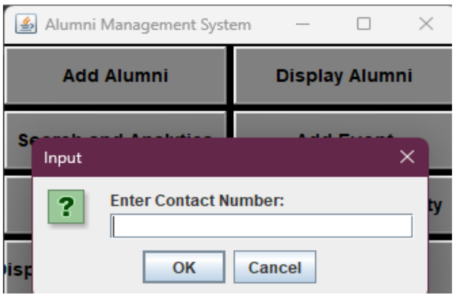
Add Alumni



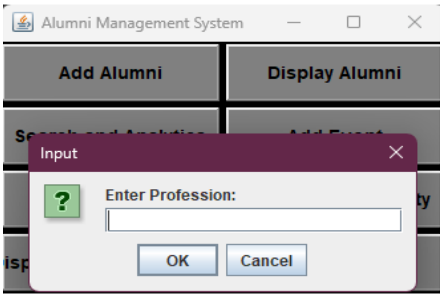
Enter Alumni Name:



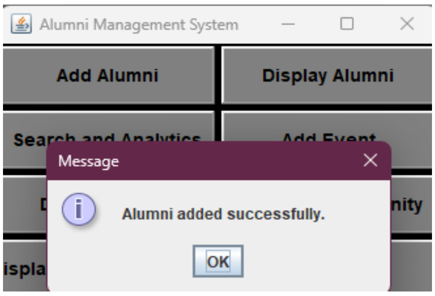
Enter Graduation Year:



Enter Contact Number:

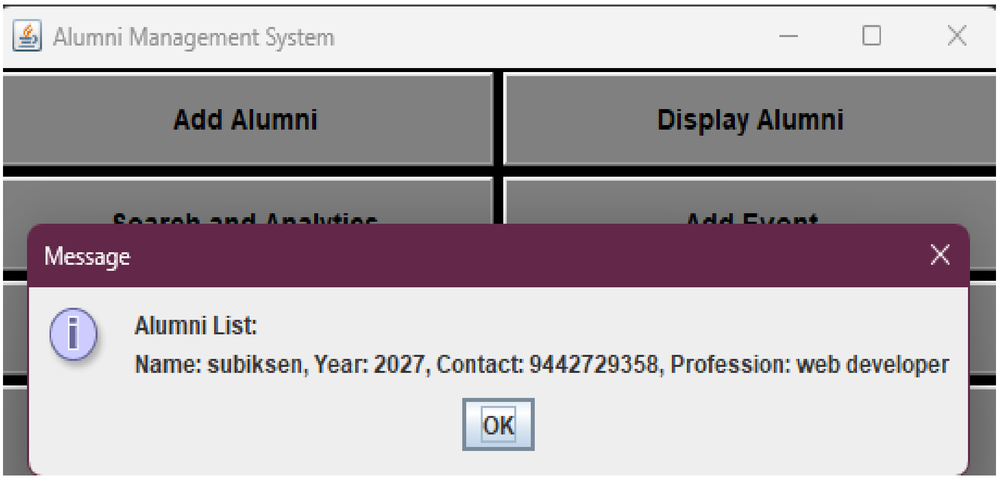


Enter Profession:



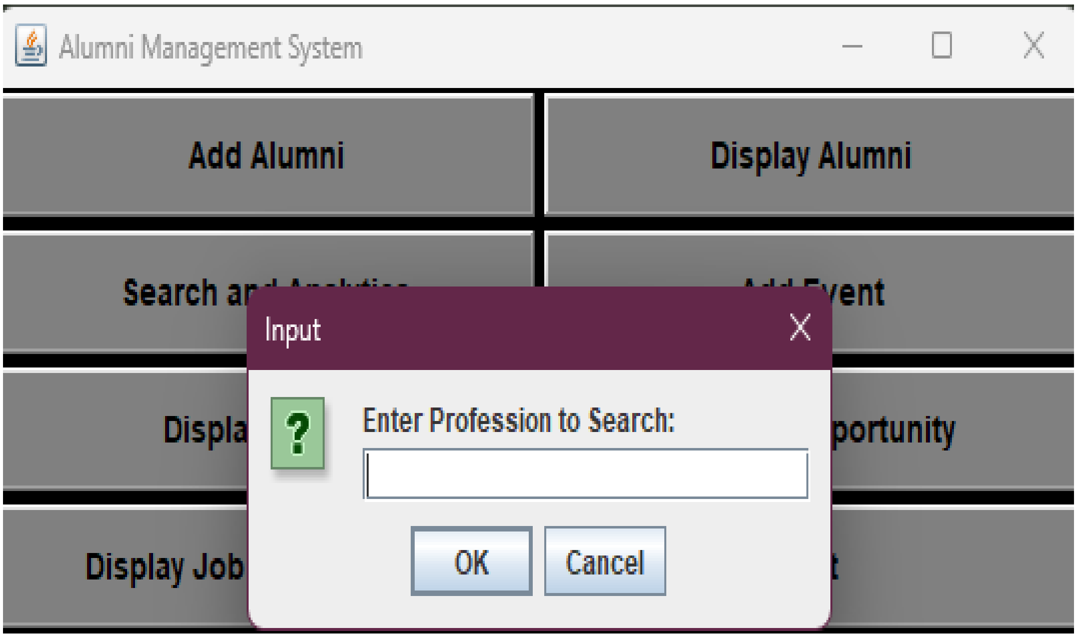
Alumni added successfully.

Display alumni

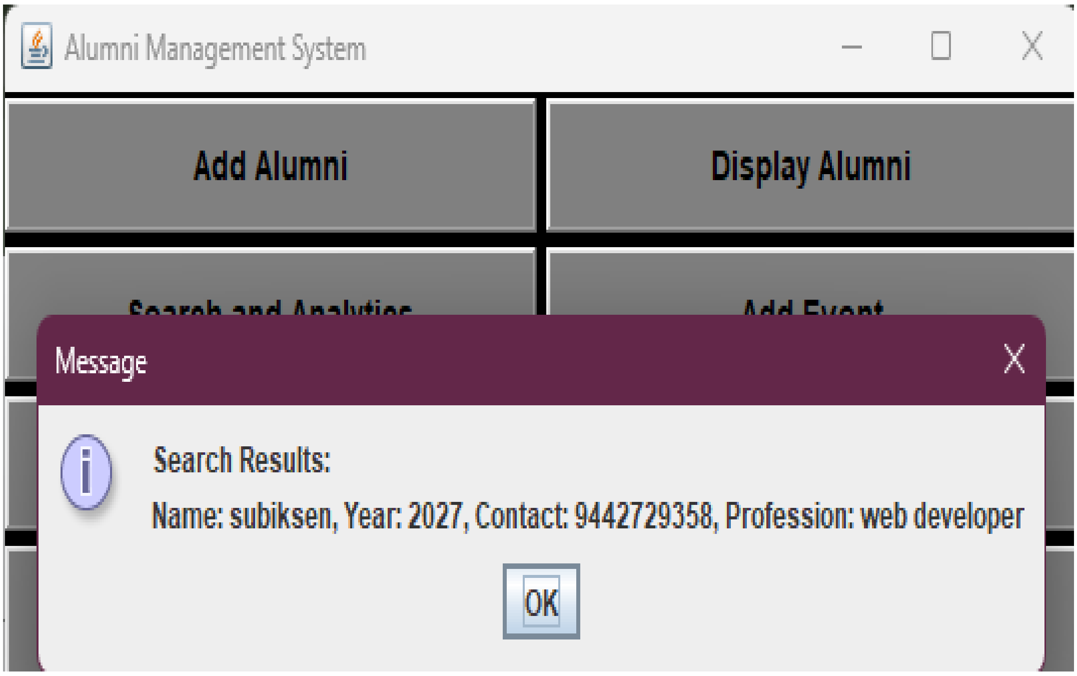


Alumni List:
Name: subiksen, Year: 2027, Contact: 9442729358, Profession: web developer

Search and Analytics



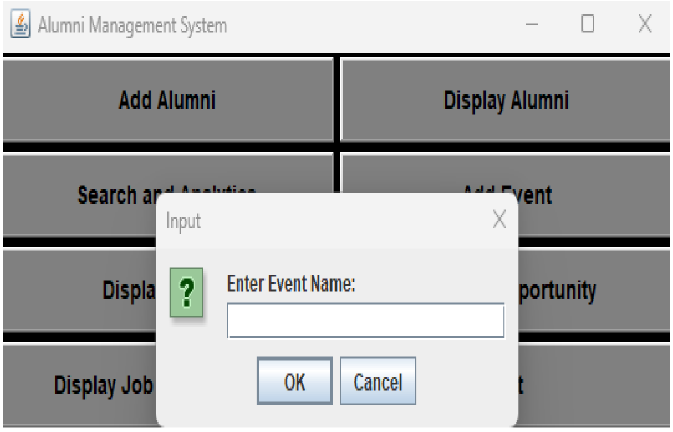
Enter Profession to Search:



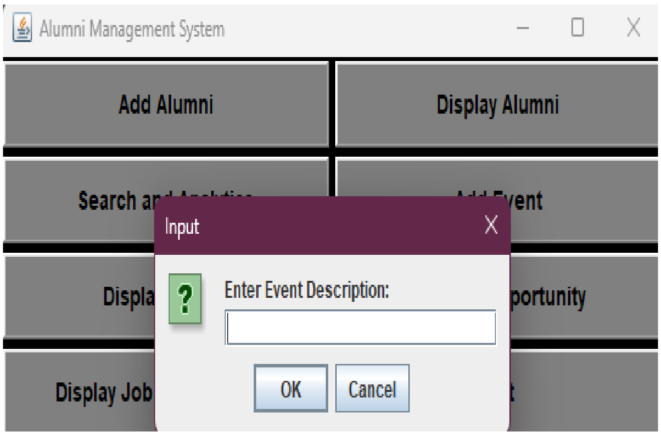
Search Results:

Name: subiksen, Year: 2027, Contact: 9442729358, Profession: web developer

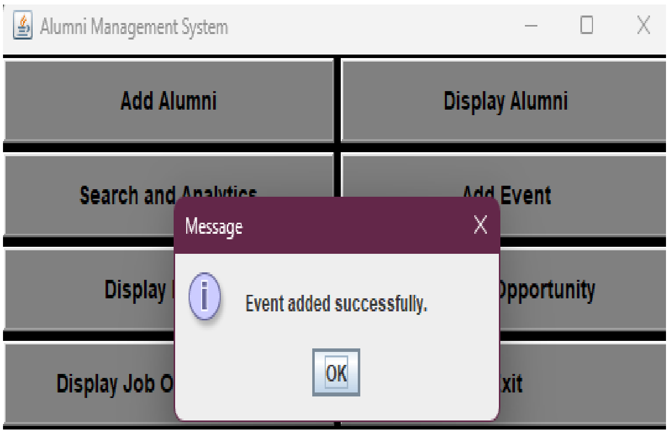
Add Event



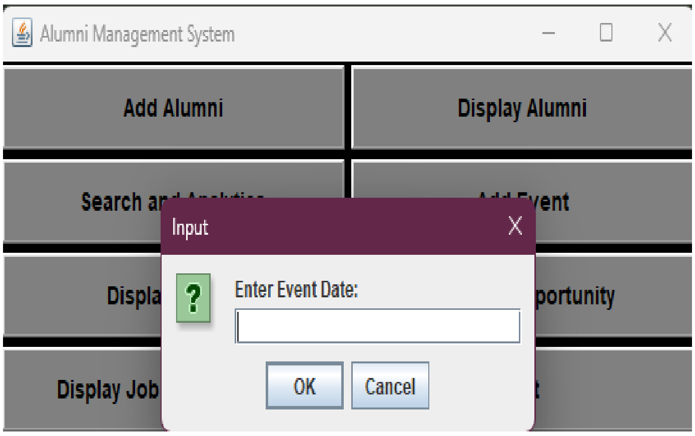
Enter Event Name:



Enter Event Description:

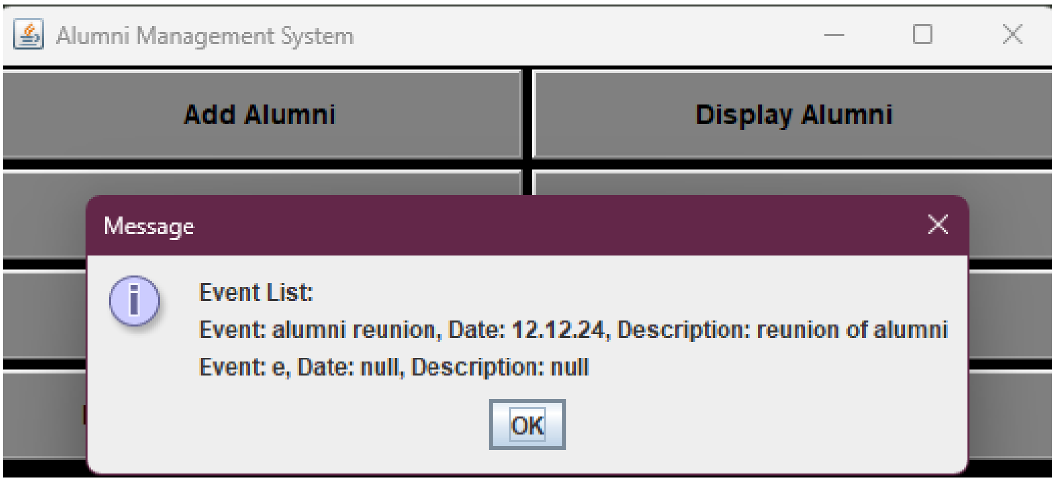


Event added successfully.



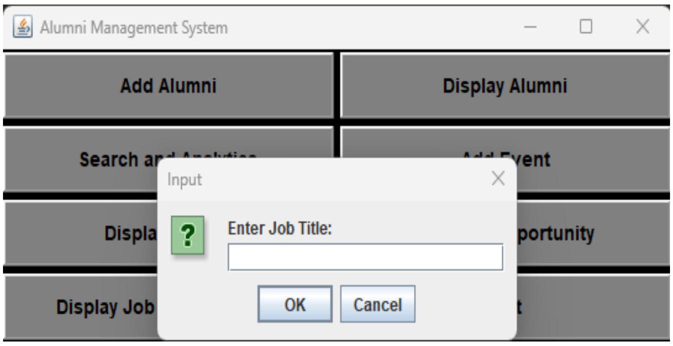
Enter Event Date:

Display Event

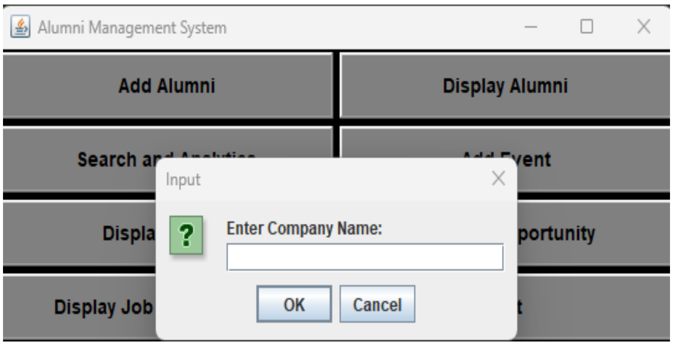


Event List:
Event: alumni reunion, Date: 12.12.24, Description: reunion of alumni
Event: e, Date: null, Description: null

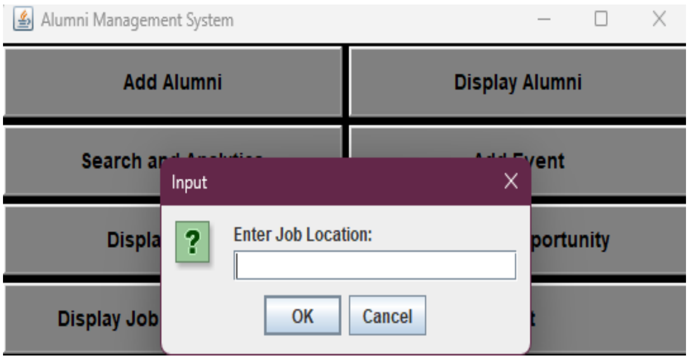
Add JobOpportunities



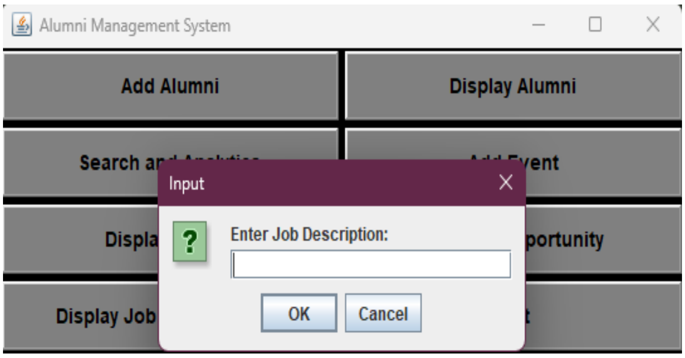
Enter Job Title:



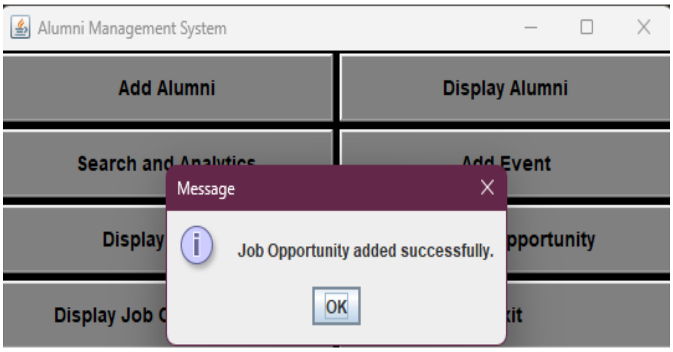
Enter Company Name:



Enter Job Location:

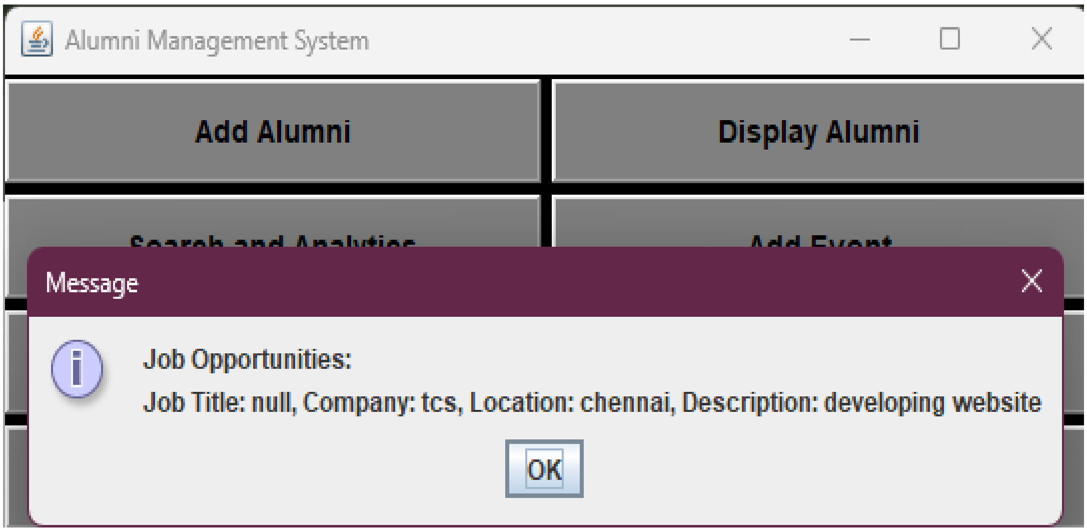


Enter Job Description:



Job Opportunity added successfully.

Display JobOpportunities



Job Opportunities:

Job Title: null, Company: tcs, Location: chennai, Description: developing website

REFERENCES

1. Kathy Sierra and Bert Bates (1996) **"Head First Java"** vol.1,pp.31-67

2. Herbert Schildt (2004) "**Java: The Complete Reference**" vol.3,pp. 57-89

3. Peter Coad, Jill Nicola (2008) **“Object-Oriented Design & Programming “** vol.5,No 1,pp.23-67