UniFix: A System to Manage Track and Resolved Student Concerns

by

Ronjon Kumar Roy Taron (Exam Roll:230011)

Md. Shaon Khan (Exam Roll:230023)

Humayara Mustafa Kuasha (Exam Roll:230036)

A Project Report submitted to the

Institute of Information Technology

in partial fulfillments of the requirements for the degree of

Bachelor of Science in

Information and Communication Technology

Supervisor: Rashed Mazumder, Ph.D.

Associate Professor, Institute of Information Technology

Institute of Information Technology

Jahangirnagar University

Savar, Dhaka- 1342

20 February 2025

DECLARATION

We hereby declare that this project is based on the results found by ourselves. If any errors and omissions happen then that's our responsibility to set them. Materials of work found by other researchers are mentioned by reference.

Ronjon Kumar Roy Taron

Exam Roll:230011

Md. Shaon Khan

Exam Roll:230023

Humayara Mustafa Kuasha

Exam Roll:230036

CERTIFICATE

This is to clarify that the project entitled UniFix: Streamlining Student Issues with Java has been prepared and submitted by Ronjon Kumar Roy Taron, Md. Shaon Khan and Humayara Mustafa Kuasha in partial fulfillment of the requirement for the degree of Bachelor of Science (honors) in Information and Communication Technology on February 20, 2025.

Assoc. Prof. Rashed Mazumder, Ph.D.

Supervisor

Accepted and approved in fulfillment of the requirement for the degree Bachelor of Science (honors) in Information and Communication Technology.

ACKNOWLEDGEMENT

First and foremost, we would want to sincerely thank God for His gifts and direction, which have enabled us to finish this project. This accomplishment would not have been possible without His grace.

We also want to express our sincere gratitude to everyone who gave us their insightful opinions. Their input has substantially raised the caliber of our work and deepened our comprehension.

We would especially like to express our gratitude to our parents, who have been our steadfast source of love, support, and encouragement along our journey. Their selflessness and faith in our potential have motivated us to reach our objectives and push the envelope.

We would like to express our gratitude to our distinguished supervisor, **Assoc. Prof. Rashed Mazumder, Ph.D**., whose knowledgeable direction and perceptive oversight were crucial to the accomplishment of this project. We were led at every turn by his steadfast assistance, wise counsel, and supply of vital materials and expertise. His guidance has been crucial in helping us stay Motivate and prepared to finish this job.

Lastly, we would like to thank our esteemed Sir for giving us the chance to investigate and put our knowledge into practice. His confidence in our ability to carry out our project's ideas has given us a valuable learning experience that will help us in the future. We are truly thankful for his trust and encouragement throughout this process.

Table of Contents

List of Figures

Abstract

UniFix is a Java-based student issue tracking system aimed at improving the reporting, prioritization, and resolution of campus-related concerns. It offers a structured and transparent platform where students can report issues, vote on concerns, and file complaints about unresolved matters. The system prioritizes the most critical issues through a voting mechanism, allowing administrators to allocate resources effectively.

Administrators utilize UniFix to track, manage, and resolve reported issues while ensuring accountability through a complaint review system. Real-time status updates and a notification system keep students informed about the progress of their submissions. A dedicated badge notification button alerts students when there is an update on an issue or a response to a complaint, helping them stay engaged in the resolution process.

To support data-driven decision-making, UniFix features a comprehensive reporting and analytics module. This tool showcases issue trends, resolution rates, and student engagement metrics through pie charts, line graphs, and dynamic counters. The system's interactive dashboard provides a clear visual representation of issue resolution performance across the campus.

UniFix is developed using Java Swing for the user interface, MySQL for database management, and JFreeChart for graphical reporting. It operates on a three-tier architecture, which ensures scalability, security, and efficiency. Designed as a desktop application, the system is structured for potential future expansion into cloud-based or mobile platforms.

Keywords: Issue Tracking, Campus Problem Resolution, Voting Mechanism, Campus Issue Management, Transparency & Accountability, Notification System, Statistical Reporting.

UniFix:A System to Manage Track and Resolved Student Concerns

1.Introduction

Universities frequently encounter challenges in effectively resolving campus-related issues reported by students. Common problems include unauthorized access to campus premises, safety concerns, poor cafeteria food quality, inadequate seating arrangements in classrooms, damaged roads, broken streetlights, delayed exam result publications, lack of essential lab equipment, and incidents of harassment or bullying. These issues significantly impact student life and academic performance.
Traditional complaint systems, often reliant on paper forms, suggestion boxes, or emails, suffer from delays, lack of tracking mechanisms, and limited transparency. Consequently, unresolved issues and a lack of accountability create frustration among students.

UniFix bridges this gap by providing an interactive platform that enables students to:

- Report campus-related concerns easily.
- Prioritize issues through a voting mechanism, reflecting their significance.
- Track the progress of issue resolutions in real-time.

Administrators, on the other hand, benefit from a structured workflow that ensures accountability and timely resolution of reported concerns. The platform fosters transparent communication between students and university authorities, driving a culture of responsibility and student engagement.

2. Background Study

In many universities, issue reporting and resolution processes are still manual, leading to delays, inefficiencies, and poor student engagement. Paper-based complaint forms, suggestion boxes, and emails are commonly used, but they lack real-time tracking, status updates, and structured workflows. Even in cases where digital complaint systems are available, they are often limited to one-way communication without effective feedback mechanisms or priority-based sorting. This leads to:

- Unclear timelines for issue resolution.
- Frustration due to unresolved complaints.
- Limited student participation in decision-making processes.

While some institutions have implemented digital grievance portals, these systems mainly serve administrative convenience, sidelining student involvement and feedback. This results in a disconnect between students' needs and administrative actions.

UniFix addresses these challenges by offering:

- Real-time Status Updates: Students can track issue resolutions at every step.
- Priority-Based Voting: Issues are sorted based on student votes, ensuring that the most critical concerns are addressed first.
- Accountability Mechanisms: The system includes a complaint feature that allows students to report administrative inaction, ensuring accountability and transparency.
- Analytical Insights: Visual progress reports using pie charts provide an overview of issue resolutions within a set timeframe, enhancing decision-making processes for university authorities.

By incorporating these features, UniFix ensures a participatory, transparent, and efficient approach to campus issue management, thereby increasing student satisfaction and administrative accountability.

3. Proposal

UniFix is designed as an interactive digital platform that streamlines communication between students and administrators, ensuring efficient issue reporting and resolution. The system is built to enhance transparency, accountability, and responsiveness within campus management. Its key objectives include:

- **Effortless Issue Reporting:** Allowing students to easily report campus-related concerns, ensuring no issue goes unnoticed.
- **Priority-Based Voting:** Enabling students to vote on reported issues, helping administrators prioritize them based on collective significance.
- **Structured Resolution Workflow:** Providing administrators with a clear, organized workflow to efficiently address and resolve issues, minimizing delays.
- **Accountability Through Feedback:** Incorporating a complaint feature that empowers students to report administrative negligence, ensuring transparency and accountability.
- **Real-Time Progress Tracking:** Generating intuitive visual progress reports, including pie charts, to track issue resolution within a week, fostering trust and engagement among students.

4. Technical Feasibility

UniFix is technically feasible due to its use of widely adopted and reliable technologies. It is developed using **Java** for core functionality, with **NetBeans** for the user interface and **Java Swing** for creating an interactive graphical experience. The system utilizes **MySQL** for robust database management, managed through **phpMyAdmin** for efficient database administration.

By using phpMyAdmin, UniFix simplifies database management tasks, such as creating and maintaining tables, executing queries, and ensuring secure data handling. The system is designed to work with a local or shared MySQL database setup, making it cost-effective and easy to maintain.

UniFix also supports centralized database connectivity, allowing access from multiple devices, thus ensuring accessibility and operational efficiency. The use of proven technologies like Java, MySQL, and phpMyAdmin ensures scalability, easy maintenance, and future upgrades, making UniFix a reliable and sustainable solution for campus issue tracking.

5. Technical Description

- **Programming Language:** Java (using **Swing** for the graphical user interface and **JDBC** for seamless database connectivity)
- **Integrated Development Environment (IDE): NetBeans**, chosen for its robust features and ease of use in Java development
- **Database Management: MySQL**, managed through **phpMyAdmin** for efficient database administration, with **XAMPP** used for local testing and development
- **Authentication Mechanism:**
    - **Student Login:** Secured using registration number and password
    - **Admin Login:** Secured using email and password
- **Data Storage Structure:** Utilizes a single **MySQL** database containing multiple tables to efficiently manage:
    - User data (students and administrators)
    - Issue tracking details
    - Voting records for issue prioritization
    - Complaints regarding administrative negligence
- **Deployment Plan:** Initially designed as a **desktop-based application**, with the potential for future expansion into a **cloud-based platform** to enhance accessibility and scalability.

6. System Design

UniFix follows a modular architecture, ensuring smooth interaction between components while maintaining data security. It is organized around two primary user roles:

- **Students:**
    - Report campus issues, vote to prioritize them, and track progress.
    - File complaints if issues are not addressed timely.
- **Administrators:**
    - Manage reported issues, prioritize them based on votes, and provide updates.
    - Address complaints about administrative negligence.

6.1 Purpose

The purpose of **UniFix** is to establish an efficient and transparent campus issue tracking system that enables students to report problems, track their resolution progress, and hold administrators accountable for timely action. The system aims to bridge the communication gap between students and campus authorities by providing a structured platform where students can express their concerns and monitor resolution in real time.

UniFix tackles issues like poor infrastructure, security threats, and administrative delays in a systematic and organized manner, improving the overall campus environment. Through the voting mechanism, students can prioritize the most pressing issues, ensuring that authorities address them promptly and appropriately. This transparent process enhances accountability and fosters a better relationship between students and administrators, ultimately leading to an improved campus experience for all.

6.2 Scope

The scope of **UniFix** is designed to serve both students and administrative authorities of the university. It offers a platform for students to report campus-related issues, track their resolution progress, and engage in the prioritization process. For administrators, the system provides tools to manage, resolve, and track the progress of reported concerns. This system ensures transparent communication between students and administrators, streamlines the prioritization of issues, and empowers decision-making through analytical data reports.

Features in Scope:

- **Secure login** for both students and administrators to ensure privacy and data protection.
- **Issue reporting** allows students to provide detailed descriptions of their concerns, facilitating accurate reporting.
- **Voting system** enables students to express the significance of reported issues, aiding in their prioritization.
- **Complaint mechanism** allows students to report administrative negligence if issues are not addressed within a set timeframe.
- **Status updates and progress tracking** give students real-time insight into the status of reported issues.
- **Data visualization** through pie charts and analytical reports to provide clarity on the resolution process and the effectiveness of actions taken.
- **Role-based access control** ensures that students can report and vote on issues, while administrators have the authority to resolve and update statuses.

Out of Scope:

- **Non-campus-related issues** such as personal concerns or external matters that fall outside the university's jurisdiction.
- **Real-time messaging** between students and administrators is currently limited to issue status updates; direct communication will not be facilitated through the system.
- **Automated issue resolution** as all decisions and actions to address issues are made manually by administrators based on student input and their internal processes.

6.3 Architecture Overview

UniFix is designed using a **three-tier architecture**, which ensures scalability, maintainability, and a clear separation of concerns. The system consists of three main layers:

- **Presentation Layer:**
  The **graphical user interface (GUI)** is developed using **Java Swing**. This layer is responsible for providing an intuitive interface for both students and administrators, enabling them to interact with the system by reporting issues, voting, and tracking progress.
- **Application Layer:**
  The **core logic** of UniFix resides in this layer. It handles key operations such as user authentication, issue tracking, voting, complaints, and generating progress reports. The

application layer acts as the intermediary between the presentation and data layers, processing and managing user inputs and business logic.

- **Data Layer:**
  The **MySQL database** stores all essential data, such as user information (for both students and administrators), reported issues, voting data, complaints, and administrative actions. **phpMyAdmin** is used to manage the database, allowing administrators to create, modify, and query the data stored in MySQL. phpMyAdmin provides a web-based interface to simplify database management tasks, ensuring seamless storage and retrieval of data.

6.4 Student Features

The **Student Features** section of UniFix offers various functionalities that enable students to report issues, interact with the system, and track their concerns efficiently. The key features for students are:

- **Sign up**: Students can register using their **registration number** and **password** to create an account, ensuring secure access to the system.
- **Login**: Once registered, students can log in to the system and access the **student desk**, where they can view and manage their reported issues.
- **Profile Update**: Students have the ability to **update** their profile information, such as contact details, except for the **registration number**, which remains fixed.
- **Submit Issues**: Through the **Inform Issue tab**, students can report any campus-related concerns or problems they encounter, providing detailed descriptions for accurate tracking and resolution.
- **Vote on Issues**: Students can **view all reported issues** and participate in the voting system to indicate the significance of each issue, allowing them to prioritize issues they believe need urgent attention.
- **Track Issue Status**: Students can monitor the **status** of their reported issues and see progress updates through the **Issue Solved tab**, ensuring they are kept informed about the resolution process.
- **Submit Complaints**: If there are delays or administrative negligence in resolving issues, students can **submit complaints** against the responsible administrators to hold them accountable.

6.5 Admin Features

The **Admin Features** section of UniFix is tailored to give administrators the tools they need to effectively manage and resolve student-reported issues. Key features for administrators include:

- **Sign up**: Administrators can register using their **email** and **password** to create a secure account, enabling them to access and manage the system.
- **Login**: After registration, admins can log in to access the **admin desk**, where they can manage and monitor the entire issue resolution process.
- **Search for Students**: Admins have the ability to **search for students** based on their registration numbers, view their profiles, and access details related to their reported issues.
- **Manage Student-Reported Issues**: Admins can **manage** all reported issues submitted by students, including updating the status of issues to reflect progress, such as "In Progress," "Resolved," or "Closed."
- **Provide Resolution Updates**: Admins are required to provide updates on the resolution of issues, ensuring that all issues are addressed within **seven days**, fostering a timely response to student concerns.
- **Address Complaints**: If students file complaints about delays or administrative negligence, administrators are responsible for reviewing and resolving these complaints promptly to maintain accountability.
- **View and Analyze Statistics**: Admins can **view and analyze** issue resolution data using **pie charts**, which offer visual insights into the status of various issues, the rate of resolution, and areas requiring attention.

7 System Diagrams

7.1 Data Flow Diagrams(DFD)

A **Data Flow Diagram (DFD)** represents how information flows within the UniFix system, showing interactions between students, administrators, and the system. It highlights key processes such as:

1. **Students**: Register, log in, report issues, vote on issues, track progress, and submit complaints.
2. **Administrators**: Log in, manage issues, update statuses, address complaints, and generate reports.

3. **System**: Authenticates users, stores data, tracks issue progress, and generates visual reports like pie charts.

The DFD provides a clear understanding of data movement and system functionality, ensuring smooth interaction between users and the platform.

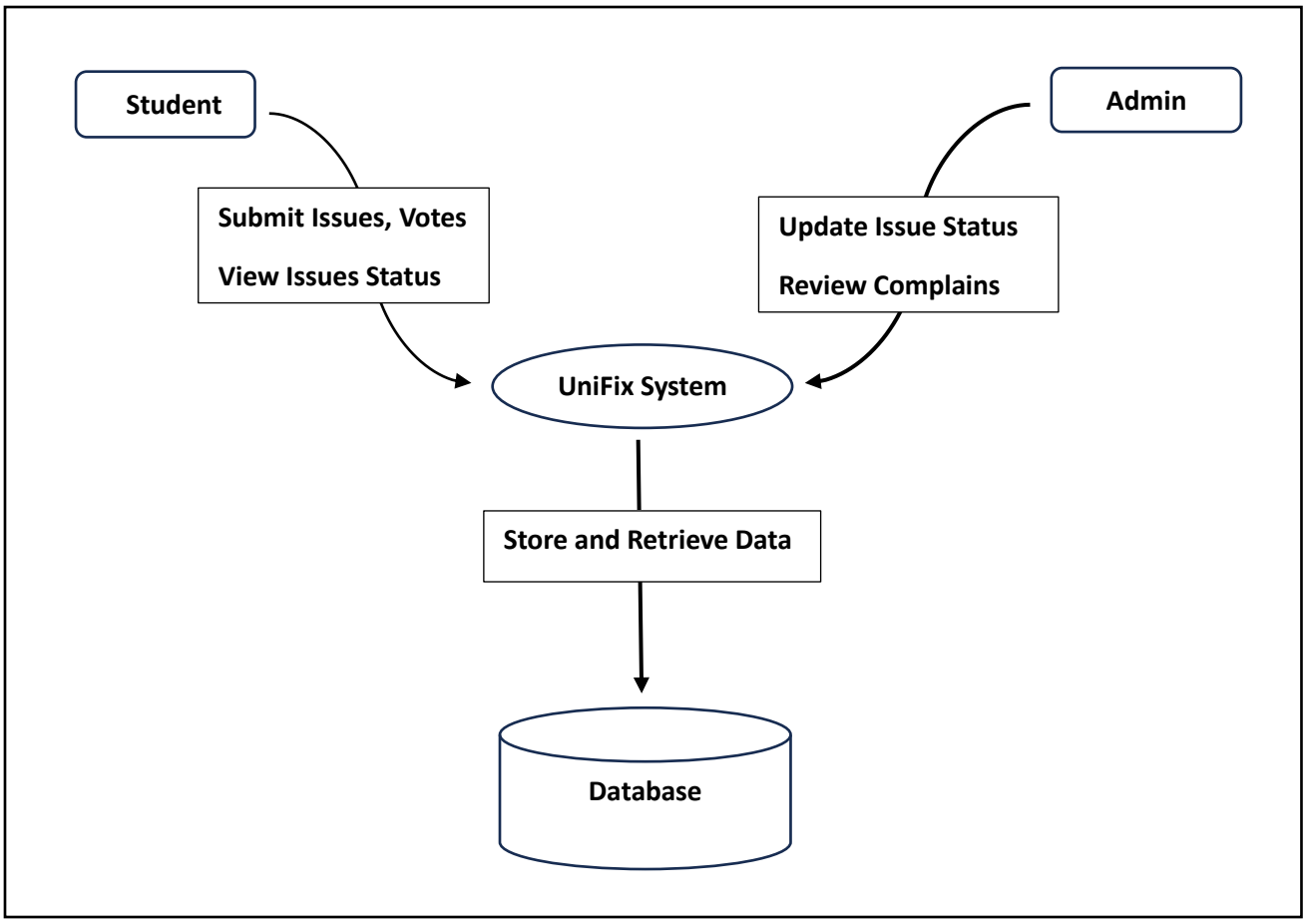

Fig 7.1.: Data Flow Diagram of UniFix

7.2 Use Case Diagram

The Use Case Diagram illustrates the interactions between users (students and administrators) and the features of the UniFix system. It defines the roles and actions each user can perform, ensuring clear understanding of the system's functionality. Key scenarios include:

- **Students:**
    - Report issues
    - Vote on issues
    - Submit complaints
    - Track issue progress

- **Administrators:**
    - Manage issues
    - Update issue statuses
    - Resolve complaints
    - View reports and statistics

This diagram helps developers understand role-based actions and ensures a structured approach to system design, clarifying how different users will interact with the system.

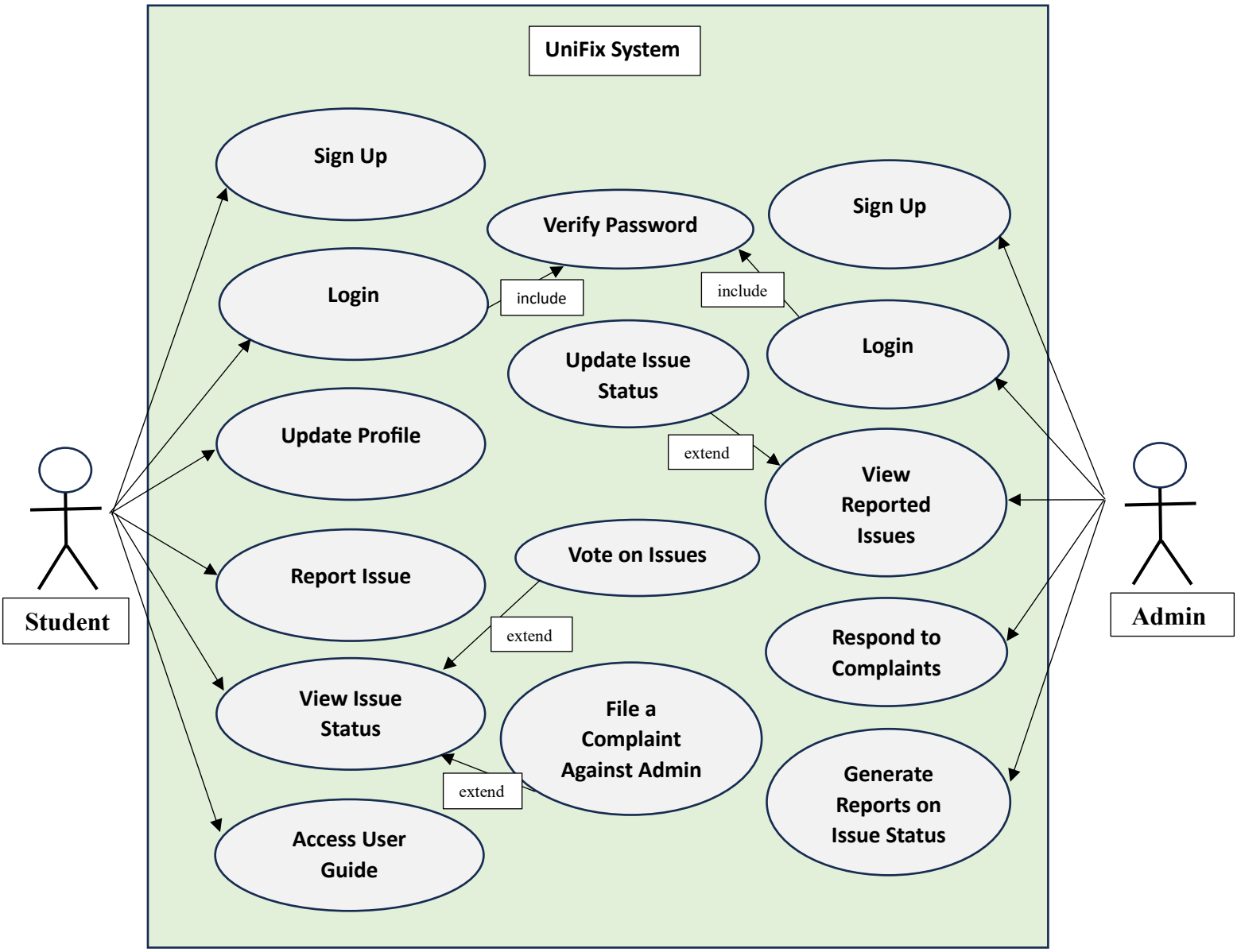

Fig-7.2.: Use Case Diagram of UniFix

7.3 ER Diagram

The Entity-Relationship (ER) Diagram outlines the structure of the UniFix database, identifying key entities and their relationships. The primary entities include:

- **Students:** Represented with attributes like registration number, name, email, and password.

- **Issues:** Describes each reported issue with attributes like issue ID, description, and status.
- **Votes:** Tied to issues, capturing the votes from students to prioritize issues.
- **Complaints:** Logs student complaints regarding administrative delays or negligence.
- **Administrators:** Includes admin credentials, roles, and responsibilities for managing issues.

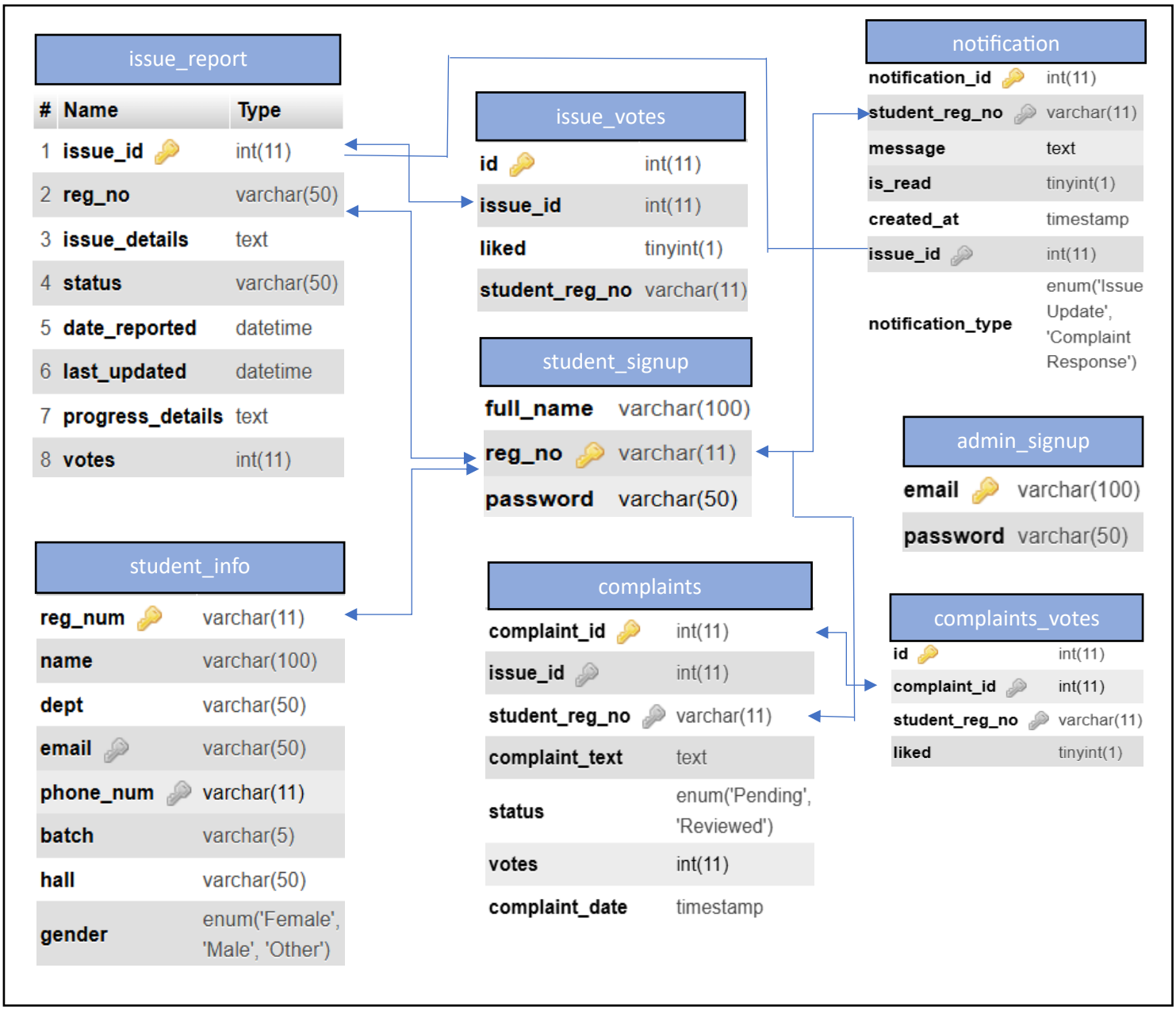

Fig-7.3.: ER Diagram of UniFix

8. Functional Requirements

Functional requirements define the essential operations and interactions that UniFix must perform to ensure it operates as intended.

8.1 Input Requirements

- **Student Features**:
    - Students can sign up and log in using their registration number and password.
    - Students can report issues, providing relevant details like issue type, description, and date of occurrence.
    - Students can vote on existing issues to indicate their urgency or priority.
    - Students have the option to file complaints against unresolved or delayed issues.
- **Administrator Features**:
    - Administrators can update the statuses of reported issues, such as "Submitted," "Assigned to Authority," "Action Taken," and "Resolved."
    - Administrators can generate reports to track the progress of issue resolution, including status updates and analytics on the number of issues resolved.

8.2 Output Requirements

- **Student Features**:
    - Students should receive real-time updates on the status of their reported issues, such as when they are being reviewed, assigned, or resolved.
    - The system should display a list of the most voted issues, allowing administrators to prioritize them accordingly.
    - Students should be able to view a comprehensive list of all submitted complaints for review and resolution.
- **System Features**:
    - The system should provide visual charts, such as pie charts, to help students track the progress of issue resolution and understand the current status.
    - Administrators should receive automated reminders for unresolved issues, ensuring timely follow-up and resolution within the specified time frame.

9. Performance Requirements

The performance requirements for UniFix ensure that the system operates efficiently, maintains reliability, and delivers a smooth user experience.

- **System Response Time**:
    - The system should process and display issue submissions and votes within **2 seconds** of submission.
    - Issue status updates should reflect in real-time or within **5 seconds** for all users, ensuring timely updates across the platform.
- **Scalability**:
    - UniFix should be able to handle at least **500 concurrent student users** without experiencing any significant degradation in performance or responsiveness.
- **Database Performance**:
    - Database queries should execute in **less than 1 second** for common actions, such as fetching issue lists, votes, or updating records.
- **Availability**:
    - The system should maintain **99% uptime** under normal usage conditions, ensuring accessibility and reliability for users.
- **Security Requirements**:
    - User passwords must be **encrypted** before being stored to ensure data security and privacy.
    - **Only authenticated users** should be able to access system functionalities, preventing unauthorized access.

10. Software Requirement Specification (SRS)

The Software Requirement Specification (SRS) outlines the necessary hardware and software components needed to deploy and run UniFix efficiently.

10.1 Required Hardware

- **Processor**:
    - **Minimum**: Intel Core i3 (or equivalent)
    - **Recommended**: AMD Ryzen 5 (offers better multi-core performance, higher clock speeds, and enhanced efficiency for multitasking)
- **RAM**:

    - **Minimum**: 4GB
    - **Recommended**: 8GB or higher for better performance, especially when handling multiple simultaneous users.
- **Storage**:
    - **Minimum**: 10GB of free disk space for database storage and logs.
- **Network**:
    - A stable internet connection is required for multi-device database access, especially if using **phpMyAdmin** for database management.

10.2 Required Software

- **Operating System**:
    - Windows 7 or later, macOS, or Linux.
- **Development Environment**:
    - **NetBeans IDE** (or any Java-supported IDE).
- **Programming Language**:
    - **Java** (Swing for the User Interface, JDBC for database connectivity).
- **Database**:
    - **phpMyAdmin** for MySQL database management (XAMPP for local database setup).
- **Additional Libraries**:
    - **JDBC Driver** for MySQL.
    - **JFreeChart** for visualizing issue resolution and statistics.

11. Implementation

The implementation of **UniFix** follows a structured development approach, utilizing **Java** for the core logic and **MySQL** for data management. The system is developed as a desktop-based application using **NetBeans IDE**, with **Java Swing** providing the graphical user interface (GUI). The application is designed to ensure seamless interaction between students and administrators while maintaining data security and efficiency.

11.1 Development Environment

The development of **UniFix** was carried out using the following tools and technologies:

- **Programming Language**: Java (Swing for GUI, JDBC for database connectivity)
- **Integrated Development Environment (IDE)**: **NetBeans**
- **Database Management System (DBMS)**: **MySQL** (XAMPP for local testing)
- **Version Control**: Manual backup (GitHub integration planned for future development)

11.2 User Interface Development

The user interface (UI) is built using **Java Swing**, providing an intuitive and interactive design for both students and administrators. The main UI components include:

- **Home Page**:
  - Provides an overview of the system and login options for both students and administrators.
- **Student Desk**:
  - Allows students to:
    - Report campus issues.
    - Vote on existing issues based on priority.
    - Track the progress of their issues and see the status updates.
    - Submit complaints regarding unresolved issues.
- **Admin Desk**:
  - Enables administrators to:
    - Review reported issues.
    - Update the status of issues (e.g., Submitted, Assigned, Action Taken, Resolved).
    - Address complaints filed by students regarding administrative delays.

11.3 Database Implementation

The **UniFix** database is implemented using **MySQL**, ensuring structured data storage and efficient retrieval. The database consists of several key tables that manage different aspects of the system's functionality:

- **student_info**: Stores student details such as registration number, name, department, and contact information.
- **issue_report**: Maintains records of reported issues, including submission date, status updates, and associated votes.
- **issue_votes**: Tracks votes cast by students to indicate the urgency of each reported issue.
- **admin_signup**: Stores login credentials for administrators, ensuring secure access to the admin desk.
- **complaints**: Records complaints submitted by students against administrative delays or negligence.

11.4 Functional Implementation

1. **Student Registration and Login**
    - Students register using their university registration number and password.
    - The system verifies login credentials using **MySQL** authentication to ensure secure access.
2. **Issue Reporting**
    - Students can submit campus-related issues, providing a description of the concern.
    - Each issue is stored in the database, and the system assigns a default status of **Submitted**.
3. **Voting System**
    - Students can vote on reported issues to indicate their level of urgency.
    - Votes are recorded in the **issue_votes** table, and the system automatically updates the total vote count for each issue.
4. **Issue Management by Administrators**
    - Administrators review reported issues and update their status to one of the following:
        - **Assigned to Authority**
        - **Action Taken**
        - **Resolved**
        - **Unresolved** (if no action is taken within seven days)
5. **Complaint Mechanism**
    - Students can file complaints regarding delays in issue resolution.
    - Complaints are stored in the **complaints** table, and administrators are required to address them.
6. **Progress Report Generation**
    - The system generates visual representations of issue resolution progress using **pie charts**.
    - This feature helps both students and administrators analyze the efficiency of issue resolution over time, offering insights into areas that need improvement.

11.5 Testing and Debugging

The implementation of **UniFix** included thorough testing to ensure system reliability and functionality across various modules. The testing process involved the following steps:

- **Unit Testing**: Each individual module of the system (such as user registration, issue reporting, and voting) was tested independently to ensure correct functionality and output.
- **Integration Testing**: After verifying the modules independently, the interactions between different modules (like issue reporting, voting, and admin updates) were tested together to ensure proper data flow and system cohesion.
- **User Testing**: Simulated user interactions were conducted to ensure that the system's interface was intuitive, and its performance met expectations under realistic usage conditions.

11.6 Challenges and Solutions

During the implementation phase, the following challenges were encountered, along with their solutions:

- **Ensuring Multi-Device Access**:
    - **Challenge**: Initially, the system was limited to a single localhost device, which restricted accessibility.
    - **Solution**: A centralized database connection was explored for future scalability, ensuring that the system could accommodate multiple devices and users simultaneously.
- **Anonymous Complaint Submission**:
    - **Challenge**: The system required an anonymous complaint feature, but ensuring its security was a concern.
    - **Solution**: Additional security measures were implemented, including encryption and validation checks, to prevent misuse and maintain the integrity of the system.
- **Real-time Status Updates**:
    - **Challenge**: Ensuring real-time status updates across users without causing excessive resource usage.
    - **Solution**: Database polling was optimized to retrieve status updates efficiently, using minimal resources, ensuring smooth user experiences.

11.7 Future Enhancements

Future improvements for **UniFix** include the following key upgrades:

- **Cloud-based Database Integration**:
    - For remote access and greater flexibility, the system will be enhanced by integrating a cloud-based database, enabling students and administrators to access the system from anywhere.
- **Mobile Application Development**:
    - To increase accessibility, a mobile application will be developed, allowing students and administrators to interact with the system on the go, enhancing user convenience.
- **Advanced Analytics for Issue Resolution Performance**:
    - The system will be enhanced with advanced analytics tools to track issue resolution performance, providing insights into system efficiency and areas for improvement.

11.8 User Interfaces

11.8.1. UniFix User Selection Interface

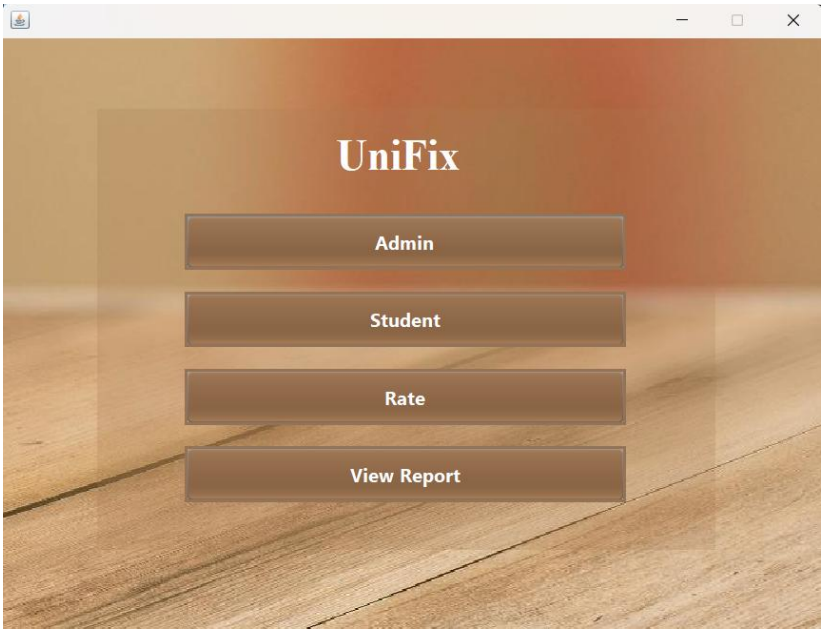

Fig-11.8.1.:UniFix User Selection Interface

The **UniFix Interface** is the main entry point of our project. It serves as the starting interface from where users can navigate to different sections of the system. This interface features three primary buttons:

- **Student Button**:
  By clicking on this button, users are directed to the **Student Login Page**, which grants access to the **Student Desk**. This section is designed specifically for student-related functionalities.
- **Admin Button**:
  This button redirects users to the **Administration Login Page**, leading to the **Admin Desk**. It is intended for administrative tasks and system management.
- **Rate Button**:
  The purpose of this button is to allow users to evaluate the effectiveness of the **UniFix System**. Users can rate the system on a scale of 1 to 5, providing feedback on their experience.

11.8.2. Student Login Page

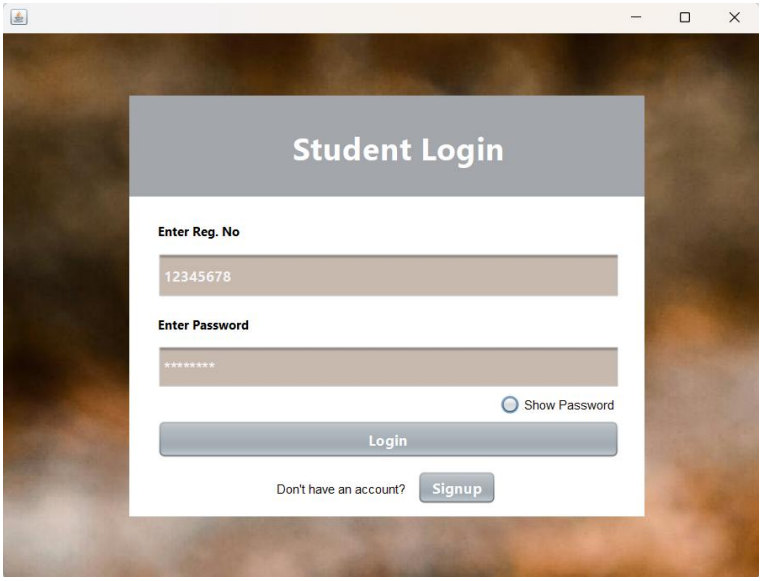

Fig-11.8.2.: Student Login Page

The **Student Login Page** allows students to access the **Student Desk** by verifying their credentials. The features of this interface are as follows:

- **Login Functionality**:
  A student can log in by entering their **Registration Number** and **Password**. Upon clicking the **Login Button**, the system validates the credentials. If correct, the student is granted access to the **Student Desk**.
- **Sign Up Navigation**:
  If a student does not have an account, they must first create one. This is achieved by clicking the **Sign Up Button** located on the **Login Page**, which redirects them to the **Sign Up Interface** for account creation.

11.8.3. Student Sign Up Page

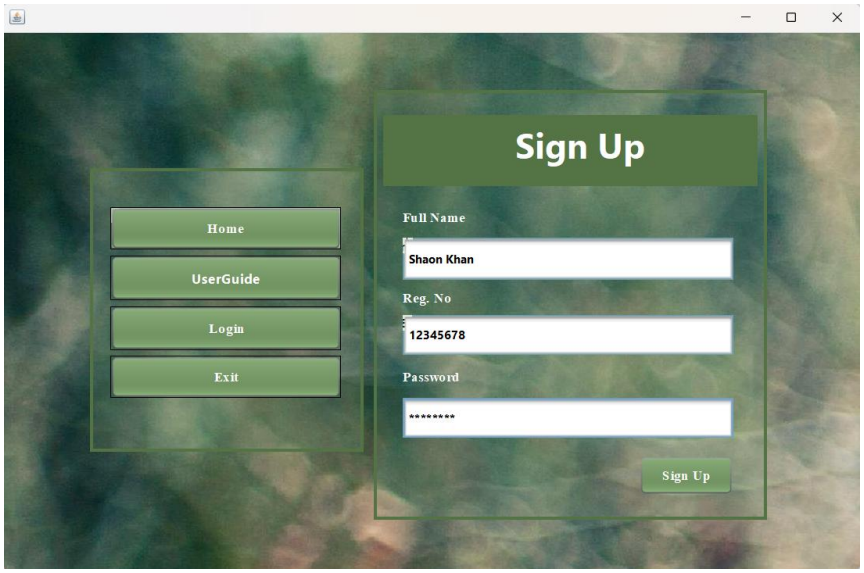

Fig-11.8.3.: Student Sign Up Page

The **Sign Up Page** allows new students to create an account, which is mandatory before accessing the **Student Desk**. The process flow is as follows:

- **Account Creation**:
  A student must provide the following details:
  - **Full Name**
  - **Registration Number (Reg. No)**
  - **Password**
    After entering this information, the account is created, allowing the student to proceed to the **Login Page**. Once logged in, they can access the **Student Desk**.
- **Additional Buttons and Their Functions**:
  - **Home**: Redirects the user to the **UniFix Interface** (main page).
  - **UserGuide**: Provides a guide on how to use the **Student Desk**, ensuring a smooth user experience.
  - **Login**: Takes the user back to the **Login Page** for authentication.
  - **Exit**: Closes the program.

11.8.4.Student Desk

Once students have successfully logged in, their primary workspace is the Student Desk. Four main tabs comprise its organization:

1. **Profile**
2. **Inform Issue**
3. **View Issues**
4. **Complain**

11.8.4.1.Profile Tab

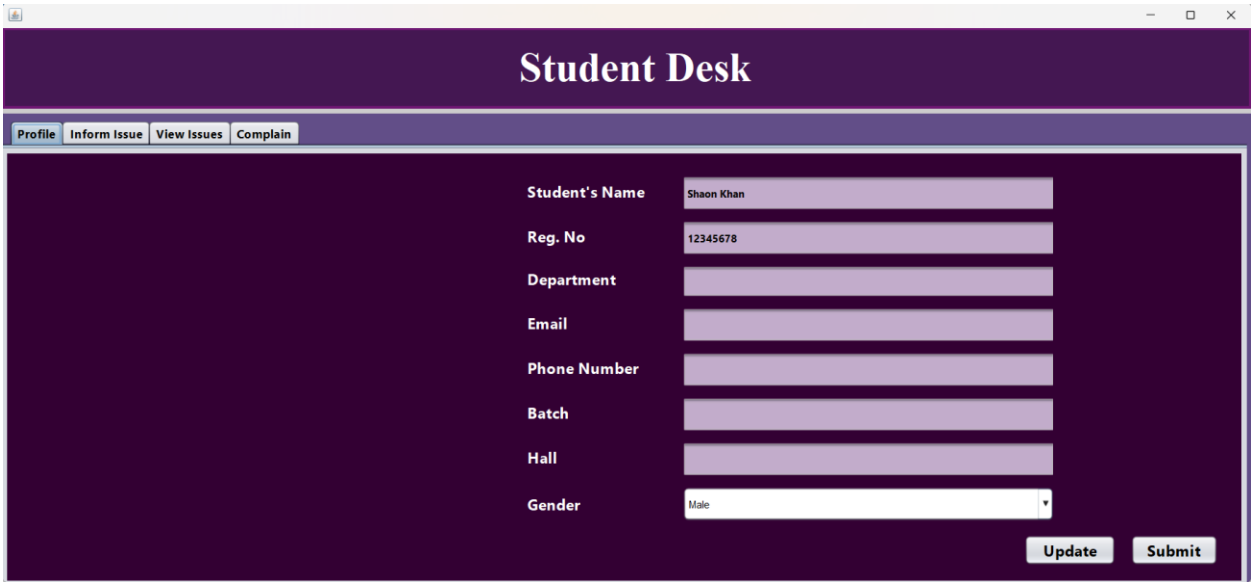

Fig-11.8.4.1.: Student Desk Profile Tab

This tab gathers and presents the student's personal data, such as:

Name, department, phone number, email address, batch, hall, gender, and registration number (Reg. No.)

Important features include:

• The information entered during the login procedure is automatically entered into the Name and Registration Number.

• If necessary, students can amend or fill in the remaining fields.

• The data is saved in the database after the Submit button is clicked.

• With the exception of the Registration Number, which cannot be changed, students can update their information at a later time if necessary.

11.8.4.2. Inform Issue Tab

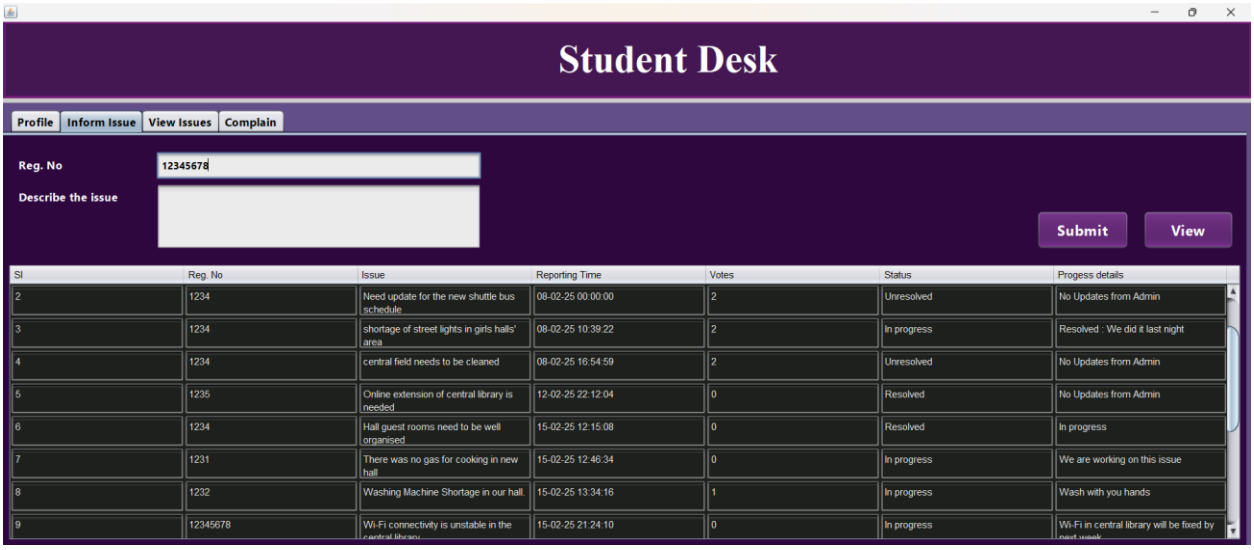

Fig- 11.8.4.2.: Student Desk Inform Issue Tab

Students can report problems with campus services or facilities using the Inform Issue Tab. The following functionalities are available on this tab:

• Text Fields:

Number of Registration (Reg. No.): The student's registration number from the login information is automatically entered into this area.

Describe the Problem: Students can use this text form to give a thorough explanation of the problem they are having on campus.

• **Submit Button:** The student clicks the Submit button once they have finished entering their information. The problem information is stored in the database for later analysis.

• **View Button:** Selecting this button brings up a table with information on the issues that have been reported. The student who reported the problem's registration number is included in the table.

An explanation of the problem the student reported.

o The report's time and date to record the issue's submission date.

o **Votes:** A measure of the number of other students experiencing the same problem, giving an indication of how common it is.

o **Status**: Indicates the issue's present state, such as "Under Review," "Resolved," etc.

o **Progress Details:** Gives details on how the problem is being resolved.

All of the problems that students have reported will be shown in this table, enabling users to monitor and comprehend the prevalent difficulties on campus.

11.8.4.3. View Issues Tab

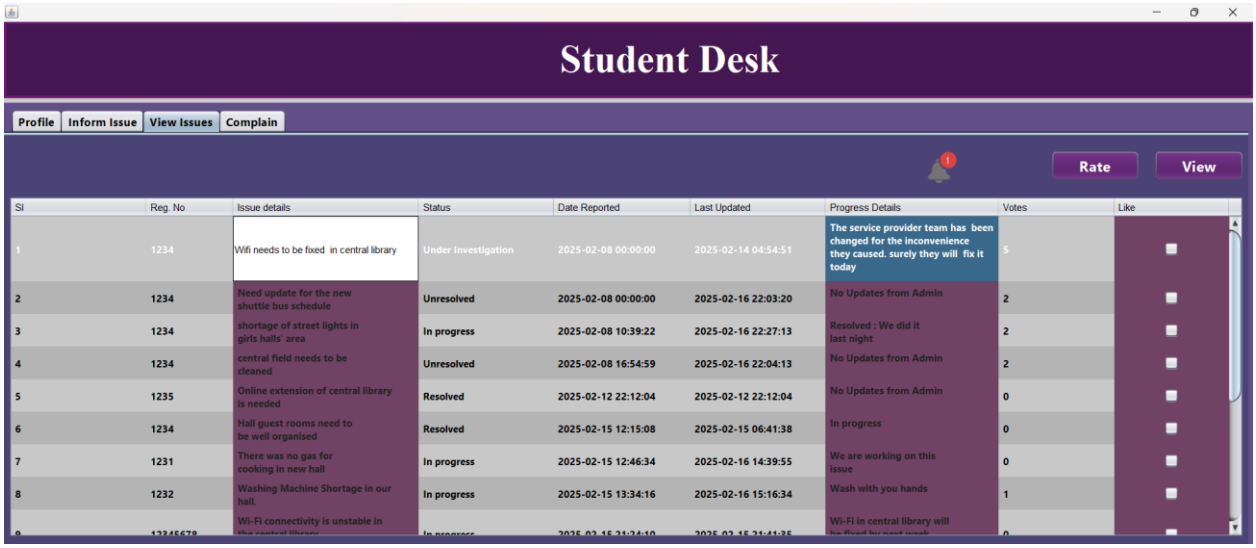

Fig-11.8.4.3.1.: Student Desk View Issues Tab

All of the issues that students have reported are summarized in the View Issues Tab, which also lets them interact with the issues' status and progress. The interface is set up as follows:

- **Issue Table**:

The following columns are shown in the table:

- **Serial Number**: A special number assigned to every reported problem.
- **Issue ID**: A unique ID given to every issue that is reported.
- **Registration Number (Reg. No):** The student's registration number who brought up the problem.
- **Issue Details**: A succinct explanation of the problem that has been reported.

- **Status:** The issue's present state, such as "Under Review," "Resolved," etc.
- **Date Reported:** The date on which the problem was initially brought to light.
- **Last Updated:** The most current information regarding the problem.
- **Progress Details:** The administrator provides frequent updates in this column that describe the state of issue resolution.

The option to vote:

For example, a voting option that allows pupils to cast their votes if the subject concerns them.

Total Votes: This column indicates how many students are dealing with a particular topic by displaying the total number of votes it has gotten.

Rate Button:

Using a scale of 1 to 5, students can assign a rating to the UniFix System.

The system will show the average rating of all users after they have submitted their ratings.

Students have the option to modify their rating at a later time to reflect their changing perceptions of the system.

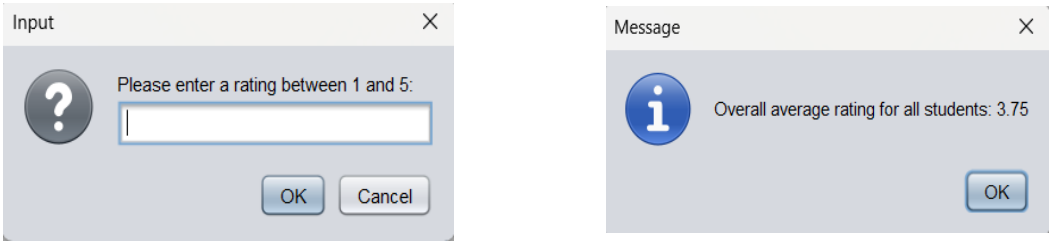

Fig-11.8.4.3.2.: Rating Process in View Issues

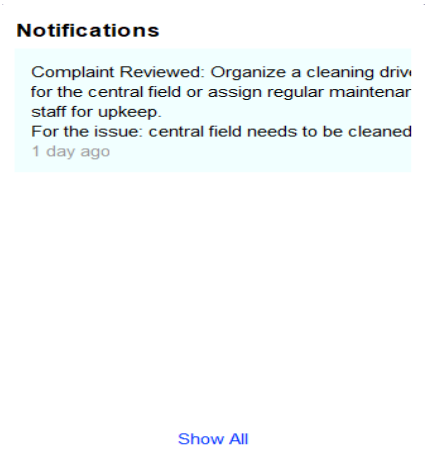

Fig-11.8.4.3.3.: Notification in View Issues

- **Notifications Bar:** This special feature displays notifications tailored to each student.
    - A notification stating the quantity of messages for a certain student will show up above the Notifications Bar once a new update is available.
    - Students only see their own notifications, not those for the entire class.
    - Students can view the administrator's message pertaining to a specific notification after clicking on it.

The purpose of this tab is to inform students about their problems and provide them with a means of interacting with the system by providing input and voting.

11.8.4.4. Complain Tab

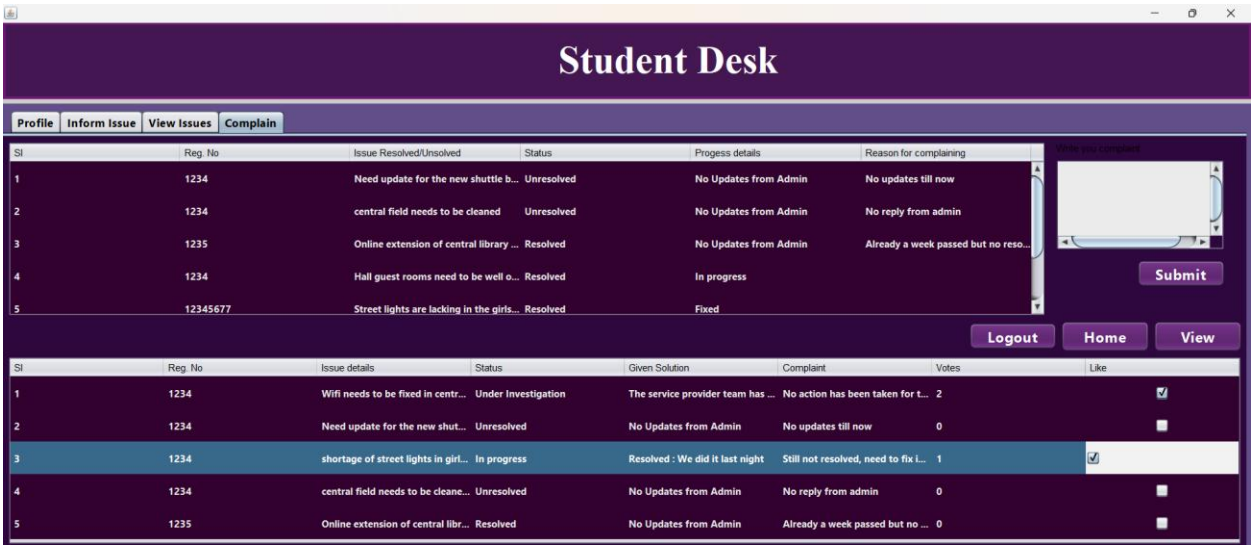

Fig-11.8.4.4.: Student Desk Complain Tab

Students can file concerns about outstanding difficulties or resolved issues that are not adequately addressed through the Complain Tab. The following features are included in the design of this tab:

- **Two JTables**
    - **Table of Issue Selection**
      This table shows two types of problems
      Issues that are still pending or being reviewed are referred to as unresolved issues.
      Issues listed as resolved may not satisfy the student, even though the issue has been resolved.
      From this table, students can choose the problem they want to voice their grievances about.
    - **Complaint Table:** The complaint information are added to this table after a student chooses a problem from the first table and files a complaint.
      Students can view and interact with existing complaints by viewing the Complaint Table, which shows all complaints that have been submitted.

- **Submit Button:**
    - The student can enter further information about their complaint after choosing an issue from the Issue Selection Table.

- The complaint will be added to the Complaint Table upon clicking the Submit button.

- **Voting Option:** Students can cast their votes for complaints in the Complaint Table that resonate with them or that they are facing.

    - By displaying the number of students impacted by the same issue, this voting system aids in prioritizing complaints.

This page promotes a more responsive and transparent system by giving students a forum to voice their displeasure or carry on the conversation about unresolved concerns.

11.8.5. Admin Login Interface

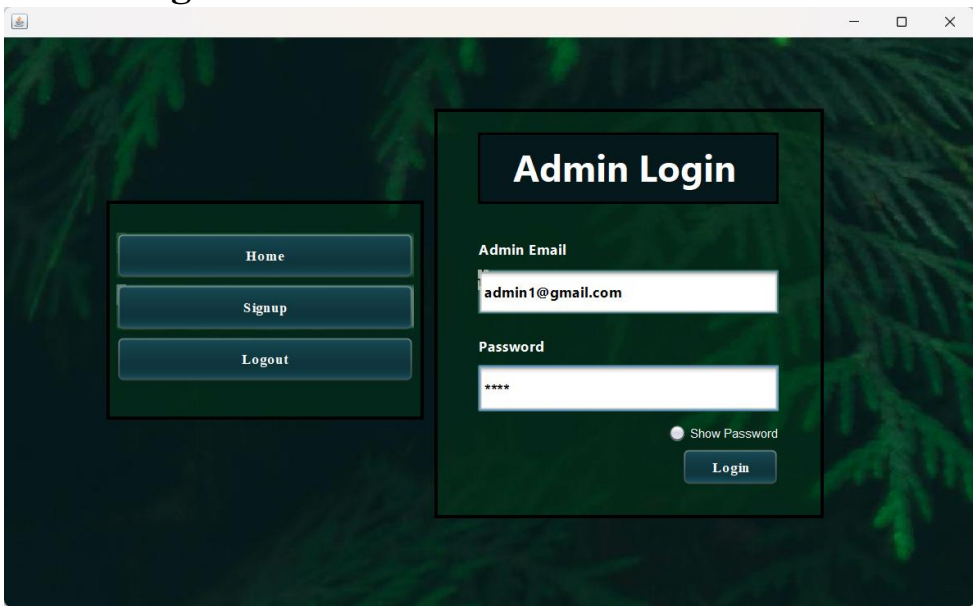

Fig-11.8.5.: Admin Login Interface

The user is taken to the Admin Login Interface after selecting the Admin Button on the UniFix (Main Page). The administrator can sign up or log in here to access the Admin Desk.

- The login form
    - Admin Email: This is the email address that the administrator has registered.
    - Password: For authentication, the administrator supplies the appropriate password.
    - Access to the Admin Desk is granted by clicking the Login Button after entering the credentials.

- Signup Option:
    - The administrator must first click the Signup Button on the login page to create an account if they don't already have one.
    - The admin can register a new account on the Admin Signup Page after this redirection.

- There are other buttons:
    - Home: Returns to the Main Page of the UniFix.
    - Logout: Allows the administrator to safely quit the system by logging them out and rerouting them to the Admin Login Interface.

By permitting only authorized people to use the Admin Desk, this interface guarantees safe access while upholding system integrity.

11.8.6. Admin Signup Interface

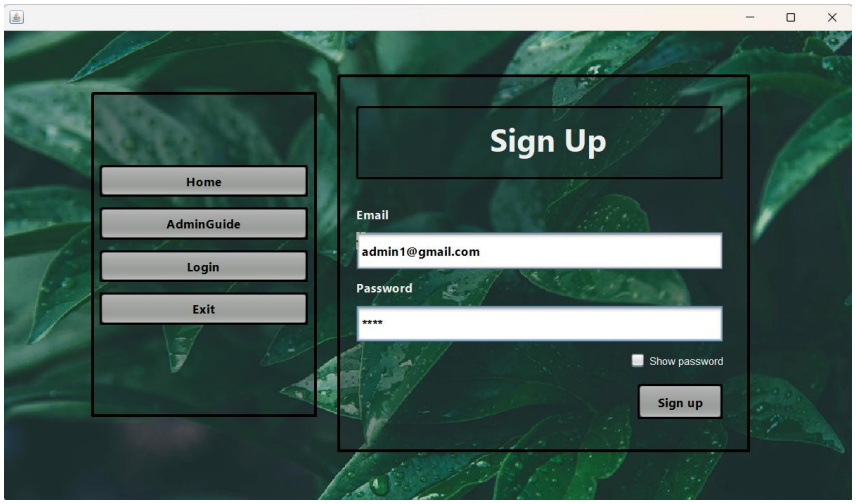

Fig-11.8.6.: Admin Sign Up Interface

The administrator must first create an account using the Admin Signup Interface if they don't already have one. The structure of this interface is as follows:

- Registration Form:
    - Email:The administrator supplies a working email address, which serves as the login username.
    - Password: The administrator creates a strong password for upcoming logins.
    - The admin account is successfully created when clicking the Signup Button after the necessary information has been entered.

- The administrator is taken to the Admin Login Interface after successfully completing the signup process, where they can use their newly generated login credentials to log in.

- There are other buttons:
  - **Home:** Takes you to the UniFix (Main Page).
  - **AdminGuide**: Offers instructions for using the Admin Desk, including how to view student complaints, manage issues, and update progress.
  - **Login:** For current administrators who already have an account, this takes them to the Admin Login Interface.
  - **Exit:** Safely closes the application.

Only authorized staff may access the Admin Desk thanks to this interface, which guarantees a simple and safe account setup process.

11.8.7.Admin Desk

The administrator's primary control panel, the Admin Desk, gives them access to manage student data, change issue statuses, and respond to complaints. Three tabs are used to organize it:

1. Student Data

2. Problem with the Update

3. Complaint

11.8.7.1.Student Tab Info

Fig - 11.8.7.1.: Student Info in Admin Desk

The administrator can see comprehensive details on students who have complained or reported problems in the Student Info Tab.

- Information Shown:
  - Name of the student
  - Registration number
  - Department
  - Phone number
  - Email
  - Batch
  - Hall
  - Gender
- View Button:
  - A table containing all of the student data is shown after selecting the View Button.
  - This gives the administrator access to every student's information who has engaged with the system.
- Search Functionality:
  - By entering the registration number in the search area, the administrator can look up the details of a particular student.
  - The table is filtered by this feature to display the student's information that corresponds to their registration number.

For quick problem solving and communication, this tab makes sure the administrator can effectively access and handle student information.

11.8.7.2.Update Issue

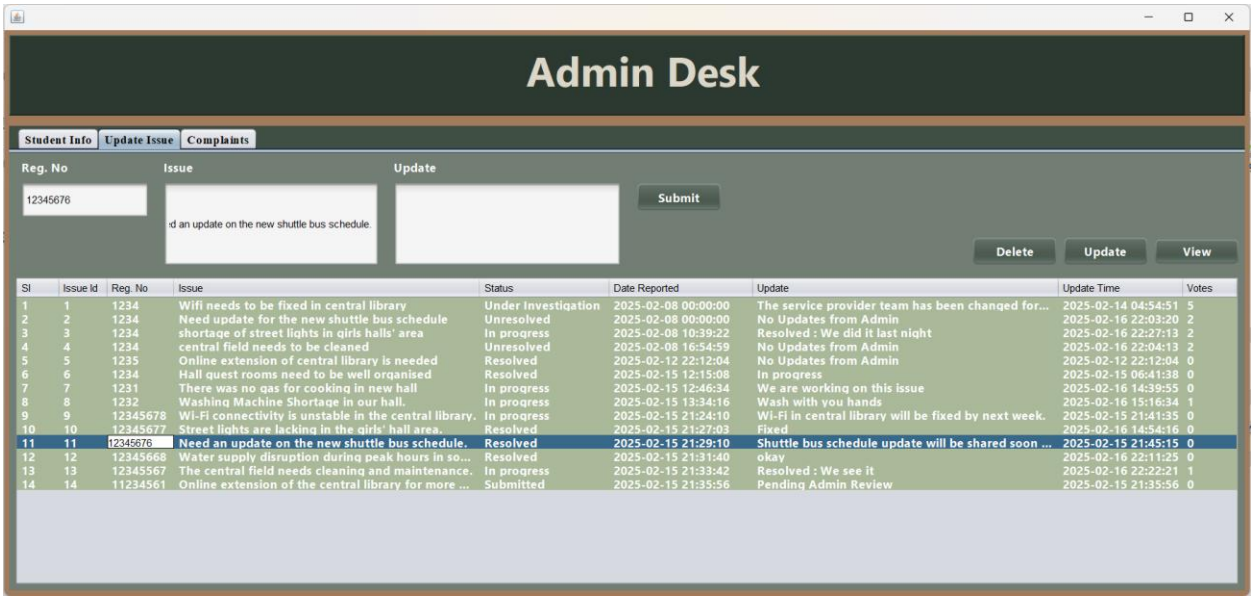

Fig -11.8.7.2.: Update Issue(Admin Desk)

Administrators can effectively examine, update, and monitor the status of issues reported by students with the use of this tab.

All reported issues are displayed using a JTable, which includes crucial information like:

Registration Number: This number identifies the student who filed the problem.

Issue Description: An overview of the problem that has been reported.

The status of the problem at the moment (e.g., "In Progress," "Resolved").

Progress Reports: Any information provided about the issue's resolution.

Date and time of the most recent modification is known as the "Last Modified Date."

Using a JTextField, administrators can choose a particular issue, provide a resolution update, and change the issue's status as necessary. Following the submission of an update via the Submit Button, the following things take place:

The new status and progress information has been added to the database.

To reflect the most recent modifications, the JTable dynamically refreshes.

To let the student know the latest information regarding their reported problem, a notification is sent.

11.8.7.3.Complain Tab(Admin Desk)

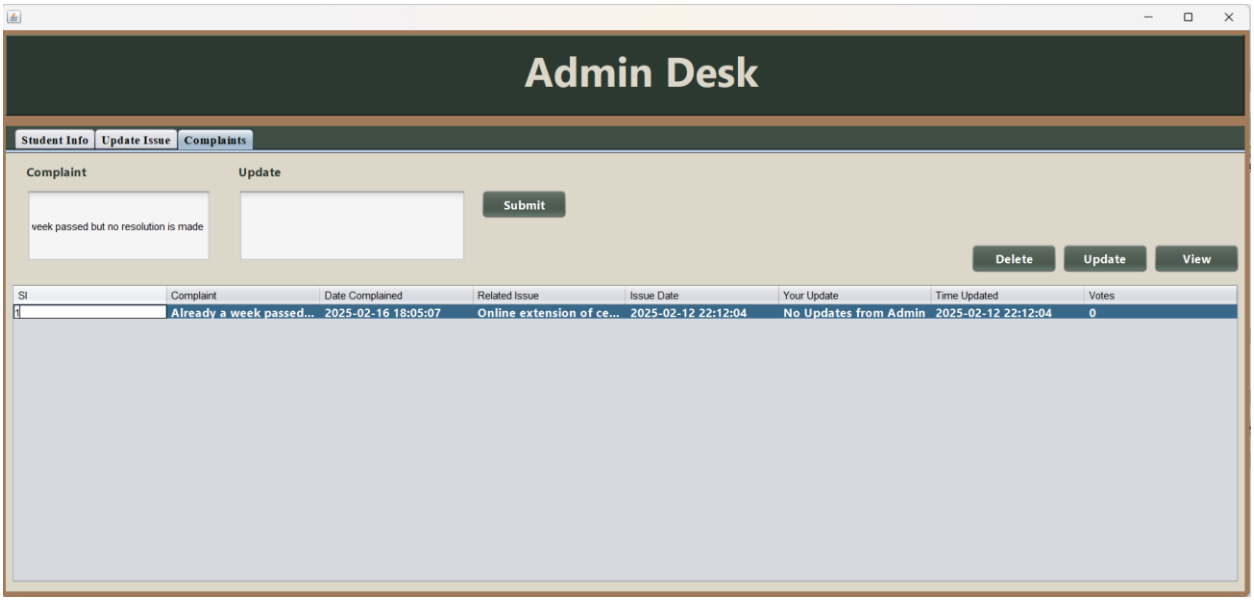

Fig-11.8.7.3.: Complaints (Admin Desk)

Student complaints regarding unresolved or delayed issues are handled under this category.

JTable: Shows complaint statuses, issue descriptions, and student information. Administrators can enter answers for specific complaints using JTextField. The "Submit" button links responses to the relevant issue reports, saves them, and updates complaint statuses.

In order to maintain open communication and administrative accountability, the system alerts the student and updates the status of the associated issue when a complaint is resolved.

11.8.8.Report

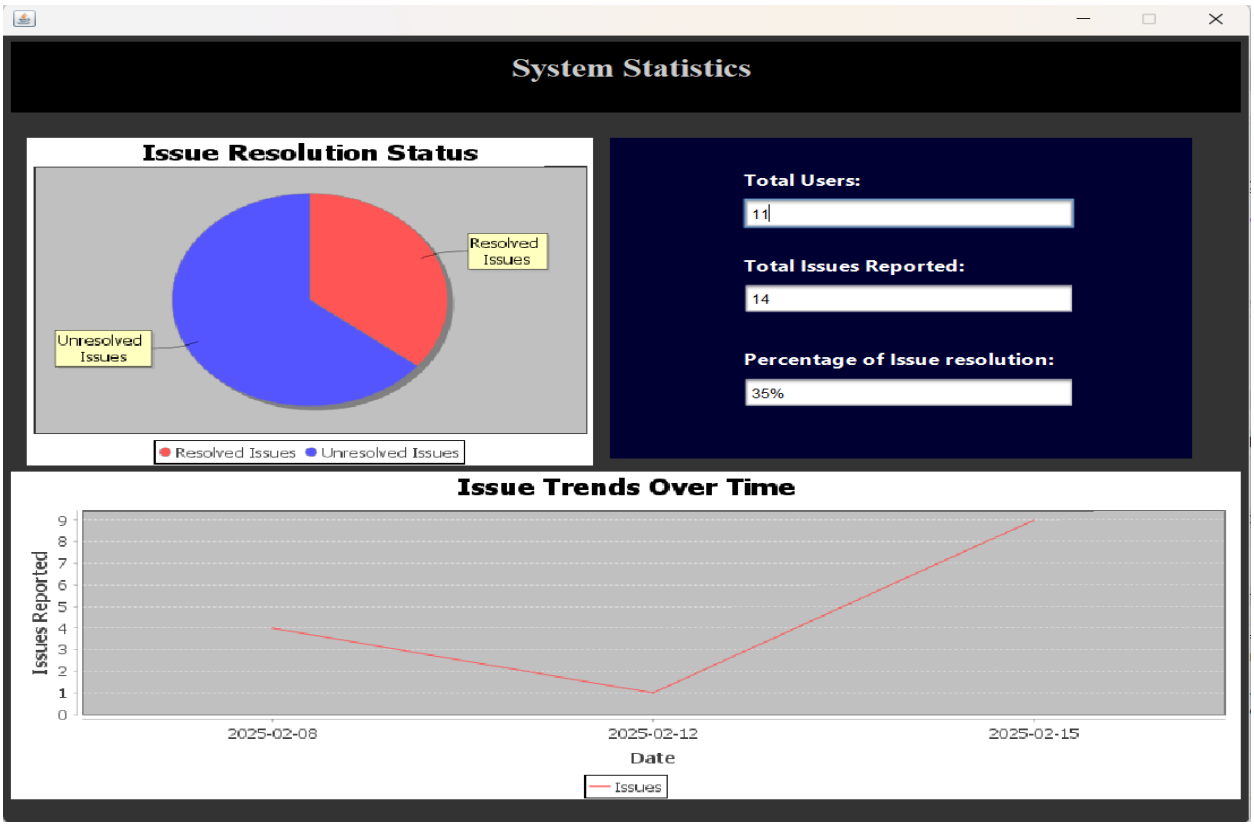

Fig-11.8.8.: Report

Both students and administrators can more easily analyze issue resolution trends thanks to the View Report tool, which offers a thorough, real-time statistical picture of the system's effectiveness. This feature, which can be accessed via the "View Report" button on the Home Page, improves the system's operating effectiveness and transparency.

There are three main sections in the report:

1. **Pie Chart**: Issues Solved vs. Unsolved

The percentage distribution of issues that have been resolved and those that have not is shown in this section. By acting as a key performance indicator, it allows stakeholders to assess how well the system handles student problems. The pie chart provides useful information about overall system responsiveness by graphically differentiating between issues that have been resolved and those that are still pending.

2. System Summary Metrics, or Counters

A brief summary of important system metrics is given in this section, including:

Total Registered Users: The total number of users who have signed up for the system.

The total number of concerns that users have reported is known as the "total issues reported."

Students Reporting Problems: The total number of distinct students who have brought up problems.

Issue Resolution Rate: The proportion of problems that are fixed in a given amount of time, such three, five, seven, or more days. Administrators can evaluate and improve response efficiency with this statistic.

3. Issue Trends Over Time on a Line Graph

The trends of issues reported over specified time periods (e.g., weekly or monthly) are depicted in this section. This line graph helps identify peak reporting periods by showing the volume of issues reported over time, which makes resource allocation and strategic planning easier.

12. Future Work

To further improve the capabilities of **UniFix**, several key enhancements are being considered for future development. A dedicated mobile application is planned to increase system accessibility, allowing users to interact with the platform from anywhere. Additionally, integrating **AI-driven issue classification** can improve the efficiency of issue categorization and prioritization, reducing administrative workload. Implementing **automated reminders** for unresolved issues will ensure prompt resolution and enhance the system's responsiveness. These planned improvements will aim to make **UniFix** even more efficient, user-friendly, and adaptable to the evolving needs of the campus community, fostering continuous refinement and optimization of the platform.

13. Conclusion

UniFix provides an efficient, structured solution for addressing campus-related issues, facilitating smooth communication between students and administrators. The system's core features—issue reporting, voting, and complaint submission—ensure transparency and accountability within the campus environment. By enabling students to prioritize critical concerns and allowing administrators to streamline workflows, **UniFix** enhances the overall issue resolution process. This framework not only improves the speed and efficiency of administrative responses but also creates an interactive space for students to actively contribute to the betterment of campus life. The system’s ability to track and resolve issues effectively contributes to a more accountable and responsive campus environment.

14.References

[1] Database Management in Java ; https://www.geeksforgeeks.org/sql-database-connection-in-java/ Accessed: 12 January 2025.

[2] Communication Protocols in Java ; https://www.baeldung.com/java/socket-communication. Accessed: 05 December 2024.

[3] Creating and Handling Interfaces in Java ; https://www.journaldev.com/1588/java-interface-example. Accessed: 03 February 2025.

[4] Pie Chart Implementation in Java ; https://www.javacodegeeks.com/2017/09/how-to-create-pie-chart-in-java.html. Accessed: 28 November 2024.

[5] Building a Voting System in Java ; https://www.geeksforgeeks.org/java-voting-system-project/ Accessed: 10 February 2025.

[6] Notification System Implementation in Java ;https://www.geeksforgeeks.org/how-to-send-notifications-in-android-using-java/ Accessed: 19 December 2024.

[7] Working with Date and Time in Java ; https://docs.oracle.com/javase/tutorial/datetime/ Accessed: 25 January 2025.

[8] Campus Authority and Communication System ; https://www.sciencedirect.com/science /arti cle/pii/S2352914819307399. Accessed: 08 February 2025.

[9] Campus Authority and Communication System ; https://www.researchgate.net/ publication /335497370_A_Smart_Campus_Communication_System. Accessed: 30 December 2024.

[10] Solving Campus-related Problems with Web and Java Technologies; https://www.researchgate.net/publication/344710836_Smart_Campus_Management_System_Using_Android_and_Cloud_Technology. Accessed: 07 January 2025

[11] Badge Notification and GlassPane-Popup; https://github.com/DJRaven/glasspane-popup. Accessed: 10 February 2025.