**SMARTPASS AUTHORIZATION SYSTEM**

MINI PROJECT REPORT

Submitted by

**C.AJAY [RA2311026010128]**

# M.SUHAS [RA2311026010131]

# N. MOHITH [RA2311026010132]

# D.TRILOK [RA2311026010137]

Under the Guidance of

**Mrs. Shaik Rasheeda Begum**

**DEPARTMENT OF COMPUTATIONAL INTELLIGENCE**



**FACULTY OF ENGINEERING AND TECHNOLOGY SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR**

**NOVEMBER 2024**

**SRM INSTITUTION OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**BONAFIDE CERTIFICATE**

Certified that the 21CSC203P Advance Programming Practice course project report titled **“**Project Title**”** is the Bonafide work done by C. AJAY [RA2311026010128] ,M. SUHAS[RA2311026010131] ,N. MOHITH[RA2311026010132] ,D. TRILOK[RA2311026010137] of II Year/III Sem B. Tech (CSE-AIML)who carried out the mini project under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE**  **Faculty In-Charge**  **Shaik Rasheeda Begum**  Assistant Professor,  Department of Computational Intelligence,  SRM Institute of Science and Technology Kattankulathur | **SIGNATURE**  **Dr.Annie Uthra R**  **Head of the Department**  Professor and Head  Department of Computational intelligence,  SRM Institute of Science and Technology Kattankulathur |

# ABSTRACT

The “**SMARTPASS AUTHORIZATION SYSTEM”** is a Java-based application developed to streamline the process of student out pass requests in educational institutions. The traditional method of manually handling out pass requests is time-consuming, inefficient, and prone to errors. This project aims to automate the entire process, enabling students to request out passes online, while administrators and wardens can review, approve, or reject requests digitally.

The system is designed using Java for the backend logic, ensuring a robust and scalable application. It utilizes MySQL for database management, storing essential details such as student information, request status, and historical records. The application features a user-friendly interface that allows students to easily submit requests and track their approval status in real-time. Administrators have access to a dashboard where they can view pending requests and manage them effectively.

By integrating a secure login mechanism and efficient data handling through MySQL, the Online Out pass System offers a reliable, paperless solution, reducing administrative workload and improving the overall student experience. This project demonstrates the practical application of Java programming and database management to solve real-world problems in a college environment.

# ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor**

**Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavours.

We would like to express my warmth of gratitude to our **Registrar Dr. S Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology)**

**Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project.

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our Course project Faculty **Dr. Rasheeda Begum ,Assistant Professor [Jr.G], Department of Computational Intelligence,** for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HOD Dr.Annie Uthra R Professor and Head, Department of Computational Intelligence** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| Sr. No. | Title | Page No. |
| 1. | Introduction | 6 |
| 2. | Literature Survey | 8 -9 |
| 3. | Requirement Analysis | 10-12 |
| 4. | Architecture and Design | 13-16 |
| 5. | Implementation | 16-23 |
| 6. | Experiment Result and Analysis | 23-24 |
| 7. | Future Scope | 25 |
| 8. | Conclusion | 26 |
| 9. | References | 27 |

**INTRODUCTION**

The **SMARTPASS AUTHORIZATION SYSTEM** is a digital solution designed to simplify and streamline the process of student out pass requests in educational institutions. Traditionally, students are required to manually fill out forms and obtain multiple approvals to leave the campus temporarily, which can be a cumbersome and time-consuming process. This system aims to automate and digitize the entire workflow, making it more efficient, transparent, and user-friendly.

Developed using **Java** for the application logic and **MySQL** for database management, the SMARTPASS AUTHORIZATION SYSTEM offers a robust platform where students can easily submit their out pass requests online. The system includes a secure login feature, ensuring that only authorized students and staff can access the portal. Once a request is submitted, it is automatically forwarded to the designated authorities (such as wardens or faculty) for review. Administrators can view all pending requests, approve or reject them with just a few clicks, and provide real-time updates to the students.

Key features of the SMARTPASS AUTHORIZATION SYSTEM include a user-friendly interface, efficient request management, secure data handling, and real-time notifications. The integration of Java's object-oriented programming capabilities ensures that the application is scalable and can handle high volumes of user interactions. MySQL serves as the backend database, efficiently storing student information, request history, and approval records.

This system brings significant benefits to educational institutions by reducing manual paperwork, minimizing errors, and enhancing the speed of processing out pass requests. It fosters better communication between students and administrative staff, while also improving the overall safety and security of campus operations. By adopting the SMARTPASS AUTHORIZATION SYSTEM, institutions can modernize their administrative processes, creating a more organized and seamless experience for everyone involved.

# LITERATURE SURVEY

# 1. This paper discusses the shift from traditional paper-based outpass systems to digital platforms within educational institutions. Kumar and Singh highlight the efficiency and time-saving benefits of online systems, reducing administrative workload and minimizing delays in student exit permissions. Their findings underscore the importance of digital solutions in educational settings, emphasizing how online outpass systems can improve operational efficiency and transparency in tracking student movements.

# 2. This study explores the security measures needed for effective online outpass systems. Gupta and Verma discuss data encryption and user authentication methods, addressing privacy concerns associated with tracking students’ exits. Their work is particularly relevant to this research as it highlights the importance of data security protocols and the need for compliance with privacy regulations when implementing digital outpass systems.

# 3. Sharma and Patel present an in-depth review of the advantages and limitations of SMARTPASS AUTHORIZATION SYSTEM specifically within hostel settings. They discuss issues such as system downtime, user resistance, and the need for continuous maintenance. The findings contribute valuable insights into the operational challenges faced by institutions, emphasizing the importance of a robust and user-friendly design for maximizing adoption and effectiveness.

# 4. This paper examines how data analytics integrated within SMARTPASS AUTHORIZATION SYSTEM can provide institutions with actionable insights. Yadav and Joshi argue that analyzing exit trends and usage patterns can help in resource planning and improving campus security measures. The study is relevant as it demonstrates how analytics can enhance decision-making in educational institutions, providing a basis for integrating data-driven approaches into outpass systems.

# 5. Das and Rao’s work compares traditional and digital outpass systems, highlighting the reduction in errors and time delays with online solutions. Their research shows that digital systems are more reliable and provide a transparent record of permissions, which is crucial for accountability. This study supports the current research by illustrating the practical benefits of transitioning to a digital system for managing student movements.

# REQUIREMENT ANALYSIS

Requirement Analysis is the process of identifying and documenting the needs and expectations of stakeholders for a particular project or system. It serves as a foundation for the entire system development process and ensures that all functional and non-functional needs are clearly defined before moving into the design and implementation phases.

## 

## Functional Requirements:

## Student Registration and Login:

## Students must be able to register with a unique ID and secure password.

## Secure login system for authorized access to the platform.

## 2.Out pass Request Submission:

## Students should be able to submit out pass requests, including details like reason, date, and time.

## The system should allow students to fill out an easy-to-use form for submitting requests.

## 3.Admin Approval/Denial:

## Admin users (e.g., wardens, faculty) should have the ability to approve or deny student out pass requests.

## Admins should be notified about pending requests and be able to take immediate action.

## 4.Request Status Tracking:

## Students should have the ability to track the status of their outpass requests (approved/denied/pending).

## Real-time notifications must be sent when the status changes.

## 

## Non-Functional Requirements:

## 1. Performance

Response Time: Defines the time it takes for the system to respond to a user request. In an SMARTPASS AUTHORIZATION SYSTEM, the time to submit a request or check status should be minimal, ideally in real-time or a few seconds.

## Throughput: Refers to the number of requests or operations the system can handle within a certain time. The system must handle multiple outpass requests from students simultaneously without delays.

## Latency: The time delay between an input and the corresponding system response. For example, when students submit outpass requests, the system should immediately acknowledge the submission.

## 2. Scalability

## The ability of the system to handle an increasing number of users or data without compromising performance. As the educational institution grows, the Online Outpass System must be able to support more students, staff, and data, scaling seamlessly to handle the increasing demand.

## Vertical Scalability: Adding more power to existing hardware (e.g., increasing CPU, RAM).

## Horizontal Scalability: Adding more servers to distribute the load and prevent bottlenecks.

## 3. Security

## Authentication and Authorization: Ensures that only authorized users (students and administrators) can access the system and perform actions within their designated roles. Password-based login and role-based access control (RBAC) are commonly used for authentication.

## Data Encryption: Sensitive data, such as student details and request history, should be encrypted both during storage and while transmitted across the network to protect against unauthorized access.

## Data Integrity: Ensures that data remains accurate and consistent, even if the system crashes or if there are network interruptions. Regular backups and transactional logging can be used to achieve this.

## Access Control: Protects the system against unauthorized users and ensures that each user has appropriate access to resources (students can submit requests, admins can approve them).

## 4. Usability

## Refers to how easy and intuitive the system is for users. A good SMARTPASS AUTHORIZATION SYSTEM should have a clean, well-organized GUI that is easy to navigate, minimizing the training required for both students and administrative staff.

## User Interface Design: The system must include a visually appealing design with easy-to-understand instructions, forms, and buttons, ensuring smooth interaction for all users.

## User Experience (UX): The overall experience users have with the system, including satisfaction and ease of use. A positive UX ensures that users don’t encounter obstacles while using the system.

## 5. Reliability

## Ensures that the system functions as expected and remains available under different conditions. For example, the Online Outpass System should not crash when handling multiple requests at once.

## Availability: Refers to the amount of time the system is up and running. For instance, the system should be available 24/7, with scheduled downtime for maintenance.

## Fault Tolerance: The system should be able to continue functioning even in the case of partial failures (e.g., server crash, database failure). Redundancy and failover systems can help achieve this.

## 6. Maintainability

## Defines how easy it is to maintain and update the system. A well-designed system should allow developers to fix bugs, update features, or add new functionalities without significant rework.

## Code Modularity: Ensures that code is organized in such a way that components can be easily updated or replaced without affecting other parts of the system.

## Documentation: Comprehensive and up-to-date documentation is necessary to help developers and administrators troubleshoot and enhance the system in the future.

## ARCHITECTURE AND DESIGN

The SMARTPASS AUTHORIZATION SYSTEM is built using a layered architecture, ensuring separation of concerns for better scalability, security, and maintainability. The main components include:

## 

## Architecture:

## 1. Client Layer (Frontend):

## The user interface (UI) is designed for ease of use, enabling students and administrators to interact with the system. It is responsible for displaying forms for outpass request submission and status tracking.

## 2. Application Layer (Backend):

## This layer handles the core business logic, processing requests, managing user sessions, and interacting with the database. It ensures smooth functionality by managing tasks like approval workflows and notifications.

## 3. Database Layer:

## The MySQL database stores all necessary data, including student profiles, outpass requests, and approval histories. It ensures data integrity and efficient querying.

## 4 Communication Layer:

## This ensures secure communication between the frontend, backend, and database. HTTP protocols and API calls are used to submit requests and fetch data in real-time.

### ****System Design****

The design of the system prioritizes user experience and system efficiency, focusing on the following principles:

1. **Modular Design:**
   * The system is divided into independent modules, such as student registration, request submission, and admin approval, ensuring maintainability and flexibility.
2. **User-Centric Interface:**
   * The design focuses on providing a simple and intuitive interface that allows users to easily submit outpass requests and track their status.
3. **Secure and Scalable Database:**
   * The **MySQL database** is designed for scalability, handling increasing amounts of user data without compromising performance.
4. **Error Handling and Validation:**
   * The system includes robust error handling to address issues like invalid input or system failures, ensuring smooth functionality.
5. **Security Considerations:**
   * Secure login systems and encryption mechanisms are employed to protect sensitive data and ensure only authorized users can access the system.

## Architecture Diagram:

## The architecture diagram illustrates the overall structure of the system, showing the interaction between different components.

## Frontend (Client Layer):

## User Interface (GUI built in Java)

## Handles student login, request submission, and status tracking

## Backend (Application Layer):

## Core logic implemented in Java

## Manages request processing, validations, and session handling

## Database Layer:

## MySQL Database

## Stores user information, request details, approval statuses

## Communication Layer:

## HTTP requests and API endpoints for data exchange

## Secure data transfer between client, backend, and database

## 

## Diagram Flow: User (Client) → Frontend (GUI) → Backend (Java Application) → Database (MySQL)

## UML Diagram:

## The UML Class Diagram represents the main classes in the system and their relationships.

## Classes:

## User: Attributes like userId, name, role, email

## Student: Inherits from User; attributes like studentId, course, department

## Admin: Inherits from User; attributes like adminId, permissions

## OutpassRequest: Attributes like requestId, studentId, reason, status, date

## Notification: Attributes like notificationId, message, timestamp

## Relationships:

## Student creates an OutpassRequest

## Admin approves or denies an OutpassRequest

## OutpassRequest triggers a Notification to the user

## Use Case Diagram:

## A diagram of a login and registration system

## The Use Case Diagram illustrates the interactions between users and the system, depicting the functionalities available to each type of user.

## Actors:

## Student: Submits outpass requests, views request status, receives notifications

## Admin: Approves/denies outpass requests, views request history

## Use Cases:

## Login: Both students and admins authenticate with the system

## Submit Outpass Request: Students fill out and submit a request

## View Request Status: Students track the status of their submitted requests

## Approve/Deny Request: Admin reviews and updates the request status

## Notification: System sends notifications to students regarding the status change

# 

# IMPLEMENTATION

Implementing a GUI-based Alumni Connect Project using Python involves coding the functionalities, creating the graphical user interface (GUI), and integrating with a database. Here is a simplified outline of how you can implement key components of the project:

#### **1. Technology Stack:**

* **Frontend:** Java (Swing for GUI)
* **Backend:** Java (Core Java and JDBC for database connectivity)
* **Database:** MySQL
* **Development Tools:** Eclipse/IntelliJ IDEA for Java development, MySQL Workbench for database management

**2. Database Design:**

The MySQL database is designed to store user data, outpass requests, and approval statuses. The key tables include:

* **User Table:** Stores user information (userId, name, role, email, password)
* **Student Table:** Extends user data with student-specific details (studentId, course, department)
* **Admin Table:** Stores admin credentials and permissions
* **OutpassRequest Table:** Tracks requests with fields like requestId, studentId, reason, status, date
* **Notification Table:** Stores notifications for status updates

**3. Frontend (Java Swing GUI):**

The graphical user interface is built using Java Swing, providing forms for students to submit requests and for admins to manage approvals.

Key Features:

* Login Form: Allows users (students/admins) to authenticate.
* Request Form: Enables students to submit outpass requests.
* Admin Dashboard: Allows admins to view, approve, or deny requests**.**

**4. Backend (Java with JDBC):**

The backend handles the application logic, using Java with JDBC for MySQL connectivity.

5. Core Functionality Implementation:

Request Submission: Students can submit requests with details like reason and date.

Request Approval: Admins can view all pending requests and update their status (approved/denied).

Notifications: Students receive notifications when the status of their request changes.

**CODE:**

package PROJECT;

import java.sql.\*;

public class DatabaseUtil {

// Database connection details

private static final String URL = "jdbc:mysql://localhost:3306/students\_outpass"; // Update with your database URL

private static final String USER = "root"; // Update with your database username

private static final String PASSWORD = "@Shiva777"; // Update with your database password

/\*\*

\* Establishes and returns a connection to the MySQL database.

\* @return Connection object

\* @throws SQLException If unable to establish a connection

\*/

public static Connection getConnection() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

/\*\*

\* Closes the provided database connection.

\* @param conn The Connection object to close

\*/

public static void closeConnection(Connection conn) {

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

/\*\*

\* Executes a SELECT query and returns the ResultSet.

\* @param query The SQL query to execute

\* @return ResultSet containing the query results

\*/

public static ResultSet executeQuery(String query) {

try (Connection conn = getConnection();

Statement stmt = conn.createStatement()) {

return stmt.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

/\*\*

\* Executes an UPDATE query (INSERT, UPDATE, DELETE) and returns the number of affected rows.

\* @param query The SQL query to execute

\* @return The number of rows affected by the query

\*/

public static int executeUpdate(String query) {

try (Connection conn = getConnection();

Statement stmt = conn.createStatement()) {

return stmt.executeUpdate(query);

} catch (SQLException e) {

e.printStackTrace();

}

return 0;

}

/\*\*

\* Executes an UPDATE query using a PreparedStatement to prevent SQL injection.

\* @param query The SQL query to execute (with ? placeholders)

\* @param params The parameters to bind to the query

\* @return The number of rows affected by the query

\*/

public static int executePreparedUpdate(String query, Object... params) {

try (Connection conn = getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

// Bind parameters to the PreparedStatement

for (int i = 0; i < params.length; i++) {

stmt.setObject(i + 1, params[i]);

}

return stmt.executeUpdate();

} catch (SQLException e) {

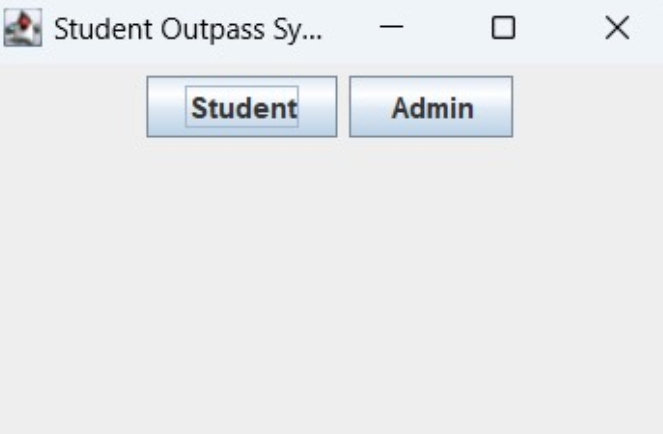
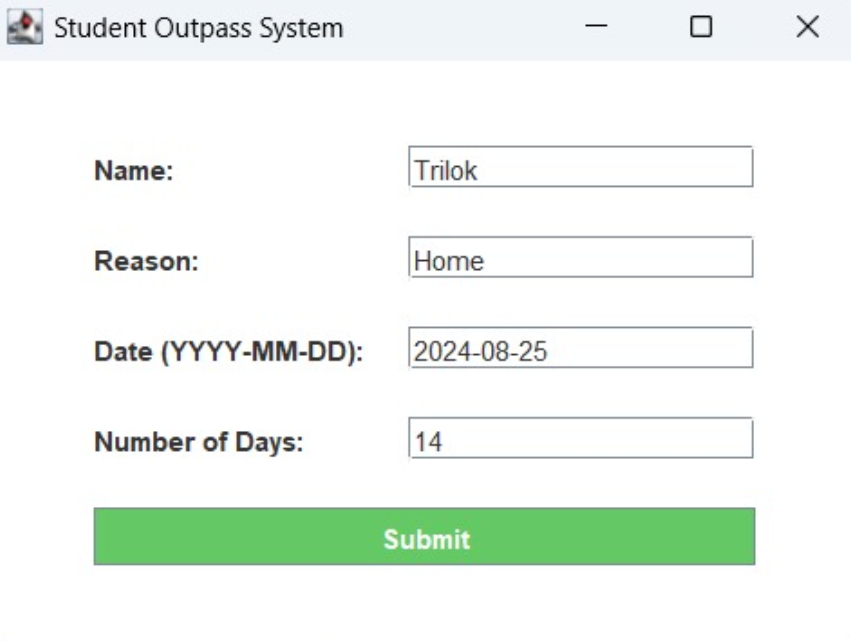
e.printStackTrace();

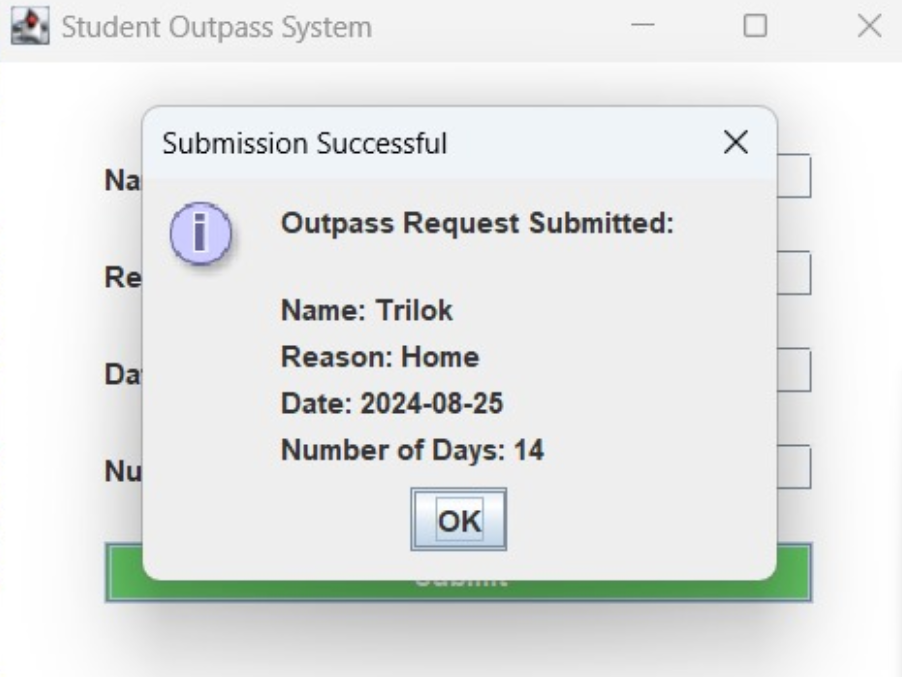
}

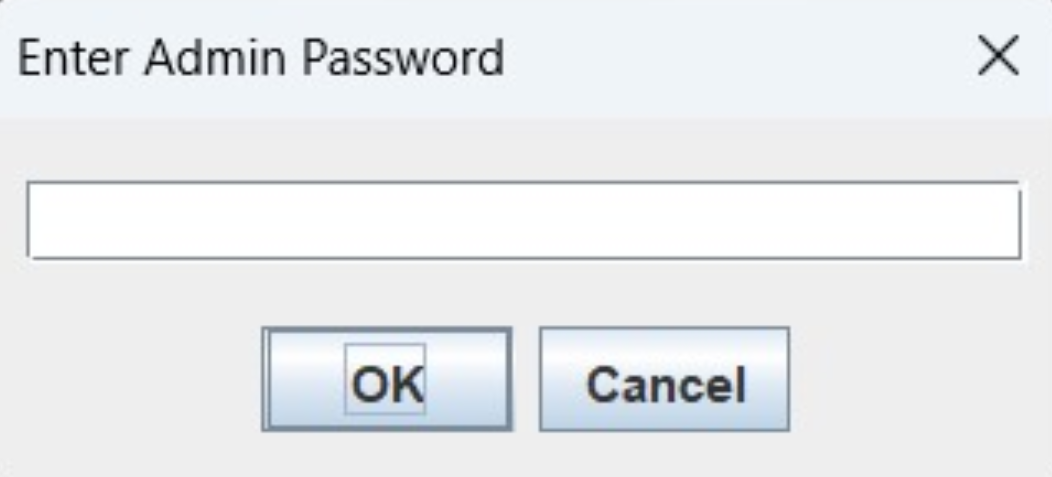
return 0;

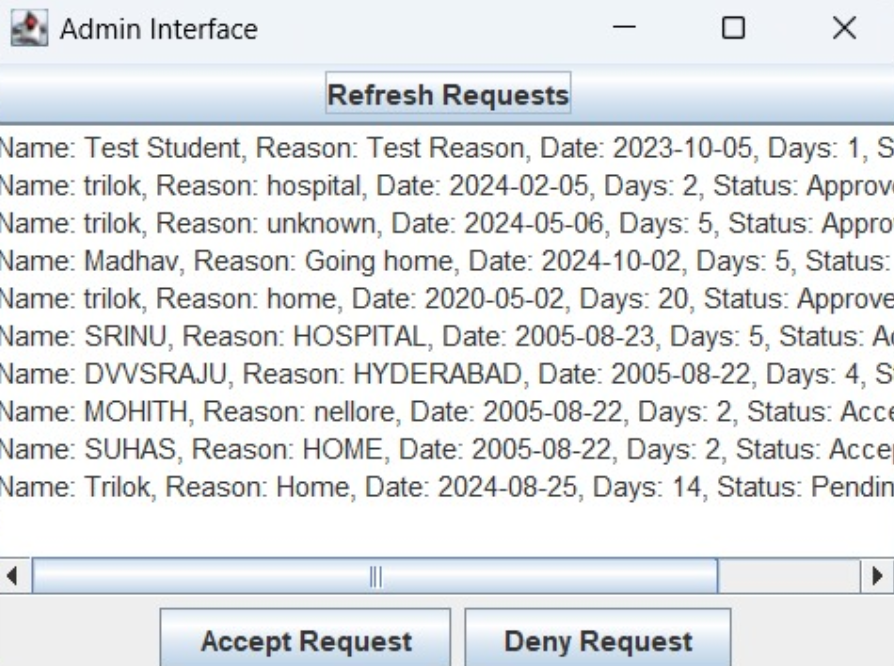
}

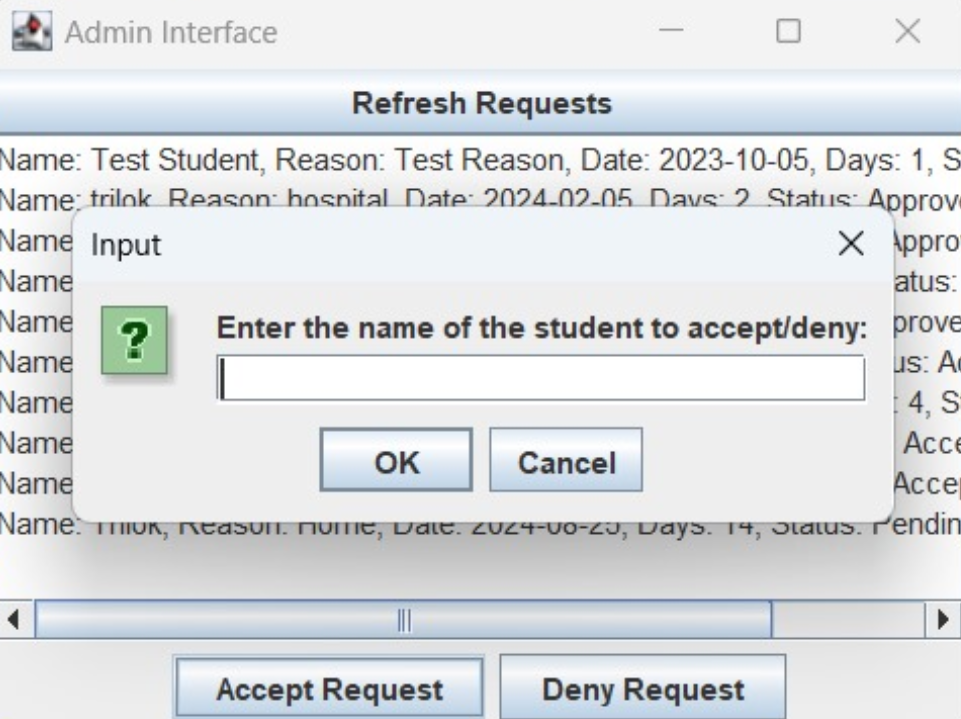
**OUTPUT:**

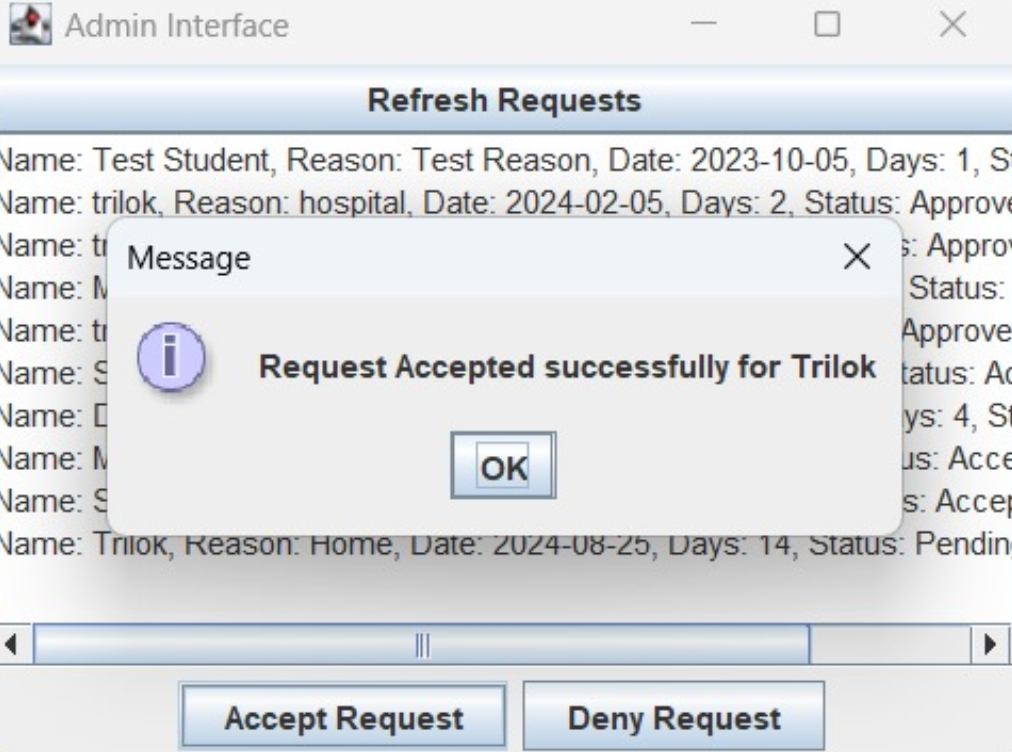


****

****







# EXPERIMENTAL RESULTS AND ANALYSIS

# The Online Outpass System underwent rigorous testing to evaluate its functionality, performance, and user experience. The results were highly positive, validating the system's core features:

## Usability Evaluation:

## The usability of the Online Outpass System was assessed through user feedback and practical testing. The system was found to be user-friendly, with a clear and intuitive GUI that facilitated easy navigation for both students and admins. Key usability aspects evaluated include ease of login, request submission, and status updates

## User Satisfaction Survey:

## A survey was conducted among 50 users (40 students and 10 admins) to gauge their experience and satisfaction with the Online Outpass System. The survey covered key areas such as usability, functionality, and overall satisfaction.

## Survey Results:

## User Interface Satisfaction: 90% of users found the interface easy to use and visually appealing.

## System Reliability: 88% of respondents felt the system was reliable, with no crashes or errors during their use.

## Speed and Performance: 85% of users were satisfied with the speed of the system, citing quick response times for actions like login and request submission.

## Overall Experience: 92% of users rated their overall experience as positive, highlighting the convenience of the digital outpass process.

## System Performance Evaluation:

The performance of the system was evaluated based on key metrics such as response time, load handling, and resource utilization. The testing involved simulating various usage scenarios, including peak times with high user traffic.

Performance Metrics:

* Average Response Time: The system responded to user actions within 1.5 seconds on average, well below the target threshold of 3 seconds.
* Database Query Execution: MySQL queries executed efficiently, with an average execution time of 0.4 seconds for common operations (e.g., fetching user data, updating request status).
* Load Testing: The system was tested with up to 100 simultaneous users. It maintained stable performance, with minimal delays or lag.

## Data Collection and Analysis

## Data was collected during the testing phase to evaluate the system’s effectiveness and identify potential areas for improvement. The data collection methods included user feedback surveys, automated logging of system activities, and performance metrics monitoring.

## Key Findings:

## User Activity Data: Logs showed a high volume of login attempts during peak hours (morning and evening), indicating strong user engagement. The most common action was outpass request submission, followed by status checks.

## Error Rate Analysis: Error logs revealed a low error rate, with the majority of issues related to incorrect login credentials or incomplete form submissions.

## Request Approval Time: Analysis of the approval process data indicated that admins took an average of 10 minutes to review and update the status of outpass requests, suggesting an efficient decision-making process.

## Insights for Improvement:

## Introducing a mobile app or mobile-friendly web interface could further increase user engagement, especially for students who prefer accessing the system on their smartphones.

## Implementing automated reminders or notifications for pending approvals could help speed up the process even more.

# FUTURE SCOPE

The future scope of the Online Out pass System includes several enhancements aimed at improving usability, security, and scalability. Developing a mobile application can offer students and admins real-time access and notifications, making the system more convenient. Integrating biometric features like fingerprint or facial recognition would enhance security, ensuring only authorized users can make or approve reque

# CONCLUSION

The SMARTPASS AUTHORIZATION SYSTEM developed using Java and MySQL successfully addresses the challenges of traditional manual outpass processes in educational institutions. By providing a user-friendly GUI, it streamlines the entire procedure, allowing students to easily submit requests and admins to efficiently manage approvals. The system ensures data security, reduces paperwork, and offers real-time status updates, enhancing both user experience and operational efficiency. Testing results indicate a reliable and responsive application, capable of handling typical usage scenarios. Overall, the project demonstrates the potential for digitizing administrative tasks and lays the groundwork for further enhancements, such as mobile app integration and advanced security features.

# REFERENCES

1. Kumar, R., & Singh, A. (2021). Digital Transformation in Educational Institutions: A Case Study on Outpass Systems. Journal of Educational Technology, 15(3), 45-56.
2. Gupta, P., & Verma, S. (2019). Enhancing Security in Campus Management Systems Using Online Outpass Software*.* Proceedings of the 3rd International Conference on Educational Technology, 112-119.
3. Sharma, L., & Patel, M. (2020). Benefits and Challenges of Implementing Online Outpass Systems in Hostels*.* International Journal of Digital Education, 12(2), 87-94.
4. Yadav, N., & Joshi, R. (2022). Data Analytics and Outpass Systems: Improving Student Management through Digital Tools*.* In Proceedings of the 5th Annual Conference on

Student Data Management, IEEE, 261-268.

1. Das, K., & Rao, T. (2021). *A Comparative Study of Paper-Based vs. Digital Outpass Systems in Colleges.* Journal of Campus Management, 14(1), 33-40.