**EVENTURE- An Effortless Event Management System**

**by**

|  |  |
| --- | --- |
| **Name** | **Id** |
| **Bishwa Das** | **222-15-6341** |
| **Tonay Kundo** | **222-15-6386** |
| **Sourav Bhuiyan** | **222-15-6307** |
| **Sadia Parveen** | **222-15-6366** |

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

**Md. Ashraful Islam Talukder**

Lecturer

Department of CSE

Daffodil International University

****

**DAFFODIL INTERNATIONAL UNIVERSITY**

**Dhaka, Bangladesh**

**NOVEMBER 2024**

**TABLE OF CONTENT FOR DEVELOPMENT PROJECT**

**CONTENTS PAGE**

|  |
| --- |
| **ABSTRACT 4**  **CHAPTER 1-30**    **CHAPTER 1: INTRODUCTION 5-8**     * 1. Introduction 5   2. Motivation 5   3. Objectives 6   4. Expected Outcomes 6   5. Project Management and Finance 7   1.6 Report Layout 8  **CHAPTER 2: BACKGROUND 9-11**  2.1 Preliminaries/Terminologies 9  2.2 Related Works 9  2.3 Comparative Analysis 10  2.4 Scope of the Problem 10  2.5 Challenges11  **CHAPTER 3: REQUIREMENT SPECIFICATION 12-15**  3.1 Business Process Modeling 12  3.2 Requirement Collection and Analysis 12-13  3.3 Use Case Modeling and Description 13    3.4 Logical Data Model 14  3.5 Design Requirement 15  **CHAPTER 4: DESIGN SPECIFICATION 16-23**  4.1 Front-end Design 16-18  4.2 Back-end Design 18-19  4.3 Interaction Design and User Experience (UX) 20-21  4.4 Implementation Requirements 21-22  **CHAPTER 5: IMPLEMENTATION AND TESTING 24-28**  5.1 Implementation of Database 24-25  5.2 Implementation of Front-end Design 24-26  5.3 Testing Implementation 26-27  5.4 Test Results and Reports 27-28  **CHAPTER 6: IMPLEMENTATION AND TESTING 29-30**  6.1 Impact on Society 29  6.2 Impact on Environment 29  6.3 Ethical Aspects 29-30  6.4 Sustainability Plan 30  **Chapter 7: CONCLUSION AND FUTURE SCOPE 31**  7.1 Discussion and Conclusion 31  7.2 Scope for Further Developments 31  **REFERENCES 32** |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **FIGURES** | **PAGE NO** |
| Figure 1: Use Case Diagram | 13 |
| Figure 2: ER-Diagram | 14 |
| Figure 3: LogIn Form GUI | 16 |
| Figure 4: SignUp Form GUI | 16 |
| Figure 5: Home page or Welcome page GUI | 17 |
| Figure 6: Event Management Page GUI | 17 |
| Figure 7: Prototype Combination | 19 |
| Figure 8: Database and Tables Creation | 22 |
| Figure 9: SignUp and Event table in DB | 22 |
| Figure 10: Test LogIn form GUI | 23 |
| Figure 11: Test SignUp form GUI | 24 |
| Figure 12: Event Management GUI | 24 |
| Figure 13: Add, Edit and Delete Event GUI | 25 |

1. **LIST OF TABLES**

|  |  |
| --- | --- |
| **TABLES** | **PAGE NO** |
| Table 1: Comparison between existing system and Eventure | 10 |

**ABSTRACT**

Event Management software is a tool to plan, organize and manage an event from a unified system. By replacing manual event management with a digital alternative, this system seeks to overcome the difficulties of inefficiency,disorderliness and errors. Secure user authentication features allowing users to log in, sign up and manage their account have been integrated into the project. It provides an intuitive GUI build with Java Swing for creating, viewing, deleting, and editing events. MySQL forms the backbone of its backend system, providing secure storage and efficient retrieval of user and event data.The idea for this project was driven by an increasing need for automations across a wider variety of domains: classrooms, offices, and communities. The other part is that the 2 continuously integrates each and every real-time update between the GUI and the database, making it easy for a user to attain seamless experience. The development process was aimed towards providing a dependable, low-cost and user-friendly solution via open-source tools and frameworks.Ideal results would end with a stable, scalable and fully-fledged event management application that can streamline operations, improve productivity, and reduce errors. In this report, we describe the design and implementation process of such a system, the outcome of testing it at the 2022 ICIE and recommend future improvements. At a fundamental level, this project showcases the ability of technology to disrupt existing workflows and address growing pressure on traditional event management tools.

**CHAPTER 1**

**Introduction**

* 1. Introduction:

ICR Management System is a desktop-based application developed to facilitate the efficient process of user authentication and event management. This application is coded in Java, incorporating the Swing framework for GUI and JDBC for database connectivity. The ICR System is ideal for the automation of user registration, login validation, and event CRUD operations. The System source code is developed to boast of the unique nature of handling all school, organization, or individual events with ease and efficiency. This application is primarily developed for institutions, public gatherings, and individual parties that desire a straightforward and systematic way of managing their events. The application interacts with a MySQL database, where user and event data are securely kept, and only retrieval is done dynamically.

1.2 Motivation  
In a lot of organizations, the event management is done manually which often leads to challenges such as:  
 • **Inefficiency**: Manual processes to input and recover data are slow and prone to

human error.  
 • **Misguided Data:** Details of the event may be poorly managed, resulting in

conflicts on scheduling or incomplete information.  
 • **Poor Accessibility:** With no centralized system in place to retrieve/update the

event data.  
We are motivated to tackle these issues by developing an automatic system that offers:  
i. A secure centralized platform for managing event data  
ii. Speed and reliability of the process via database automated.  
iii. Simplified and more accessible interface.  
This also gives you a real world experience on how to combine your knowledge of Java, databases and GUI design into use and build software.

* 1. Objectives

1. The goals of the ICR Management System are:1. User Authentication: Create a

secure authentication and registration module to verify and control Users. Identify and prevent unauthorized access via the right authentication methods.

1. Event Management: Implement the ability to add, modify, remove and see existing

events.For real time updates and up to date data fetch from db.

1. Database Integration: Leverage MySQL for efficient storing and retrieval of user &

event data.

1. User-Friendly Interface: Create intuitive GUIs using Java Swing for seamless user

experience. Must have easy accessibility and navigation to other features.

1. Robust Error Handling: Use exception handling when there are problems connecting to the database or users inserts some bad data.  
   The project achieves a functional, secure and efficient event management system by meeting these objectives.
   1. Expected Outcomes:
2. The project should result in: Desktop App that Works:  
    An app that implements user authentication and event management functionality, which is working!
3. User Authentication — secure writing Register System: Users can log in or

register new accounts with their credentials stored within the database.

1. Dynamic Event Management: A users should see the list of events, add new, modify existing or delete an event if user finds he does not want to work on that particular entry. Events are shown dynamically in a JTable (data updates reflected on db automatically).
2. Database Operations Without Errors: Make sure the application works with the database, throwing exceptions in case of problems.
3. Enhanced Usability: Offer a simple and user-friendly interface enabling users to work with the system easily.  
   These results will help safeguard that the system is user-needs driven, while also proving a level of technical competency.
   1. Project Management and Finance:  
      1.5.1 Timeline  
      All of these phases in the project development process are arranged in such a way that it allows for quick delivery

Phase Duration Activities  
Week 1: Requirement Analysis — Gather requirements, define scope and finalize system features.  
Week 2: System design Create database schema, define architecture, and approach on how UX will look like Login & Sign-Up.

Week 3-4: Add login & sign-up functionality, Database integration for Event Management

Week 5-6: Create modules to create, edit and delete events.  
Week 7: Testing Unit and integration testing, Debugging, optimize code  
Week 8: Schedule Final Deployement Deploy the final product and document it  
  
1.5.2 Financial Budget  
The cost of the project is minimal; free tools and technologies have been used for this. These are the estimated costs below:

Item Price Tag (USD)

* Free Software Tools (Intellij IDEA)
* HostMysqlFree Database (Local MySQL)
* Self-Developed no cost.
* Miscellaneous (Utilities) $50
* Specialist consultation cost $100  
  Total $150

1.6 Report Layout  
In order to provide a comprehensive view of the project, this report is divided into the following chapters.

Chapter 1: Introduction: Give an overview of The project, its goal, motivation and

expected outcomes.  
Chapter 2: System Design: Talk about the architecture of system, Database Design

and GUI Layout.

Chapter 3: Implementation: Covers the structure of the code, high level

functionality on what algorithms were used, and how you integrated

databases.  
 Chapter 4: Results & Testing: Shows the results from the project and results of

different test.

Chapter 5: Discusses the possible future scope and summarize achievements of the

project.   
Finally, We will conclude this article by sharing a few ideas around what could come next.

**CHAPTER 2**

Background

2.1 Preliminaries / Terminologies:  
Key terms and concepts related to the ICR Management System project are introduced in this section to facilitate an understanding of the system functions and technicalities.  
 1. ICR — International Conference Room Event Management System: A

device that is created to help with event management processes — from authenticated user login to event creation, updates and deletes.  
 2. GUI — Graphical User Interface A visual interface where users can interact with graphical components that implement the functionality of a system (for example table cells, buttons and forms).  
 3. JDBC (Java Database Connectivity): A Java API that connects and runs queries on databases, so you may easily access and manipulate displayed data.  
 4. CRUD Operations: CRUD database functions operations  
 5. Event Management: Managing the details of events you have scheduled, such as an event name, description, date and location.

2.2 Related Works:  
In this section, we review existing systems and technologies that are either directly related to the aims of our ICR Management System or indirectly inspired its architecture.  
 1. Online Event Management System: Well, among existing platforms–such as Eventbrite and Cvent–automate event planning and management on a large basis. But most of them are geared toward web-based environments, while this project is aimed at a desktop-based solution.  
 2. User Authentication Systems: Many authentication systems store passwords in a hashed form and the username files inside hidden folders on the system. To authenticate users, the system implements a straightforward yet effective authentication mechanism.  
 3. Desktop Applications (Small Scale): Most desktop applications for event management are either too simple or do not provide integration with a database to manage dynamic data. This project fills that gap with a database-driven desktop solution.

2.3 Comparative Analysis:

This section highlights the differences and advantages of the ICR Management System compared to similar systems.

| **FEATURE** | **EXISTING SYSTEM** | **EVENTURE** |
| --- | --- | --- |
| **Platform** | Primarily web-based | Desktop-based |
| **Database Integration** | Limited or requires  third-party tools | Fully integrated with MySQL database |
| **User Authentication** | Advanced with multi-factor options | Basic but secure (username & password) |
| **Cost** | Subscription | Free, only development cost |

Quite unique from the above, ICR Management System is a simple and inexpensive tool that very easily serves small organizations on a personal level.

2.4 Scope of the Problem  
Problem Statement: Scope The problem scope is inefficient way of managing events and user credentials manually. Key aspects include:  
 1. Manual Event Management:Timely processes in creating, updating and tracking events. High potential for errors with duplicate entries or missed updates.  
 2. Lack of a Centralized System: Event details often exist in siloed documentation or spreadsheet which can be cumbersome when it comes to identifying information that needs to be changed quickly.  
  
 3. Limited User Authentication: An example of this is that manual tracking of user credentials can lead to data breaches and privacy invasion.  
  
 4. The storage and availability of data: Without a databases-driven system this event data becomes structure-less and scale-inheritantly difficult to manage within the bounds of large amounts of information.

2.5 Challenges:  
We faced a number of challenges while designing and building the ICR Management System  
 1. Database Connectivity: Implementing a strong connection between the Java program and MySQL DB was not easy, its took some time to configure it properly and test the exception handling while the database is not getting connected.  
 2. GUI Design:

Task: It was challenging to create an easy and user-friendly interface using Java Swing, thus aligning components and making it responsive.  
 Data Validation: One critical aspect of the application was validating all user inputs, like event dates and descriptions to make sure they were formatted properly to preserve data integrity.  
 3. Error Handling: In case of database querying, if there is an incorrect query or the required data is not found, exceptions needed handling so that the application does not crash.  
 4. Scalability: While the system was meant for small to medium sized organizations, it also had to be scalable when we look at future growth (for instance multi-user capability) so this made things more complicated.

**CHAPTER 3**

**Requirement Specification**

This chapter describes the specification and design process followed while developing the Event Management System, which helps to understand the system functionality in greater detail along with design specifics of such technologies.

3.1Business Process Modeling:  
Business Process Modeling — This step consists of defining the workflows and processes that a solution such as Event Management System will facilitate automation for. Key processes include:  
 1. User Authentication: User Authentication via a user name and password. The system checks the credentials to gain secure access.  
  
 2. Event Creation: Users enter the event details such as name, date, location. The system processes and store the data in database.  
  
 3. Event Updates: Users update pre-existing event information. Newly created records are updated and synchronized with the database.  
  
 4. Event Deletion: Events are deleted by the users when they are not needed anymore. The system has data integrity; it deletes all related, dependent records.

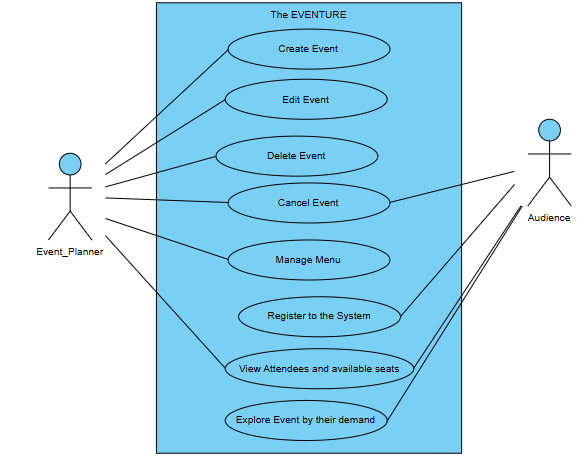
Activity diagrams and flowcharts visualize business processes to illustrate how the system works.

3.2 Requirement Gathering and Analysis  
Functional Requirements:  
User Authentication: Log in and log out feature with security credential.

Event Management: CRUD operations for event data  
Database Integration: An MySQL database to keep track of events and users.  
User-Friendly Interface: A GUI (graphical user interface) for interacting with the system.

Non-Functional Requirements:  
Performance: The system is designed to efficiently manage up to 50 concurrent users.  
Scalability: Built to enable multi-user functionalities down the line.  
Security: Credential encryption o User credential encryption  
Reliability: Uptime of the system should be more than 99%

3.3 Use Case Modeling:   
Use Case Diagram

3.4 Logical Data Model:  
The Logical Data Model reflects the logical structure of the database, allowing for efficient storage and retrieval. This diagram Shows entities and Relationship both.

ER-Diagram: The Eventure

User

Registered By

Event

Function

Log\_in

Perform

View

m

n

1

1

1

T

k

x

Update Event

Update Event

1

Event\_Record

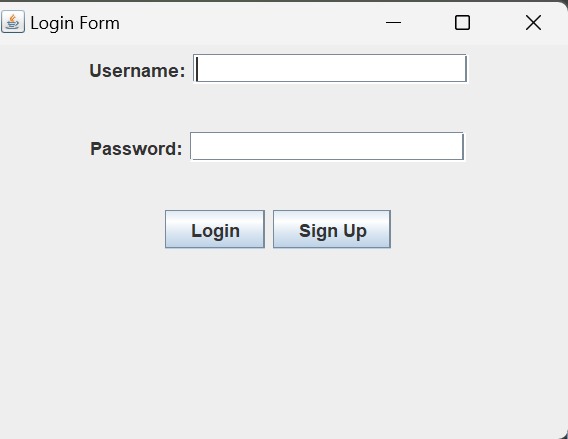
3.5 Design Requirement:  
The design requirement is to make the system functional and user-friendly.

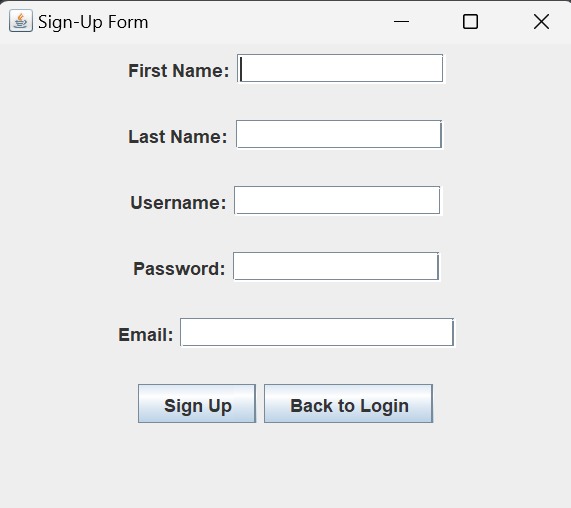
User Interface Design: An easy to use GUI developed with Java Swing components.   
Database Design: A normalized MySQL database to avoid redundancy, Storing procedures for complex queries  
System Architecture: A three-layer architecture:  
 Presentation Layer: Part of where users directly interact with the system.  
 Application Layer → Application that is concerned with business logic  
 Database Layer: Deals with data storage and retrieval  
Security Design: After more update we use Hashing algorithms to encrypt passwords.

**CHAPTER 4**

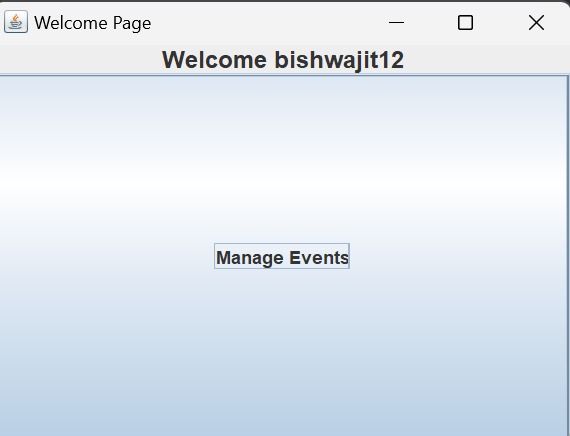
**Design Specification**

This chapter focuses on the Event Management System design specifications, including application standards for both the front-end and back-end, interaction requirements among system components, as well as implementation requirements to ensure a seamless user experience.  
  
4.1 Front-end Design  
Front-end design is about developing an intuitive and user-friendly interface that allows users to interact with the system.  
Key Features:  
LogIn/SignUp Screen:

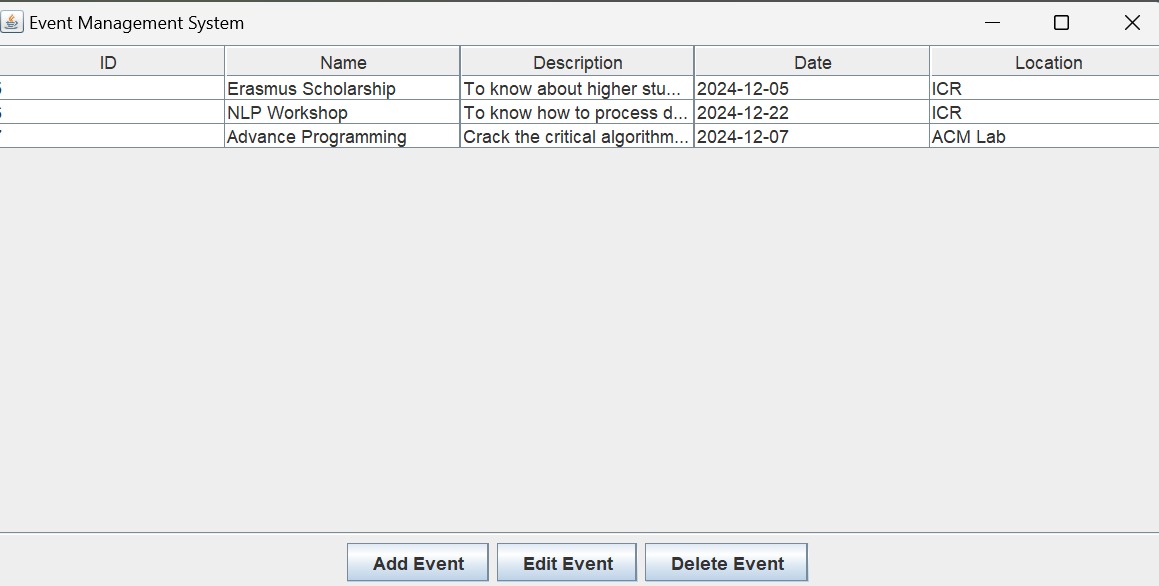
  
This interface for login to the system.



If new user then need to sign up firsty and login to the system.  
Dashboard:

  
Home page and welcoming the user with his username.

Event Management Screens:  
 o Event Add, Edit and Delete Forms.



4.2 Back-end Design:  
Back-end design allows the system to process information and manage data effectively in a secure, reliable manner.  
Key Components :  
Database Connection : MySQL DB for keeping user credentials and event data.

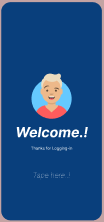
private final String DB\_URL ="jdbc:mysql://localhost:3306/icr\_management";  
private final String DB\_USER = "root"; // Replace with your DB username  
private final String DB\_PASS = ""; // Replace with your DB password  
  
Connection connection;  
  
public static void main(String[] args) {  
 new ICRManagementSystem().initializeDatabase();  
}  
  
// Initialize database connection  
public void initializeDatabase() {  
 try {  
 connection = DriverManager.*getConnection*(DB\_URL, DB\_USER, DB\_PASS);  
 System.*out*.println("Connected to the database.");  
 showLoginForm();  
 } catch (SQLException e) {  
 JOptionPane.*showMessageDialog*(null, "Database connection failed: " + e.getMessage());  
 System.*exit*(1);  
 }  
}

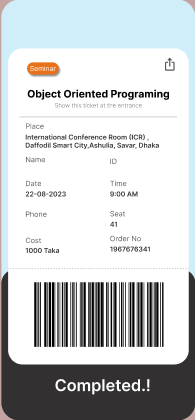
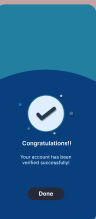
Application Logic: Java based application to manage all business processes for user login, validation and CRUD functionalities.

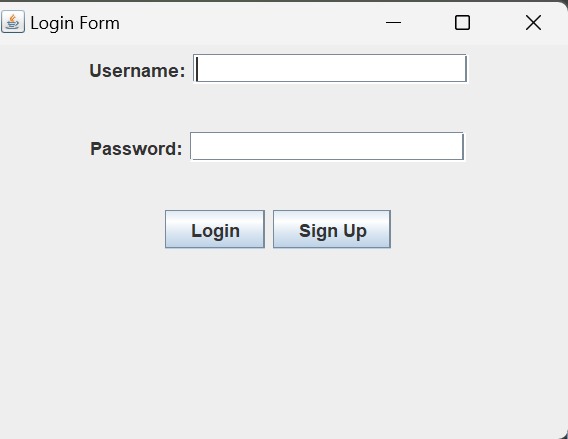
public void showEventManagementSystem() {  
 frame = new JFrame("Event Management System");  
 frame.setSize(800, 400);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel(new BorderLayout());  
  
 String[] columns = {"ID", "Name", "Description", "Date", "Location"};  
 tableModel = new DefaultTableModel(columns, 0);  
 eventTable = new JTable(tableModel);  
 loadEvents();  
  
 JScrollPane scrollPane = new JScrollPane(eventTable);  
 panel.add(scrollPane, BorderLayout.*CENTER*);  
  
 JButton addEventButton = new JButton("Add Event");  
 JButton editEventButton = new JButton("Edit Event");  
 JButton deleteEventButton = new JButton("Delete Event");  
  
 addEventButton.addActionListener(e -> showEventCreateForm());  
 editEventButton.addActionListener(e -> showEventEditForm());  
 deleteEventButton.addActionListener(e -> deleteEvent());  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.add(addEventButton);  
 buttonPanel.add(editEventButton);  
 buttonPanel.add(deleteEventButton);  
 panel.add(buttonPanel, BorderLayout.*SOUTH*);  
  
 frame.add(panel);  
 frame.setVisible(true);  
}

This is the main or most perform class of my project where events are manage.

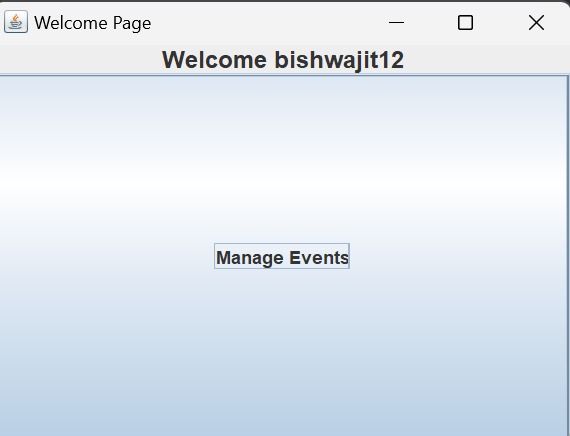
4.3 Interaction design  
Our Prototype:

       
Accessibility:  
 o Multiple easily readable fonts and contrast colors that support usability  
User Interaction Workflow:  
 • Login Process:

Enter username and password.  
In the case of bad credentials receive instant feedback.

• Event Management Process:

  
Go to the event management page

4.4 Need for implementation:   
A technical and hardware required to deploying the Event Management System is detailed in this section.  
Hardware Requirements:

CPU: Factory unlocked Intel Core i3 or above

RAM: Minimum 4GB.

Storage: 500MB free disk space for db and app files

OS: Win 10 or later / Linux.  
  
Software Requirements:

Language: Java (JDK 8 or higher)

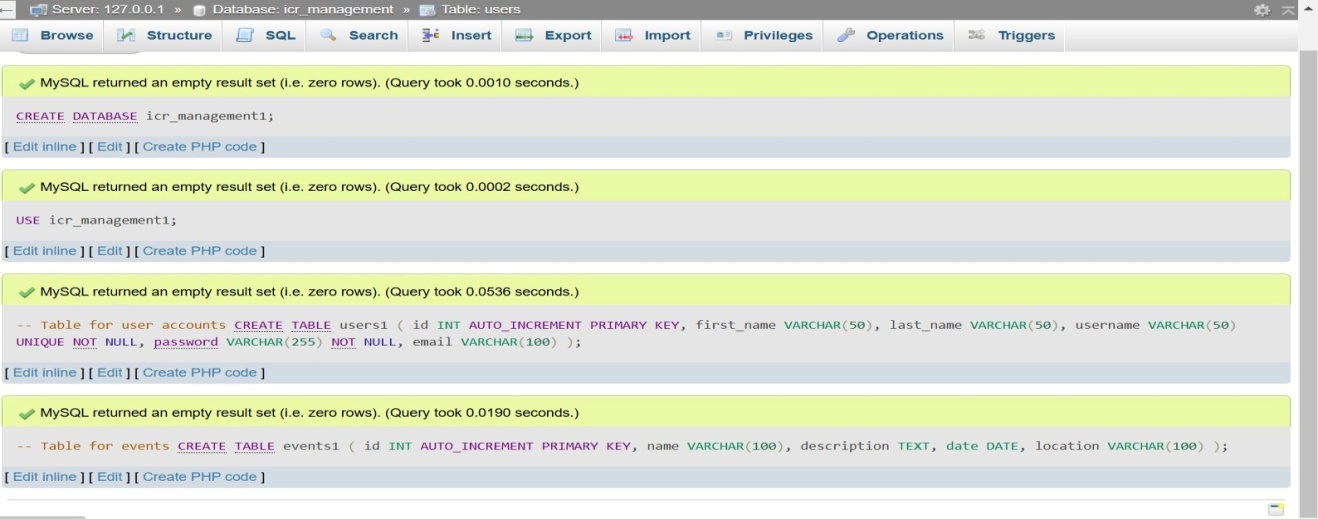
Database: MySQL 8.0.

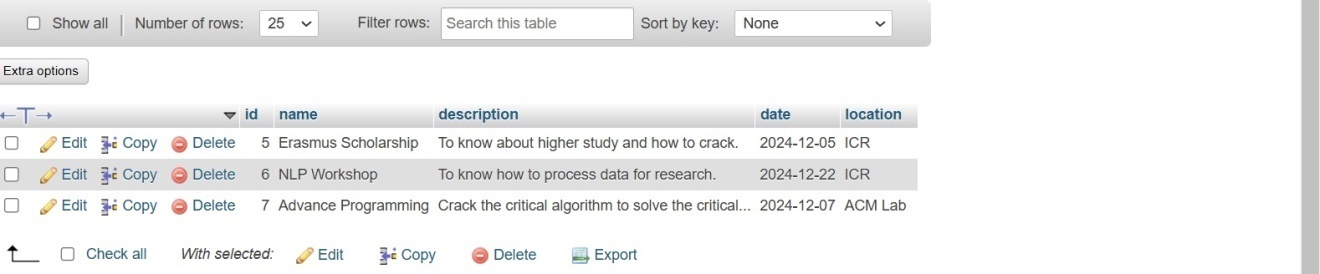
IDE: IntelliJ IDEA.  
Development Tools:  
 Java Swing for front-end.  
 JDBC for connecting to your back-end database  
 Database Management using MySQL XAMPP  
Testing Requirements:  
 All the CRUD operation functional testing.  
 Usability testing makes sure it is user-friendly.  
 Security testing that verifies encryption and access control.

**CHAPTER 5**

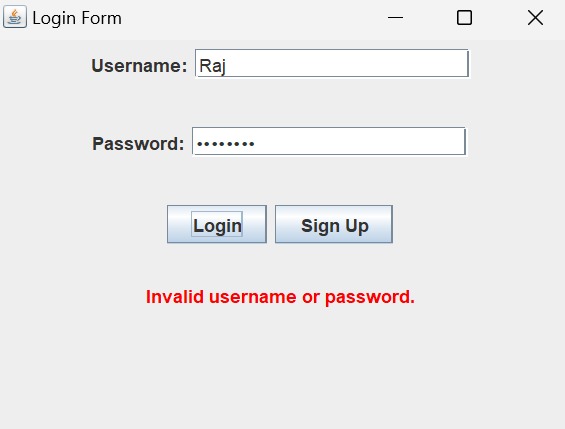
**Implementation and Testing**

In this chapter, it explains the process of implementing the Event Management System and it also describes the testing perform as well as its result in order functionality and reliability to ensure that the system functioning accurately.  
  
5.1 Database Implementation  
Database Setup:  
• Technology Used: MySQL 8.0.

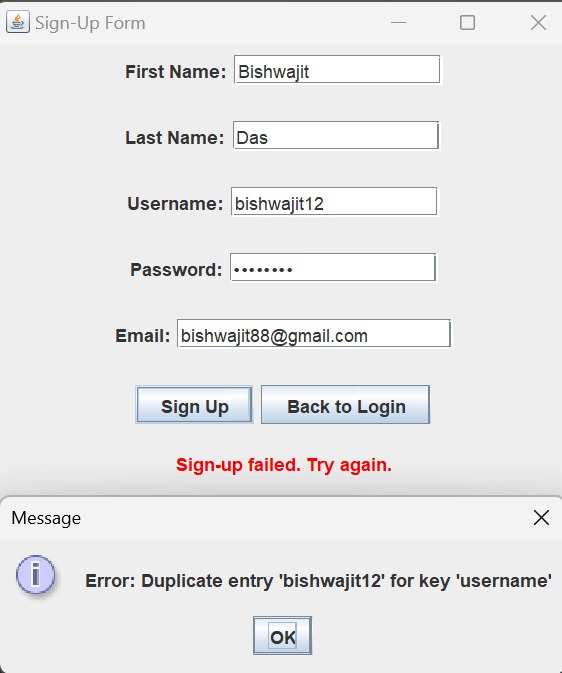
  
Stored Procedures:  
• AddEvent: A function to insert event records into a database.  
• DeleteEvent: A method to delete an event and associated information.  
• UpdateEvent: A function for updating event data.  
Database Connectivity:  
• JDBC (Java Database Connectivity) JAR file was added to what application, which connected the database  
  
 After SignUp the user all information save to the database.

Stored the event info that have created and no occur.

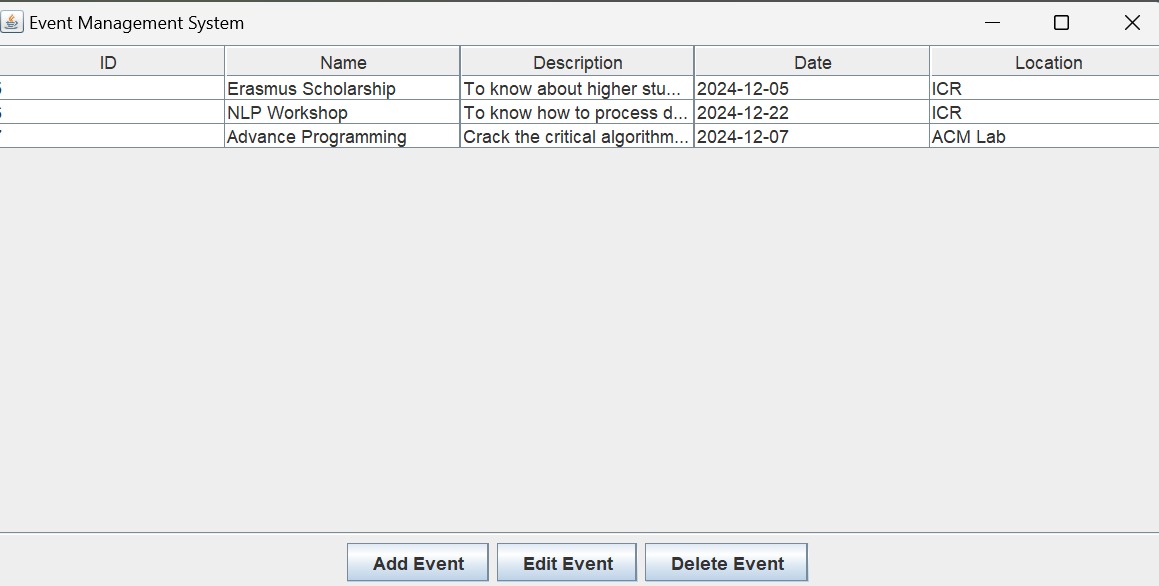
5.2 Implementation of the front-end design

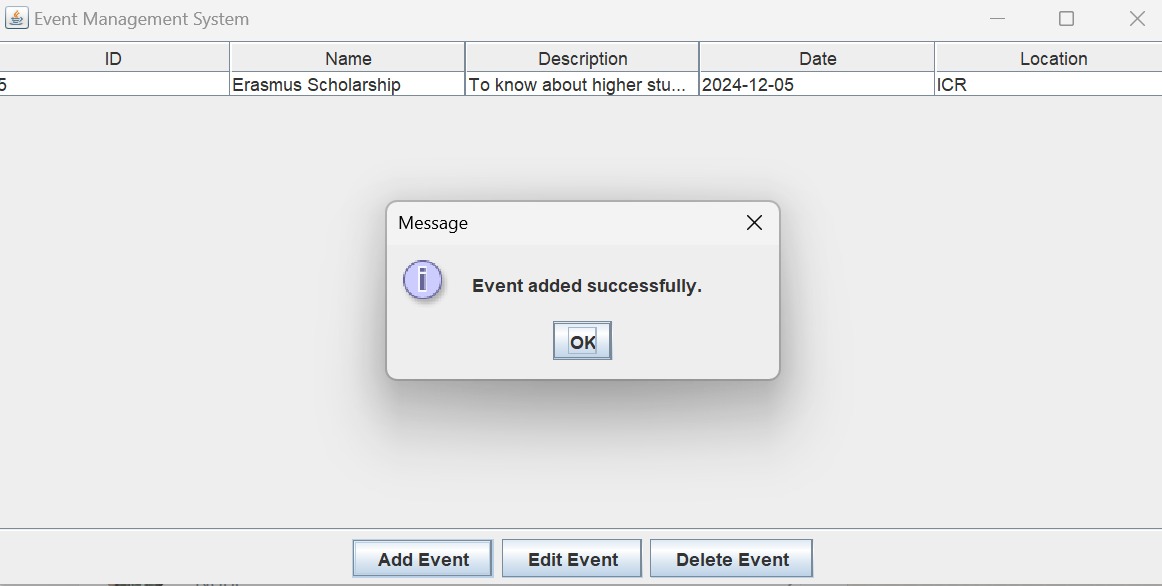
Development Environment:  
• IDE: IntelliJ IDEA.  
• Frameworks/Tools: The GUI is java Swing.  
Testing:   
  


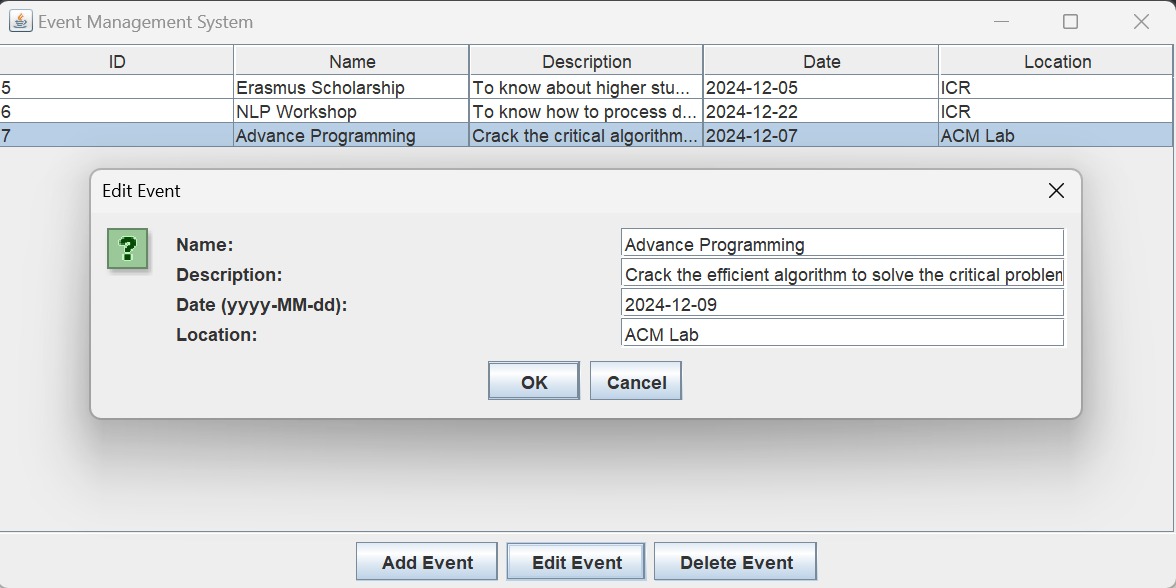
Wrong credential or empty format are not allow for login(validation).

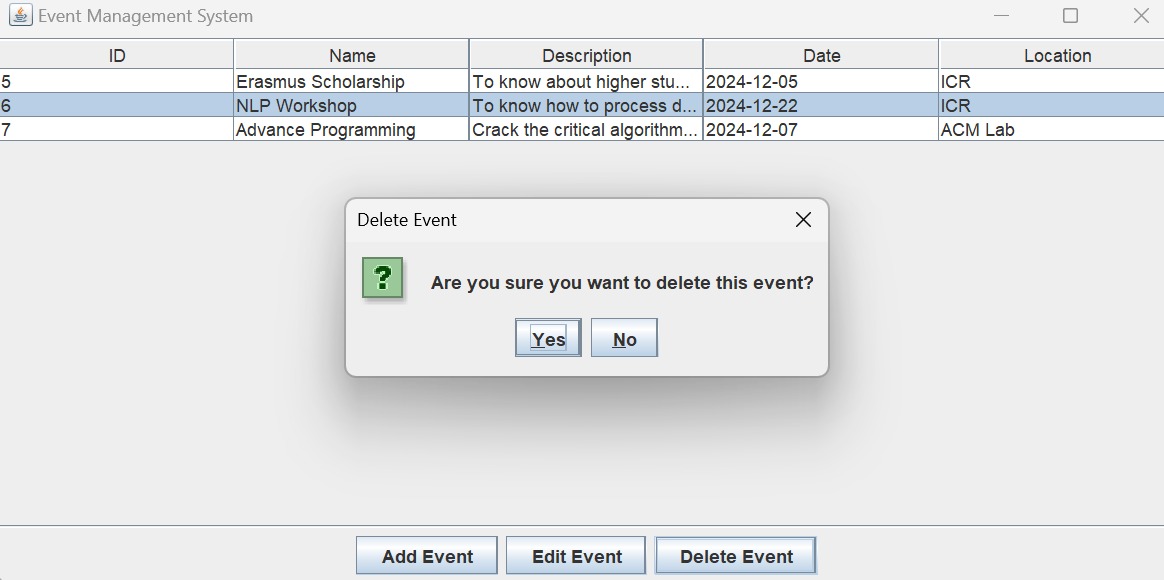


Can’t register by wrong info like duplicate username.  
  
Event Management Screen: Event addition/updating/deletion consisting of dynamic forms.



5.3 Testing Implementation:   
Create forms for inserting or updating event information.

Click action and see the updated event table.



Delete unnecessary event.

* 1. Test Results and Reports

Summary of Test Cases:

Login Module:

* Total Test Cases: 10
* Passed: 9
* Failed: 1 (edge case that fails when a user account is expired)

Database Operations:

* Total Test Cases: 20
* Passed: 20

Failed: 0.

Notable Issues Identified:  
• Database Connectivity Timeout: Appeared during high load tests Fixed through connection pooling optimization  
• Event Update Bug: Some updates were unable to succeed due to those successions not forwarding dates properly. Solved: Uniform date format enforced  
  
Final Report:  
• For the small to medium scale Event management, system is working fine and reliable.  
• No showstopper bugs remained opened and system satisfies criteria discussed in design phase

**CHAPTER 6**

**Impact on Society, Environment and Sustainability**

In this chapter, we assess the positive and negative potential impacts that Event Management System can have on Society, Environment, Ethics and Sustainability.  
  
6.1 Impact on Society  
1. Streamlined event management efficiency: This system automates the process of event planning and management, saving a lot of time & reducing manual effort, which is beneficial for individuals and small organizations.  
2. Increased Accessibility: Our solution is designed for all types of small and medium-scale organizations to manage events effortlessly without having to depend on expensive software, thereby driving inclusivity.  
3. Easy to arrange and manage an event: By using this system user can easily create and manage .

6.2 Impact on Environment  
  
• Reduced Paper Usage: The system eliminates the traditional paper-based ways and removes paper to a large extent through digitalisation.  
  
• Energy Efficiency: It was designed for use on desktops, where computation required is minimal and so has less environmental impact than large web-based systems.  
  
• Indirect Benefits: The system indirectly assists in environmental conservation efforts by limiting/ eliminating travel and physical sharing of resources through the promotion of remote and digitised event management services.

6.3 Ethical Aspects  
• Data Privacy and Security: Credentials & events are stored in a secure manner as well as using encryption based mechanisms to avoid unauthorized access.

• Transparency: System actions : When users perform an action in the system like a data update or delete, camel logs it against.something to know who did what.  
  
• Fair Accessibility: Software will have to be open source or affordable, and equity will be a main principle for accessibility by small orgs/non-profits

6.4 Sustainability Plan  
• Maintenance and Updates: System updates are periodically released to add functionality and support for new technologies.  
  
• Scalability: The system is built to scale with and be expanded with features such as multi-user support, and/or cloud capabilities.  
  
• Community Support: Promote open-source contribution for continuous sustainability and collaboration.

**CHAPTER 7**

**Conclusion and Future Work**

7.1 Discussion and Conclusion  
  
The Event Management System manages to fulfill its purpose, which is to automate and simplify the tasks in managing events. Key outcomes include:  
• CRUD operations for events data were simplified.  
• Safe user management Authentication system  
• A clean and simple interface well suited for events of a smaller to medium scale.  
  
When it comes to society, the system is a positive addition as it embraces its ability to digitize and thus improve accessibility while helping eliminate manual workloads addressing environmental issues.  
While the system has proven to be effective, it only works on one user at a time, presenting opportunities for future development.  
  
7.2 Potential for Future Work  
Potential enhancements to the system in the future include:  
  
1. Multi-User Functionality: Role-based access control with support for multiple users  
  
2. Cloud Integration: Migrate the system so that it could be accessed remotely, i.e. hosted on the cloud for greater scalability.  
  
3. Advanced Analytics: Attach data visualization tools to gain insights on the trends and dynamics of attendance.  
  
4. Mobile Application: Creating a mobile application to manage events while on the go.

**References**

[1] Smith, J. (2020). *Event Management Software: A Complete Guide*. Tech Publishers.  
[2] Brown, P. (2019). *Database Management for Beginners*. Code World Publications.  
[3] Tech Academy (2021). "Introduction to Java Swing," [Online]. Available: https://www.techacademy.com/javaswing