**Chapter 1**

**INTRODUCTION**

**Project Context**

The increasing reliance on technology in educational institutions has led to a growing need for efficient systems to manage various aspects of administrative tasks. One crucial aspect is the tracking of attendance in computer laboratories, where students engage in practical sessions to enhance their skills. The traditional methods of manual attendance tracking are prone to errors, time-consuming, and lack the necessary precision required in today's fast-paced educational environments.

The Attendance Management System (AMS) emerges as a solution to address the shortcomings of conventional attendance monitoring methods. This system aims to streamline the process of recording and managing student attendance in computer laboratories, providing an automated and accurate approach. By leveraging technology, AMS seeks to enhance the overall efficiency of attendance tracking, ensuring that academic institutions can focus more on delivering quality education and less on administrative hassles.

The scope of AMS extends beyond mere attendance recording; it encompasses features such as real-time data updates, user-friendly interfaces, and robust reporting capabilities. This comprehensive approach ensures that educators, administrators, and students can benefit from a system that not only meets the immediate need for accurate attendance tracking but also provides valuable insights into attendance patterns and trends over time.

In conclusion, the Attendance Management System presents a timely and essential solution for educational institutions grappling with the challenges of manual attendance tracking. As we delve into the subsequent chapters, we will explore the system's architecture, functionality, and implementation strategies to gain a deeper understanding of how AMS can revolutionize the way computer laboratory attendance is managed.

**Theoretical Background**

The development and implementation of a Attendance Management System (AMS) align with the concept of efficient resource management and monitoring in educational institutions. Research in this area emphasizes the need for systems to record attendance, respond to absence, and ensure efficient usage of computer resources.

For instance, a study on "Access Control and Monitoring: A System for Computer Laboratory" highlights the importance of monitoring and controlling computer resources to ensure secure and efficient usage by students (Zapanta et al., 2021). Additionally, the "College Smart Classroom Attendance Management System Based on Internet of Things" provides technical support for the development of technology and the reform of educational management, emphasizing the continuous innovation and adjustment of education management mechanisms to provide more convenient and reliable information services for teachers and students (Zhao et al., 2022).

These theories support the thesis of a AMS by emphasizing the importance of effective management and monitoring of computer laboratories to enhance the learning environment and ensure optimal resource utilization.

**Statement of the Problem**

The goal of this project is to establish the Attendance Management System (AMS) for streamlined attendance management. Specifically, this study aims to address the following question:

1. What are potential problem and issue of current manual tracking of attendance in terms of:
   1. Time Inefficiency
   2. Error-Prone Recording
   3. Lack of Device Tracking Capabilities
   4. Instances of Lost and Stolen Devices
   5. Misplacement of Paper Records
   6. Inefficient Response to Lost or Stolen Devices
   7. Security Risks
   8. Privacy Concern
2. Do you believe that the suggested AMS (Attendance Management System) would be valuable in enhancing the attendance tracking system?
3. Have you ever accused that you are the one damage, lost and being theft because the people damage the item without any tracking devices that’s why important to track people?’

**Purpose and Description of the Project**

The primary purpose of the " Attendance Management System" project at Talisay City College is to usher in a transformative era in the management of computer laboratories. This initiative seeks to modernize and optimize the current attendance tracking processes, addressing persistent challenges such as the vulnerability of devices to theft or damage and the inherent inefficiencies of a manual, paper-based system. The primary goal is to institute a comprehensive digital solution that not only mitigates security concerns but also enhances administrative efficiency. This project envisions providing indisputable evidence in cases of device-related issues, thereby ensuring a secure and reliable method of tracking all users within the computer laboratories.

The " Attendance Management System" project is poised to bring about a paradigm shift in how Talisay City College oversees and manages its computer laboratories. Instead of relying on traditional manual methods, the project emphasizes the implementation of key features and components to streamline operations. These include meticulous user tracking, where the system will monitor and record entry and exit times of all laboratory users, creating a comprehensive and accurate attendance log. Moreover, the project introduces a centralized device management system, facilitating swift identification of stolen or damaged equipment and enabling timely documentation of incidents.

A pivotal aspect of the project is the transition to a digital database for storing attendance records, user information, and device status. This move away from a paper-based system reduces the risk of data loss or misplacement and ensures that information is readily accessible only to authorized personnel. Additionally, the project incorporates an automated reporting system, generating insightful reports on attendance patterns, device usage, and irregularities. This data-driven approach empowers administrators to make informed decisions in laboratory management.

Furthermore, an alert system is integrated to notify designated administrators promptly in cases of suspicious activities or deviations from normal usage patterns. This proactive approach ensures timely intervention and resolution of potential issues, contributing to an environment of heightened security and accountability. The digitalized system significantly enhances accountability by accurately recording each user's attendance and activities, providing crucial evidence in cases of disputes, lost devices, or damages.

In conclusion, the " Attendance Management System" project signifies a significant leap forward for Talisay City College, embracing technological advancements to enhance administrative excellence. By implementing this comprehensive system, the college not only addresses security concerns associated with laboratory devices but also streamlines administrative processes for a more efficient and accountable environment. This initiative underscores the institution's commitment to providing a secure and accessible educational experience, marking a pivotal milestone in the integration of technology for the betterment of the academic community at Talisay City College.

**OBJECTIVES OF THE PROJECT**

**General Objective**

The overarching goal of the Attendance Management System (AMS) is to modernize and optimize the process of attendance tracking in educational institutions, specifically within computer laboratories. The general objective is to introduce an automated and efficient system that enhances the overall accuracy, security, and responsiveness of attendance management.

**Specific Objective**

The specific objectives of the Attendance Management System (AMS) are: 1) Develop precise and real-time automated tracking for computer laboratories, 2) Implement features for instant and accurate attendance updates, 3) Enhance security measures to safeguard attendance records, and 4) Prioritize a user-friendly interface for users.

**SCOPE AND LIMITATIONS OF THE PROJECT**

**Scope**

The scope of the Attendance Management System (AMS) encompasses the development and implementation of an automated attendance tracking solution tailored for computer laboratories in educational institutions. AMS aims to streamline attendance management, providing real-time data updates, and analytics features. The system's user-friendly interface is designed for educators, administrators, and students, ensuring ease of use. The project's scope extends to enhancing data security through advanced measures and promoting widespread adoption within existing educational workflows.

**Limitations**

        The researchers expected a list of limitations along the way considering the scope stated above. The limitations of the project outcome cover the following:

1. AMS relies on the availability of reliable network connectivity for real-time tracking.
2. The system does not address hardware-related issues of the devices, focusing solely on tracking their status and location.

**Significance of the Study**

          The aim is to develop Attendance Management System for the user in Talisay City College Laboratory Room 1, 2, and 3 for Faculty, Teachers, and students. Moreover, it will be very advantageous to the following:

**Teachers:** The study is significant for teachers as it streamlines the attendance tracking process, allowing them to focus more on delivering educational content rather than spending valuable time on manual record-keeping. It provides teachers with accurate and readily available attendance data, aiding in the assessment of student participation and engagement.

**Faculty Staff:** Faculty staff benefit from the study by experiencing improved efficiency in administrative tasks related to attendance management. The automated system reduces the burden of manual data entry, enabling faculty staff to allocate their time and efforts more effectively in supporting the educational environment.

**Administrators:** Administrators find significance in the study as it enhances overall management capabilities. The system's features, including device tracking and improved security measures, contribute to a more secure and organized computer laboratory environment. This aids administrators in ensuring the proper functioning of the facilities and responding promptly to security incidents.

**Students:** For students, the study brings about a more streamlined and transparent attendance tracking system. This reduces the likelihood of errors in attendance records and provides students with a fair and accurate representation of their attendance, which is crucial for academic progress and accountability.

**Researchers:** Researchers benefit from the study's focus on digitalizing attendance tracking, as it provides a foundation for future research endeavors. The data generated by the system can be utilized for academic studies on educational technology, administrative efficiency, and security measures in academic environments.

**Future Researchers:** The study is significant for future researchers as it sets a precedent for the integration of technological solutions in educational settings. The findings and insights from this study can guide future researchers in exploring and enhancing attendance tracking systems, contributing to the continuous improvement of educational technologies.

Certainly, let's include the information about the cost associated with the internet load in your methodology.

**METHODOLOGY**

## Technical Feasibility

The Attendance Management System (AMS) aims to streamline and enhance the efficiency of monitoring student attendance in a computer laboratory setting. This methodology outlines the technical feasibility, technical expertise, integration with existing systems, data migration and conversion, scalability, security considerations, reliability and performance, the cost of technology, and the investment in internet load associated with the development and implementation of AMS.

1. System
   1. System Requirements
      1. Define functional and non-functional specifications for AMS.
      2. Computer Hardware must have Standing screen display size ‎23.8 Inches, Screen Resolution ‎1920 x 1080, Processor ci3, RAM 8 GB sdram, Hard Drive ‎500 HDD, Graphics Coprocessor ‎Intel UHD Graphics 630, Hardware Platform ‎PC, Operating System ‎Windows 10 UP+ , Processor Brand ‎Intel, Computer Memory Type ‎DDR4 SDRAM, Hard Drive Interface ,Serial ATA-150,
      3. Utilize Java Netbeans (IDE), JDK 17, XAMPP, MySQL Database, and MySQL Connector for compatibility and efficient fulfillment of specifications.
      4. Java Netbeans for user interface and system logic.
      5. MySQL Database for secure and organized storage of attendance data.
   2. Technical Expertise:
      1. Leverage Java Netbeans' widespread usage and compatibility with JDK 17 for easy recruitment or training of developers.
      2. Utilize common expertise in MySQL Database and MySQL Connector for readily available talent.
2. Alignment with Current System
   1. Java Netbeans Integration
      1. Utilize the IDE's compatibility with JDK 17 for seamless integration with various technologies.
      2. Ensure adaptability to existing infrastructure, including XAMPP.
   2. MySQL Database Integration:
      1. Guarantee compatibility with XAMPP for smooth integration with other databases and systems.
3. Data Transfer and Transformation
   1. MySQL Database Versatility:
      1. Efficiently use XAMPP for migration and conversion of existing data.
      2. Ensure data integrity and optimal system performance during the transition.
      3. Exploit Java Netbeans support for data migration processes.
4. Scalability
   1. Technology Scalability:
      1. Java Netbeans, JDK 17, XAMPP, MySQL Database, and MySQL Connector combination for scalability.
      2. Accommodate potential growth in attendance data and user loads.
      3. Allow easy adjustments to handle increased data volumes.
5. Security Considerations
   1. Java Netbeans Security Features:
      1. Implement robust security features to ensure confidentiality and integrity of attendance data.
   2. MySQL Database Security Measures:
      1. Utilize advanced security measures in XAMPP, MySQL Database, and MySQL Connector, including user access controls and encryption, to protect sensitive information.
6. Reliability and Performance
   1. Java Netbeans Reliability:
      1. Facilitate the development of reliable and high-performance applications.
7. MySQL Database Performance:
   * 1. Optimize XAMPP and MySQL Database performance to contribute to overall system reliability and efficiency.
8. Cost of Technology
   1. Open-Source Advantage:
      1. Utilize Java Netbeans as an open-source IDE to reduce development tool costs.
   2. Cost-Effective Database Management:
      1. Exploit the cost-effectiveness of XAMPP, MySQL Database, and MySQL Connector compared to other systems.
   3. Internet Load Investment:
      1. Allocate 800 pesos for internet load to facilitate development activities in the absence of internet access.
9. Basic System Specifications
   1. User Registration and Login Functionality:
      1. Implement secure user registration and login features using Java Netbeans and JDK 17.
   2. Data Management:
      1. Efficiently handle the collection, storage, organization, and utilization of attendance data in MySQL Database.
   3. Automated Eligibility Checks and Verification Processes:
      1. Implement automated processes to check and verify attendance eligibility using Java Netbeans and MySQL Connector.
10. User Interface and Experience
    1. Seamless User Experience
       1. Ensure a seamless and user-friendly experience for administrators and applicants using Java Netbeans for UI design and JDK 17 for enhanced features.
       2. Improve user satisfaction and streamline the attendance tracking process.

The **graphical user interface (GUI)** design is integral to user interaction. The capabilities of Java Swing in creating a user-friendly GUI are assessed, and the suitability of its event handling mechanisms for capturing user interactions is confirmed.

**Security measures** play a pivotal role in system integrity. The effectiveness of the login authentication mechanism is evaluated to ensure secure access. Additionally, considerations for implementing encryption methods to safeguard sensitive data, particularly during transmission, are examined.

**Database operations**, including insertion and retrieval of attendance data and hardware/software information, are validated. Proper transaction handling mechanisms are essential to maintain data consistency. Error handling measures, such as exception handling and the implementation of logging for tracking errors and debugging, are also addressed.

**User experience** is a critical aspect, focusing on the intuitiveness of the interface for ease of use and clarity in navigation. The responsiveness of the application to user inputs is tested to ensure a smooth user experience.

**Testing procedures** encompass unit testing for individual components, integration testing to verify the seamless integration of different modules, and user acceptance testing to gauge the system's alignment with user needs.

By exploiting the combined capabilities of Java Netbeans, JDK 17, XAMPP, MySQL Database, and MySQL Connector, the technical feasibility of the AMS is addressed comprehensively, considering key aspects such as hardware, software, expertise, integration, alignment, scalability, security, reliability, cost-effectiveness, and the investment in internet load.

**Operational Feasibility**

**Flow Chart**

Figure 1 outlines the key processes of the Attendance Management System. Focused on optimizing efficiency, the system begins with user-friendly login and authentication. Account management allows easy registration and password reset. Post-login, the intuitive interface enables seamless attendance input. The subsequent stages ensure effective data handling, clear presentation, and user-friendly management options, emphasizing operational feasibility through streamlined processes and user acceptance.

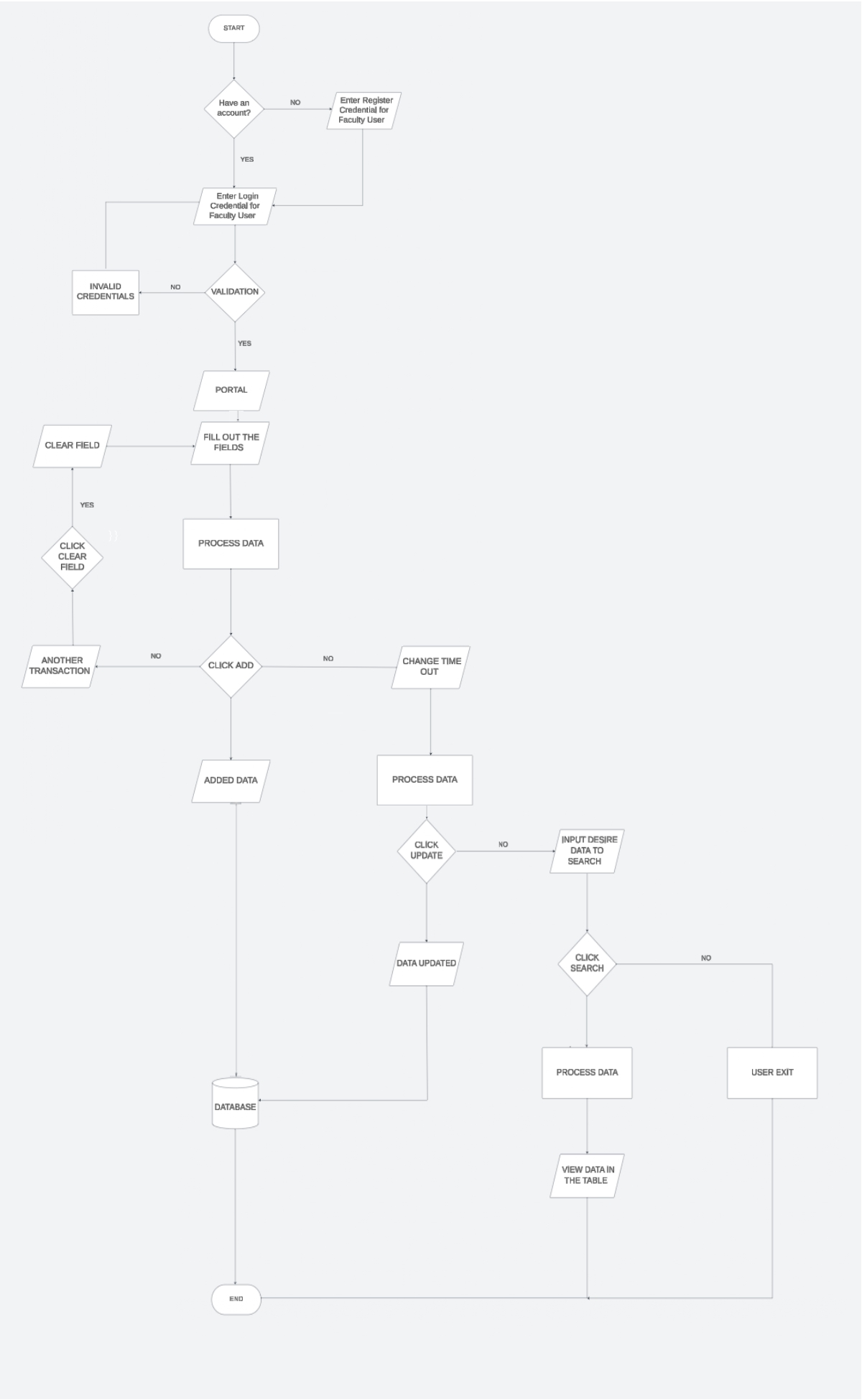


Figure 1

**Flow chart**

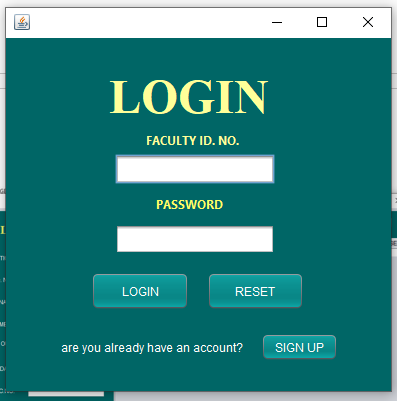


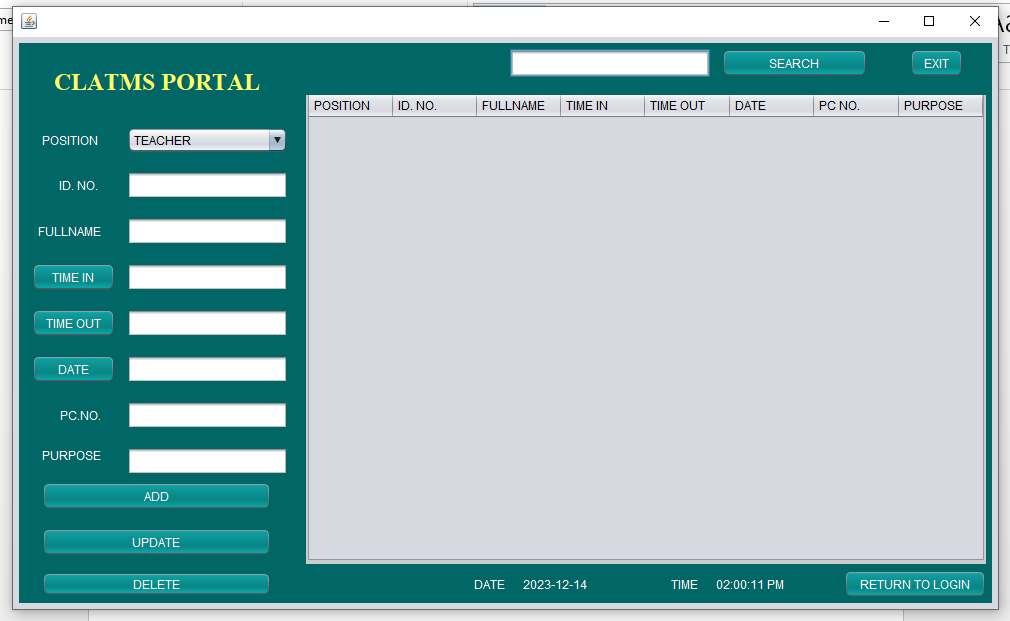
REGISTRATION



FACULTY







STORED IN DATABASED

FACULTY MAY LOGIN

AMS PORTAL

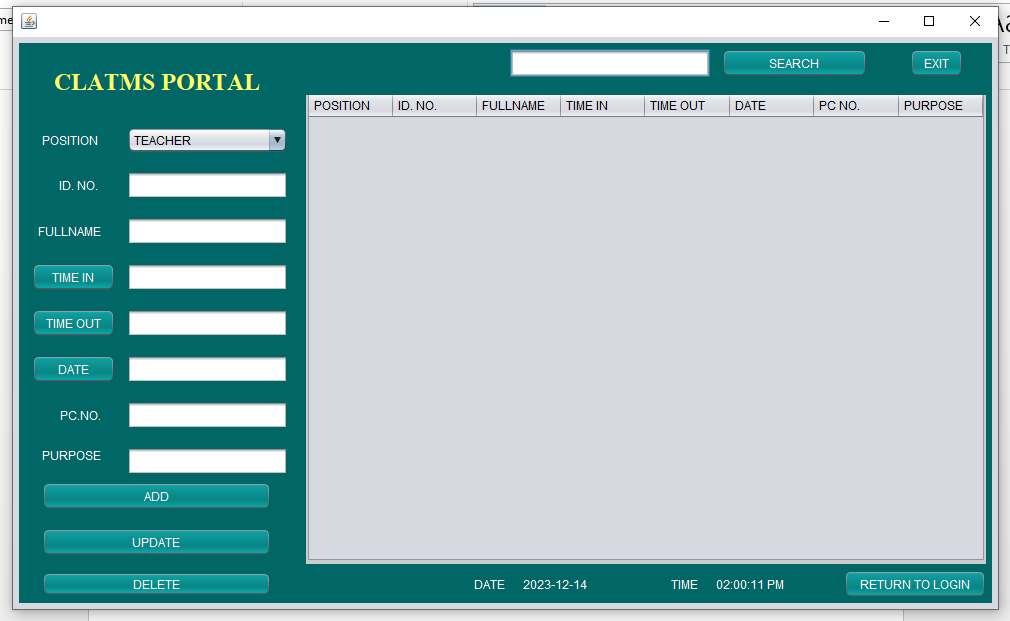
Figure 2

Faculty Registration Case Use System



FACULTY WILL LOGIN AND INPUT ALL GATHER DATA

GET USER INFORMATION





IT WILL APPEARS IN THE TABLE’S PORTAL

DATA WILL STORED IN THE XAMPP server and MySQL DATABASED

Figure 2

Login, Input User Information Use Case System

**Economic Feasibility**

The **Cost/Benefit Analysis** serves as a crucial tool for assessing the economic feasibility of the proposed system. It aims to determine the viability of investing in the software system and provides insights into the potential costs associated with its development. The analysis is segregated into two distinct tables, namely Development Cost and Operational Cost.

**Development Cost**: Encompassing expenses incurred throughout the system's developmental phase, the Development Cost table outlines the financial investments dedicated to the creation of the proposed system.

**Operational Cost**: Focused on expenditures post-implementation, the Operational Cost table sheds light on the ongoing financial commitments required to maintain and operate the system effectively.

The analysis considers various factors and resources, including the utilization of a computer equipped with Windows 10, the latest version of JDK, NetBeans IDE, and XAMPP MySQL for database management. Additionally, **Table 3** show the cost like, limited internet access is acknowledged, with a cost-effective approach of loading every three days for 50 pesos. The total cost estimation is calculated based on this budget-conscious strategy, extending for a month with a cumulative cost of 50 pesos multiplied by 30.

This economic feasibility assessment is essential for stakeholders to make informed decisions regarding the allocation of resources, ensuring that the benefits derived from the system outweigh the associated costs, both during development and in the operational phase.

**Table 1**

**List of Equipments used during System Development**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cost Analysis** | | | | |
| **Category** | **Description** | **Unit Price** | **Price (php)** | **Qnty** |
| Hardware Devices | Computer Set (laptop) | 15,000 | 15,000 | 1 |
| Software Tool | Cost of Windows 10, JDK, NetBeans(IDE),XAMPP MySQL | 0 | 0 | 1(each) |
| Internet Access | Load 50 for 3 days (60 days) | 3,000 | 3,000 | 1(each/3days) |
| Maintanance and Support | Food, Transportation, and other Expenses | 2,000 | 2,000 | Good for 5 Persons |
| Total Cost |  |  | 20,000 |  |

**Schedule Feasibility**

The schedule feasibility delineates the activities conducted throughout the project's implementation, encompassing the project's start and end dates. Figure 3 visually represents the activities undertaken during the analysis, planning, design, implementation, and maintenance phases of the project.

Planning (August): Commencing its journey in August, the project initiated the planning phase. This pivotal stage lays the foundation for the entire project, concentrating on articulating the project's scope, objectives, and requirements. It stands as a fundamental step in steering a prosperous and well-organized software development process.

Analysis (September): As the calendar turns to September, the project seamlessly transitions into the analysis phase. This phase represents a noteworthy milestone in the development life cycle, engaging in a meticulous examination of user requirements and system specifications. The principal objective is to ensure the proposed system harmoniously aligns with the needs and expectations of its end users.

Design (September to October): Advancing into September and October, the project enters the design phase. This critical juncture involves crafting system architecture, database design, and careful planning of user interfaces. The design phase plays a crucial role in sketching the comprehensive blueprint that will direct subsequent development and implementation processes.

Implementation (November to December): With the groundwork laid in prior phases, November signals the commencement of the implementation phase. Extending through December, this period is exclusively dedicated to the tangible coding and development of the system. The implementation phase serves as the transformative stage where the theoretical design metamorphoses into a functional system, inching the project closer to fruition.

Maintenance (November to December): Concurrently aligned with the implementation phase, the project immerses itself in maintenance, initiating in November. Maintenance, a continuous process, underscores perpetual support, bug rectification, and system optimization. The simultaneous execution with the implementation manifests a proactive stance, ensuring prompt issue resolution and bolstering the overall stability and reliability of the system.

|  |  |  |
| --- | --- | --- |
| S C H E D U L E | AUGUST | PLANNING |
| SEPTEMBER | DOCUMENTATION |
| OCTOBER | SYSTEM DESIGNING |
| NOVEMBER | DOCUMENATION AND SYSTEM DESIGNING |
| DECEMBER | PRESENTATION |

**Figure 3**

**Schedule Feasibility**

**DEFINITION OF TERMS**

**Computer Laboratory**

A dedicated space equipped with computers and related resources for educational or research purposes. In the context of your project, it refers to a specific facility where students or individuals use computers.

**Attendance Management System**

A software application designed to automate the process of recording and monitoring attendance. It typically includes features for data input, storage, retrieval, and analysis related to attendance records.

**Java Netbeans IDE**

NetBeans Integrated Development Environment (IDE) is a software application that provides comprehensive tools for Java development. It includes features for coding, debugging, and deploying Java applications.

**JDK (Java Development Kit)**

A software development kit used for Java programming. It includes the necessary tools, executables, and binaries required for Java application development.

**XAMPP**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists of Apache HTTP Server, MariaDB database, and interpreters for scripting languages like PHP and Perl.

MySQL Database:

MySQL is an open-source relational database management system (RDBMS). It is commonly used for managing and storing structured data and is a popular choice in web development.

**JDBC (Java Database Connectivity)**

A Java-based API (Application Programming Interface) that allows Java applications to interact with relational databases. JDBC provides a standard interface for connecting Java applications with different databases.

**GUI (Graphical User Interface) / Java Swing**

GUI refers to the graphical user interface of a software application, providing visual elements such as windows, buttons, and menus. Java Swing is a set of GUI components for Java applications, allowing the creation of interactive and visually appealing user interfaces.

**Implementation**

The process of turning a design or plan into a working system or application. In software development, it involves coding, testing, and deploying a software solution based on the specified requirements.

**Windows 10 OS**

Windows 10 is a widely used operating system developed by Microsoft. It provides a user-friendly interface and supports various software applications, including those developed in Java using tools like NetBeans and the JDK.

**Chapter 2**

**REVIEW OF RELATED LITERATURE AND SYSTEMS**

**Related Literature**

As highlighted in the preceding chapter, the cost factor holds significant importance in the context of the "Attendance Management System" project. Consequently, the design of a system aligning with the specified objectives necessitates comprehensive research and studies, which are detailed in this chapter. This chapter comprises three key subtopics: a prior case study on attendance tracking systems, examinations of hardware components, and investigations into relevant software systems.

The integration of technology in educational institutions is transforming administrative processes, with a particular emphasis on enhancing efficiency and accuracy. A fundamental aspect of academic administration is attendance tracking, crucial for maintaining the integrity of the learning environment. This literature review explores existing systems and research related to attendance tracking in educational settings, focusing on the development of a " Attendance Management System " for schools. By examining the advancements and challenges presented in various attendance tracking solutions, this review seeks to inform the design and implementation of an effective system tailored to the unique requirements of computer laboratories.

According to Arulogun et al. (2013) proposed an RFID-based Students Attendance Management System, utilizing RFID technology for real-time data collection. This system provides insights into the challenges and solutions related to attendance tracking, offering valuable lessons applicable to the proposed Attendance Management System. The adaptability of RFID technology to the computer laboratory environment is a focal point for consideration.

Another Study by Hoo and Ibrahim (2019) conducted a literature survey on hardware requirements for a Biometric-Based Attendance Tracking System. Their work delves into the intricacies of biometric authentication, particularly fingerprint recognition, ensuring precise attendance monitoring and eliminating proxy attendance. The security considerations and hardware insights from this study are pertinent to the development of a secure and accurate attendance tracking solution for the proposed management system.

Moreover, Tin (2023) explores the landscape of School Management Software and how campus management systems, such as "Minga.io," integrate attendance tracking modules. This comprehensive approach to educational administration, managing student records alongside attendance monitoring, provides valuable insights into the balance between specialized functionality and integration with larger institutional systems. Extracting relevant features from such comprehensive systems informs the compatibility of the proposed Attendance Management System within broader school management frameworks.

In conclusion, the literature review underscores the diversity of attendance tracking systems in educational settings. The challenges and successes documented in the works of Arulogun et al. (2013), Hoo and Ibrahim (2019), and Tin (2023) offer guidance for the development of a specialized "Attendance Management System." By synthesizing the knowledge gained from these sources, the proposed solution can address the unique requirements of computer laboratories, ensuring accuracy, security, and seamless integration within the broader educational management context.

**Related Systems**

Specializes in the design and development of tailored solutions dedicated to the effective management and tracking of attendance in computer laboratories. Includes:

**Laboratory Attendance Management System**

The Laboratory Attendance Management System, as described by Loksharan (2019), offers an efficient means of marking attendance and retrieving information in a computer laboratory setting. Developed using PHP and MySQL, the system is designed for easy integration with web applications and effective database management. Its significance lies in streamlining attendance tracking, thereby contributing to proper supervision, accurate payroll calculations, and efficient resource allocation. This system aligns with the objectives of the " Attendance Management System," aiming to improve attendance management in computer laboratories by providing valuable data for decision-making (Loksharan, 2019).

**Computer Laboratories Attendance System (CLAS)**

According to Nietes (2016), the Computer Laboratories Attendance System (CLAS) is an automated solution that enhances how professors record and track student attendance in computer laboratories. Functioning to streamline attendance maintenance, CLAS simplifies processes for professors, aiding in accurate attendance tracking, essential for proper supervision, budgeting, and resource allocation. In connection to the " Attendance Management System " capstone project, both systems share the common goal of tracking attendance within a computer laboratory setting, albeit potentially employing different approaches (Nietes, 2016).

**Access Control and Monitoring System for Computer Laboratory**

Zapanta, Talirongan, and Talirongan (2021) describe the Access Control and Monitoring System for Computer Laboratory as a tool for system administrators to effectively track resources and manage computer labs. The system allows lab assistants to register users, remotely lock/unlock workstations, maintain waitlists, and monitor room occupancy. The importance of this system lies in efficient resource management and security for the smooth operation of computer laboratories. In connection to our capstone project, this system shares the objective of enhancing resource management and security, contributing to the overall efficiency of computer laboratories (Zapanta et al., 2021).

In conclusion, the Laboratory Attendance Management System, Computer Laboratories Attendance System (CLAS), and Access Control and Monitoring System for Computer Laboratory are all directly related to the " Attendance Management System " capstone project. These systems collectively aim to improve attendance tracking and management within computer laboratories, providing valuable data for informed decision-making. While each system has unique features and approaches, the shared objective is to enhance the overall efficiency and supervision of computer laboratories.

**Chapter 3**

**RESULTS AND DISCUSSION**

This section delves into the comprehensive analysis of survey data collected from key stakeholders, including 3 faculty members, 2 teachers, and 20 students enrolled in the Bachelor of Science in Industrial Technology program at Talisay City College. The respondents were selected through a convenience sampling method, representing individuals who regularly utilize Computer Laboratory 1 and 2. This survey was conducted to identify and understand the challenges faced by faculty teachers and those in charge of the computer laboratory, particularly in the context of the current manual attendance tracking system.

**Table 2 *What are potential problem and issue of current manual tracking of attendance?***

|  |  |  |
| --- | --- | --- |
|  | AGREE | DISAGREE |
| **1.1 Time Inefficiency** | | |
| 1.1.1 Do you agree that the current manual attendance tracking system in the computer laboratory is time-inefficient? | Teacher:2 Faculty:2 Student:17 | Teacher:0 Faculty:1 Student:3 |
| **1.2 Error-Prone Recording** | | |
| 1.1.2Do you agree that the manual attendance system often leads to errors in recording attendance? | Teacher:2 Faculty:3 Student:17 | Teacher:0 Faculty:0 Student:3 |
| **1.3 Lack of Device Tracking Capabilities** | | |
| 1.1.3 Do you agree that there is a lack of device tracking capabilities in the current attendance tracking system? | Teacher:2 Faculty:2 Student:16 | Teacher:0 Faculty:1 Student:4 |
| **1.4 Instances of Lost and Stolen Devices** | | |
| 1.1.4 Do you agree that the current attendance tracking method has cases of lost and stole devices? | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |
| **1.5Misplacement of Paper Records** | | |
| 1.1.5 Do you agree that there are instance sometimes misplace the attendance tracking paper? | Teacher:2 Faculty:3 Student:17 | Teacher:0 Faculty:0 Student:3 |
| **1.6 Inefficient Response to Lost or Stolen Devices** | | |
| 1.1.6 Do you agree that there inefficient response to lost or stolen devices? | Teacher:2 Faculty:3 Student:15 | Teacher:0 Faculty:0 Student:5 |
| **1.7 Security Risks** | | |
| 1.1.7 Do you agree that there is a security risks like publicity of user information of users like ID number may use is future bad actions like fraud? | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |
| **1.8 Privacy Concerns** | | |
| 1.1.8 Do you agree that the current attendance tracking system not addresses privacy concerns related to attendance data? | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |

The data presented on the Attendance Management System reveals several consistent patterns of agreement and disagreement among teachers, faculty, and students.

Concerning Time Inefficiency (1.1), there is a notable agreement across all groups, with teachers (2), faculty (2), and students (17) concurring that the current manual attendance tracking system in the computer laboratory is time-inefficient. This collective agreement underscores a widespread perception of delays or inefficiencies in the existing process.

Addressing the issue of Error-Prone Recording (1.2), a consensus emerges as teachers (2), faculty (3), and students (17) all agree that the manual attendance system often leads to errors in recording attendance. This shared acknowledgment emphasizes the need for improvements to enhance the accuracy and reliability of attendance records.

Regarding the Lack of Device Tracking Capabilities (1.3), there is alignment among teachers (2), faculty (2), and students (16) in recognizing a perceived deficiency in device tracking capabilities in the current attendance tracking system. This points to a shared concern about accountability and the need for improved tracking features.

The Instances of Lost and Stolen Devices (1.4) are acknowledged by teachers (2), faculty (3), and students (20), with unanimous agreement that the current attendance tracking method has cases of lost and stolen devices. As per Vladimir statement says and also our School Laboratory Faculty Admin In-charge also our Java teacher Sir Cupal says” there are cases that there devices has been lost and stolen”. This highlights security vulnerabilities in the existing system, emphasizing the importance of enhanced security measures.

In addressing the Misplacement of Paper Records (1.5), teachers (2), faculty (3), and students (17) collectively agree that there are instances where attendance tracking papers are sometimes misplaced. This agreement underscores the potential risks associated with manual record-keeping and supports the argument for transitioning to a more secure and efficient digital system.

Concerning the Inefficient Response to Lost or Stolen Devices (1.6), there is agreement among teachers (2), faculty (3), and students (15) that the response to such incidents is inefficient. This points to the need for improved protocols and technological solutions to address security concerns effectively.

The perception of Security Risks (1.7) is shared among teachers (2), faculty (3), and students (20), with agreement that there are risks, including the potential publicity of user information like ID numbers that may be exploited for fraudulent activities. This emphasizes the critical need for robust security measures to safeguard sensitive information.

Lastly, Privacy Concerns (1.8) are acknowledged by teachers (2), faculty (3), and students (20), with unanimous agreement that the current attendance tracking system does not address privacy concerns related to attendance data. This underscores the necessity for implementing privacy-focused features in the system to protect user information.

In summary, the consistent agreement among teachers, faculty, and students on various shortcomings in the current manual attendance tracking system strongly advocates for the implementation of a more efficient, secure, and privacy-conscious system to address the identified challenges.

**Table 3 *Do you concur that the suggested*** *AMS* ***(Attendance Management System) would be valuable in enhancing the attendance tracking system?***

|  |  |  |
| --- | --- | --- |
|  | Agree | Disagree |
| 2. Do you believe that the suggested AMS (Attendance Management System) would be valuable in enhancing the attendance tracking system? | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |

Table 3 , the responses to the question regarding the perceived value of the suggested AMS (Attendance Management System) in enhancing the attendance tracking system reveal unanimous agreement across teachers, faculty, and students. Teachers registered a total agreement of 2, faculty members expressed complete agreement with a count of 3, and students overwhelmingly supported the proposed system with a unanimous agreement of 20. Conversely, there were no dissenting opinions among any of the surveyed groups.

This unanimous agreement suggests a widespread recognition of the potential benefits associated with the implementation of AMS. Participants likely view the system as a means to address existing challenges in the manual attendance tracking process. Potential reasons for this consensus may include expectations of improved efficiency, enhanced accuracy and reliability in attendance recording, increased security measures, and overall convenience for both administrators and students. The collective endorsement of AMS is of paramount importance as it signifies a unified understanding of the system's potential advantages. This alignment is instrumental in fostering successful implementation and acceptance of the proposed system, marking a pivotal step towards overcoming the identified shortcomings in the current manual attendance tracking system within the computer laboratory.

**Table 4 *Have you ever accused that you are the one damage, lost and being theft because they were people damage the item without any tracking devices that’s why important to track people?***

|  |  |  |
| --- | --- | --- |
| 3. Have you ever accused that you are the one damage, lost and being theft because the people damage the item without any tracking devices that’s why important to track people? | Teacher:2 Faculty:2 Student:14 | Teacher:0 Faculty:1 Student:6 |

Table 4, the responses to the question on whether individuals have experienced accusations of causing damage, loss, or theft due to the absence of tracking devices provide insights into the perceived importance of tracking people in the context of accountability and preventing false accusations.

The overwhelming agreement among teachers (2), faculty (2), and students (14) indicates a shared belief that the absence of tracking devices has resulted in individuals being wrongly accused. This consensus suggests that there may have been instances where incidents of damage, loss, or theft occurred, and without tracking mechanisms, individuals were unfairly held responsible. This highlights a perceived need for tracking systems to establish a clear and accurate account of events, ultimately preventing unjust blame and fostering a more trustworthy environment.

Reasons for Agreement:

Accountability and Transparency, respondents likely believe that tracking systems contribute to accountability and transparency, providing a reliable means to verify individuals' actions and whereabouts. This is crucial in avoiding baseless accusations and resolving disputes fairly.

Real-life Experiences, the agreement may be influenced by personal experiences or observed incidents where accusations were made without adequate evidence, reinforcing the belief in the importance of tracking mechanisms.

Reasons for Disagreement

Privacy Concerns, the respondents, particularly students (4) and faculty (1) who disagree, may express concerns about privacy invasion associated with tracking devices. This disagreement suggests a tension between the perceived benefits of tracking for accountability and the desire to protect individual privacy.

In the broader context, the significance of tracking people lies in its potential to address issues of false accusations, enhance accountability, and contribute to a more secure and responsible community. However, the disagreement regarding privacy concerns underscores the need for a balanced approach in implementing tracking systems, ensuring that the benefits of accountability are achieved without compromising individual rights and privacy. Discussions and considerations around the ethical use of tracking technologies are essential in addressing these concerns and fostering a harmonious and trusted environment within the community.

**SYSTEM DEVELOPMENT**

The following tables provide an inventory of Tools, Software, Hardware, and Peopleware utilized in the development of the system.

Table 5 specifically outlines the tools and software employed in system development, encompassing Windows 10 OS, JAVA Netbeans IDE, JDK, and XAMPP MySQL for Database to compile code for both hardware and software components.

**Table 5**

**List of Tool and Software used in System Development**

|  |  |
| --- | --- |
| **SOFTWARE** | **USES** |
| Windows 10 OS | Operating system for overall system functionality. |
| JAVA NetBeans IDE | Integrated Development Environment for Java application development. |
| JAVA JDK 17 | Java Development Kit for comprehensive Java application development. |
| XAMPP MySQL | Database management tool for storing and retrieving system data. |
| MySQL (JDBC) Connector | A JDBC driver for connecting MySQL databases to Java applications. |

Table 6 shows the list of hardware used in system development. Arduino Mega 2560 Pro is used as a controller to the following modules: SIM800L GSM, Ultrasonic Sensor, Micro Servo, Buzzer, and Potentiometer.

**Table 6**

**List of Hardware used in System Development**

|  |  |
| --- | --- |
| **HARDWARE** | **USES** |
| Computer Set with Compatible with Windows 7, 8, and 10 and minimum of 2 GB RAM (4 GB or more recommended). Sufficient free disk space (varies but generally around 1 GB for NetBeans and 400 MB for XAMPP). | Essential for running software applications and development environments. |
| Screen Display | The computer set supports a screen resolution of 1024 x 768 pixels or higher, providing a clear and detailed display for programming, debugging, and other software-related activities. |

**Peopleware**

This is the section where we outline the responsibilities of the individuals contributing to this project. The team consists of Herbert Tomi-as Acedre, who serves as the Project Manager, System Analyst, Programmer, and QA Tester/Technical Writer. Additionally, Jhuztin Obenza plays a role in QA as a Quality Assurance Tester. Vannessa Dayondon handles expenses specialization, while Mariel Bazar takes on the role of the Database Administrator, and Rasbel Sebial contributes to the same capacity.

**Chapter 4**

**CONCLUSION AND RECOMMENDATIONS**

**Conclusion**

The examination of the existing manual attendance tracking system at Talisay City College has uncovered significant challenges, encompassing issues such as time inefficiency, error-prone recording, lack of device tracking, security risks, and privacy concerns. The unanimous agreement among teachers, faculty members, and students underscores the urgency for a change, emphasizing the inadequacies of the current system.

In light of these findings, the conclusion is drawn that a transition to a more advanced and secure solution is imperative. The statistical analysis reinforces the necessity for an enhanced system that addresses the identified challenges and provides a more efficient and reliable approach to attendance tracking.

**Recommendations**

Based on the comprehensive analysis, we recommend the implementation of the "Attendance Management System" at Talisay City College. This initiative specifically targets Computer Laboratories 1, 2, and 3, aiming to address the challenges associated with the manual attendance tracking system and enhance security measures within the computer laboratories.

Key Features of the Proposed System:

* Digital identification using a Java application for accurate attendance records.
* Real-time tracking of individuals within the computer laboratories.
* Detailed logs to identify individuals present during specific timeframes.
* Quick and precise identification of users responsible for device loss or damage.

The proposed system is envisioned as a pivotal step toward modernizing and securing the educational environment at Talisay City College. We believe it aligns with the college's commitment to academic excellence, providing an innovative solution to enhance the overall educational landscape.

In conclusion, the successful implementation of this project is expected to not only improve attendance tracking but also contribute to a safer and more secure learning environment for the entire Talisay City College community. We look forward to bringing this vision to fruition and making a lasting impact on the educational experience at the institution.

**Chapter 5**

**OUTPUT OF THE STUDY**

This chapter presents the output of the study.

**Rationale**

The " Attendance Management System " is envisioned as a transformative solution to address the identified challenges within the manual attendance tracking system at Talisay City College. The rationale behind this project stems from the imperative need to enhance efficiency, accuracy, and security in the recording of attendance within the computer laboratories. The manual system's shortcomings, including time inefficiency, error-prone recording, and a lack of device tracking capabilities, underscore the urgency of implementing a modernized approach. The proposed system, leveraging digital identification through a Java application and real-time tracking features, aims to revolutionize the current process. By streamlining attendance recording and introducing advanced security measures, this project aims to provide a more robust and reliable system, aligning with the college's commitment to academic excellence and technological advancement.

**Target Clientele**

The primary beneficiaries of the " Attendance Management System " are the respondents involved in the study, including teachers, faculty members, and students of Talisay City College. Additionally, the project specifically targets the Computer Laboratories 1, 2, and 3 at the college. The system is designed to cater to the unique needs and challenges faced by these stakeholders, ensuring a user-friendly interface and addressing the intricacies of attendance tracking within the computer laboratories. By tailoring the solution to the specific requirements of Talisay City College, the project aims to significantly improve the user experience and overall effectiveness of attendance management for both educators and students.

**Scheme of Implementation**

The implementation of the proposed system will follow a structured and phased approach. The scheme involves:

Project Planning

* Define project objectives, scope, and deliverables.
* Formulate a detailed project plan outlining timelines, resource requirements, and milestones.

System Development:

* Develop the " Attendance Management System " based on the specified features and requirements.
* Conduct rigorous testing to ensure functionality, security, and user-friendliness.

Pilot Testing:

* Implement the system in a controlled environment, such as one of the computer laboratories.
* Gather feedback from users to identify any refinements or improvements needed.

Full Deployment:

* Roll out the system across Computer Laboratories 1, 2, and 3 after addressing feedback from the pilot phase.
* Provide necessary training to users to ensure a smooth transition to the new system.

Monitoring and Evaluation:

* Continuously monitor system performance and user satisfaction.
* Conduct regular evaluations to identify areas for optimization and enhancement.

Documentation and Support:

* Develop comprehensive documentation for users and administrators.
* Provide ongoing support and maintenance to address any issues or updates.

The scheme of implementation is designed to ensure a systematic and effective transition to the new attendance tracking system, maximizing its benefits for both the targeted clientele and Talisay City College as a whole.

**BIBLIOGRAPHY**

1. **Internet Source**

Arulogun, O. T., Olatunbosun, A., A, F. O., & Olaniyi, O. M. (2013). RFID-Based Students Attendance Management System. *ResearchGate*. https://www.researchgate.net/publication/235598499\_RFID-Based\_Students\_Attendance\_Management\_System

Hoo, S. C., & Ibrahim, H. (2019). Biometric-Based Attendance Tracking System for Education Sectors: A literature Survey on hardware requirements. *Journal of Sensors*, *2019*, 1–25. https://doi.org/10.1155/2019/7410478

LOKSHARAN, S. (2019, November). *Laboratory Attendance Management System.docx*. Scribd. Retrieved November 28, 2023, from https://www.scribd.com/document/431651516/Laboratory-Attendance-Management-system-docx

Nietes, D. (2016, November 16). *COMPUTER LABORATORIES ATTENDANCE SYSTEM (CLAS)*. prezi.com. Retrieved November 28, 2024, from https://prezi.com/csjooyz4y\_ja/computer-laboratories-attendance-system-clas/

Tin, M. (2023, October 23). *School Management Software: How campus management systems fit in*. Minga.io. https://minga.io/school-management-software-campus-management-platforms/

Zapanta, D. B., Talirongan, H. . ., & Talirongan, F. J. B. (2021a). Access Control and Monitoring: a system for computer laboratory. *ResearchGate*. https://www.researchgate.net/publication/349278732\_Access\_Control\_and\_Monitoring\_A\_System\_for\_Computer\_Laboratory

Zapanta, D. B., Talirongan, H. . ., & Talirongan, F. J. B. (2021b). Access Control and Monitoring: a system for computer laboratory. *ResearchGate*. https://www.researchgate.net/publication/349278732\_Access\_Control\_and\_Monitoring\_A\_System\_for\_Computer\_Laboratory

Zhao, M., Zhao, G., & Qu, M. (2022). College Smart Classroom attendance Management system based on Internet of things. *Computational Intelligence and Neuroscience*, *2022*, 1–9. https://doi.org/10.1155/2022/4953721

**APPENDICES**

**APPENDIX A**

**LETTER TO THE RESPONDENTS**

November 7, 2023

**MR. RICHARD CUPAL  
MR. BHABETS A. CANONAYON**

Good day!

I hope this letter finds you well.

We, students of Talisay City College pursuing a Bachelor of Science in Computer Technology, are seeking your permission for a survey related to our capstone thesis, "Computer Laboratory Attendance Tracker Management System.”, with the respondent of maximum 4 Teachers who usually use the Computer Laboratory, 10 Students and 2 Admin Staffs (Ex. Working Student/Student Assistant and etc.)

Survey Objectives is to evaluate if there are current attendance tracking methods in the Computer Laboratory. Identify challenges and concerns students face in attendance tracking. Assess overall satisfaction with the Computer Laboratory facilities.

We assure you that the survey will adhere to college guidelines, maintaining confidentiality for academic purposes only.

Your approval is vital for the success of our research and the improvement of Computer Laboratory facilities. We appreciate your support and look forward to a favorable response.

Thank you.

Sincerely yours,

Leader: Members:

Herbert T. Acedre Jhuztin Niño Obenza

Vanessa Dayondon

Rasbel Sebial

Mariel Bazar

Approved By:

**RICHARD CUPAL, LPT BHABETHS A. CANONAYON, LPT**

Adviser College Comp-Lab3 In-Charge

**APPENDIX B**

**LETTER OF INVITATION TO THE CLIENTELE**

**November 28, 2023**

Ms. **Alyana Marie D. Ong**

Industrial Technology, Faculty

Talisay City College

Poblacion, Talisay City, Cebu

**Subject: BS INDUSTRIAL TECHNOLOGY STUDENTS’ CAPSTONE PROJECT PANELIST INVITATION**

Respected **Ms. Alyana Marie D. Ong**

I hope this message finds you well. As we approach the culminating event of our academic year, I am thrilled to extend an invitation for you to join us as a panelist for our students' Capstone Project Presentations.

Our students have been diligently working on their Capstone Projects, and their presentations on **December 15, 2023**, mark the culmination of their academic journey. Your expertise and insights would be invaluable in evaluating and providing feedback to our aspiring graduates.

The Capstone Presentations will take place at **COMPUTER LABORATORY** 2 and are scheduled from **8:00 AM – 10:30AM**. To ensure our panelists are comfortable and energized throughout the sessions, complimentary snacks and refreshments will be provided.

Your participation as a panelist would involve a brief introduction of yourself, engaging with the students' presentations, and concluding with constructive feedback and comments.

We greatly admire your accomplishments and believe that your expertise aligns perfectly with the objectives of our Capstone Projects. Your presence would not only enrich the event but also inspire our students as they embark on their professional careers.

Your confirmation of attendance by **December 15, 2023** would be highly appreciated to ensure smooth logistical arrangements for the event.

Thank you for considering this invitation. We sincerely hope you can join us and contribute to the success of our students' Capstone Project Presentations while enjoying our complimentary snacks.

We eagerly anticipate your positive response and your potential participation in this momentous occasion.

Warm regards,

**Richard C. Cupal, LPT**

Industrial Technology, Faculty

**Ms. Alyana Marie D. Ong**

Clientele/Faculty

**APPENDIX C**

**SURVEY QUESTIONAIRE**

We, fourth-year Computer Technology students at Talisay City College (BSIndTech program), are conducting research on the " Attendance Management System." We kindly ask for your cooperation in answering our confidential questionnaire, as your input is crucial to our study's success. Thank you for your participation.

|  |  |  |
| --- | --- | --- |
|  | AGREE | DISAGREE |
| **1. What are potential problem and issue of current manual tracking of attendance in terms of:** |  |  |
| **1.1Time Inefficiency** | | |
| 1.1.1 Do you agree that the current manual attendance tracking system in the computer laboratory is time-inefficient? | Teacher:2 Faculty:2 Student:17 | Teacher:0 Faculty:1 Student:3 |
| **1.2 Error-Prone Recording** | | |
| 1.1.2 Do you agree that the manual attendance system often leads to errors in recording attendance? | Teacher:2 Faculty:3 Student:17 | Teacher:0 Faculty:0 Student:3 |
| **1.3 Lack of Device Tracking Capabilities** | | |
| 1.1.3 Do you agree that there is a lack of device tracking capabilities in the current attendance tracking system? | Teacher:2 Faculty:2 Student:16 | Teacher:0 Faculty:1 Student:4 |
| **1.4 Instances of Lost and Stolen Devices** | | |
| 1.1.4 Do you agree that the current attendance tracking method causes delays in accessing the computer laboratory? | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |
| **1.5 Misplacement of Paper Records** | | |
| 1.1.5 Do you agree that there are instance sometimes misplace the attendance tracking paper? | Teacher:2 Faculty:3 Student:17 | Teacher:0 Faculty:0 Student:3 |
| **1.6 Inefficient Response to Lost or Stolen Devices** | | |
| 1.1.6 Do you agree that there inefficient response to lost or stolen devices | Teacher:2 Faculty:3 Student:15 | Teacher:0 Faculty:0 Student:5 |
| **1.7 Security Risks** | | |
| 1.1.7 Do you agree that there is a security risks like publicity of user information of users like ID number may use is future bad actions like fraud? | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |
| **1.8 Privacy Concerns** | | |
| 1.1.8 Do you agree that the current attendance tracking system not addresses privacy concerns related to attendance data | Teacher:2 Faculty:3 Student:20 | Teacher:0 Faculty:0 Student:0 |
| **2. Do you believe that the suggested AMS (Attendance Management System) would be valuable in enhancing the attendance tracking system?** | Teacher:2 Faculty:3 Student:20 | 5Teacher:0 Faculty:0 Student:0 |
| **3. Have you ever accused that you are the one damage, lost and being theft because the were people damage the item without any tracking devices that’s why important to track people?** | Teacher:2 Faculty:2 Student:14 | Teacher:0 Faculty:1 Student:6 |

**CODES**

LOGIN CODES

import General.ConnectionProvider;

import java.sql.\*;

import java.sql.PreparedStatement;

import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

/\*\*

\* Creates new form Login

\*/

public Login() {

initComponents();

}

// ===================================Start Function =========================================================//

public void showLogin(){

try {

// Create the SQL query with placeholders

String getQuery = "SELECT \* FROM log\_in";

// Create a connection

ConnectionProvider dbc = new ConnectionProvider();

String jdbcDriver = dbc.getJdbcDriver();

String dbConnectionURL = dbc.getDbConnectionURL();

String dbUsername = dbc.getDbUsername();

String dbPassword = dbc.getDbPassword();

Class.forName(jdbcDriver);

Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);

// Create the PreparedStatement

PreparedStatement statement = connection.prepareStatement(getQuery);

statement.executeQuery();

ResultSet resultSet = statement.executeQuery();

// Close resources

resultSet.close();

statement.close();

connection.close();

// Display a success message

JOptionPane.showMessageDialog(null, "LOGIN Successful!", "Success", JOptionPane.INFORMATION\_MESSAGE);

System.out.println("Retrieved Successfully!");

} catch (ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

};

// ===================================End Function =========================================================//

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

username\_field = new javax.swing.JTextField();

password\_field = new javax.swing.JPasswordField();

jButton1 = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(0, 102, 102));

jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

jLabel2.setForeground(new java.awt.Color(255, 255, 153));

jLabel2.setText("FACULTY USERNAME");

jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 90, -1, -1));

jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

jLabel3.setForeground(new java.awt.Color(255, 255, 153));

jLabel3.setText("PASSWORD");

jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 220, -1, -1));

username\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

username\_fieldActionPerformed(evt);

}

});

jPanel1.add(username\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 180, 160, 30));

jPanel1.add(password\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 240, 160, 30));

password\_field.getAccessibleContext().setAccessibleName("");

jButton1.setBackground(new java.awt.Color(0, 102, 102));

jButton1.setForeground(new java.awt.Color(255, 255, 255));

jButton1.setText("LOGIN");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 290, 98, 38));

jLabel4.setForeground(new java.awt.Color(255, 255, 255));

jLabel4.setText("Create an account?");

jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 350, -1, -1));

jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 48)); // NOI18N

jLabel5.setForeground(new java.awt.Color(255, 255, 153));

jLabel5.setText("LOGIN");

jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 20, -1, -1));

jLabel1.setForeground(new java.awt.Color(0, 255, 255));

jLabel1.setText("SIGN UP");

jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel1MouseClicked(evt);

}

});

jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 350, -1, -1));

jLabel6.setForeground(new java.awt.Color(255, 153, 153));

jLabel6.setText("\"Use this format for Working Student 0000-0000@LASTNAME\"");

jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(27, 116, -1, -1));

jLabel7.setForeground(new java.awt.Color(255, 153, 153));

jLabel7.setText("\"Use this format for Faculty Teacher 0000-00-00-000@LASTNAME\"");

jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(18, 138, -1, -1));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 404, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 408, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void username\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Get the entered username

String username = username\_field.getText();

// Define the expected pattern using regular expression

String pattern = "\\d{4}-\\d{4}-\\d{2}@\\w+";

// Check if the entered username matches the pattern

if (username.matches(pattern)) {

// Valid username format

System.out.println("Username is in the correct format: " + username);

} else {

// Invalid username format

System.err.println("Error: Invalid username format. Please use the format 1046-2020-22@FULLNAME");

// You can also show a message dialog or set an error label to provide feedback to the user

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// TODO add your handling code here:

try {

// Your existing code inside the try block

String getQuery = "SELECT username, password FROM log\_in WHERE username = ? AND password = ?";

ConnectionProvider dbc = new ConnectionProvider();

String jdbcDriver = dbc.getJdbcDriver();

String dbConnectionURL = dbc.getDbConnectionURL();

String dbUsername = dbc.getDbUsername();

String dbPassword = dbc.getDbPassword();

Class.forName(jdbcDriver);

try (Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);

PreparedStatement statement = connection.prepareStatement(getQuery)) {

// Set values for the placeholders

statement.setString(1, username\_field.getText());

statement.setString(2, new String(password\_field.getPassword())); // Use getPassword for password fields

try (ResultSet resultSet = statement.executeQuery()) {

// Process the result set

if (resultSet.next()) {

PORTAL portal = new PORTAL(username\_field.getText());

portal.setVisible(true);

this.dispose();

JOptionPane.showMessageDialog(null, "LOGIN Successful", "Success", JOptionPane.INFORMATION\_MESSAGE);

System.out.println("User retrieved successfully");

} else {

JOptionPane.showMessageDialog(null, "Invalid username or password", "Error", JOptionPane.ERROR\_MESSAGE);

System.out.println("User not found");

}

}

}

} catch (ClassNotFoundException | SQLException ex) {

// Handle exceptions here

ex.printStackTrace();

}

}

private boolean isValidUsername(String username) {

// Define the expected pattern using regular expression

String pattern = "\\d{4}-\\d{4}-\\d{2}@\\w+";

// Check if the entered username matches the pattern

return username.matches(pattern);

}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

// TODO add your handling code here:

Register res = new Register();

res.setVisible(true);

res.pack();

res.setLocationRelativeTo(null);

this.dispose();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

private javax.swing.JPasswordField password\_field;

private javax.swing.JTextField username\_field;

// End of variables declaration

}

**REGISTER CODES**

**import General.ConnectionProvider;**

**import java.sql.\*;**

**import javax.swing.JOptionPane;**

**import java.util.regex.\*;**

**/\***

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

**\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template**

**\*/**

**/\*\***

**\***

**\* @author PC08-LAB1**

**\*/**

**public class Register extends javax.swing.JFrame {**

**/\*\***

**\* Creates new form RESET**

**\*/**

**public Register() {**

**initComponents();**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">**

**private void initComponents() {**

**jPanel1 = new javax.swing.JPanel();**

**jLabel1 = new javax.swing.JLabel();**

**jLabel2 = new javax.swing.JLabel();**

**jLabel3 = new javax.swing.JLabel();**

**username\_field = new javax.swing.JTextField();**

**password\_field = new javax.swing.JTextField();**

**jButton2 = new javax.swing.JButton();**

**jLabel4 = new javax.swing.JLabel();**

**jLabel5 = new javax.swing.JLabel();**

**jLabel6 = new javax.swing.JLabel();**

**setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);**

**jPanel1.setBackground(new java.awt.Color(0, 102, 102));**

**jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());**

**jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 48)); // NOI18N**

**jLabel1.setForeground(new java.awt.Color(255, 255, 153));**

**jLabel1.setText("REGISTER");**

**jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 40, -1, -1));**

**jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N**

**jLabel2.setForeground(new java.awt.Color(255, 255, 153));**

**jLabel2.setText("FACULTY USERNAME");**

**jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 110, -1, -1));**

**jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N**

**jLabel3.setForeground(new java.awt.Color(255, 255, 153));**

**jLabel3.setText("NEW PASSWORD");**

**jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 220, -1, -1));**

**username\_field.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**username\_fieldActionPerformed(evt);**

**}**

**});**

**jPanel1.add(username\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 180, 179, 32));**

**password\_field.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**password\_fieldActionPerformed(evt);**

**}**

**});**

**jPanel1.add(password\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 250, 179, 32));**

**jButton2.setBackground(new java.awt.Color(0, 102, 102));**

**jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N**

**jButton2.setForeground(new java.awt.Color(255, 255, 255));**

**jButton2.setText("REGISTER");**

**jButton2.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**jButton2ActionPerformed(evt);**

**}**

**});**

**jPanel1.add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 310, 179, 43));**

**jLabel4.setForeground(new java.awt.Color(0, 255, 255));**

**jLabel4.setText("<< Return to Login");**

**jLabel4.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jLabel4MouseClicked(evt);**

**}**

**});**

**jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(26, 6, -1, 21));**

**jLabel5.setForeground(new java.awt.Color(255, 153, 153));**

**jLabel5.setText("\"Use this format for Working Student 0000-0000@LASTNAME\"");**

**jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(43, 129, -1, -1));**

**jLabel6.setForeground(new java.awt.Color(255, 153, 153));**

**jLabel6.setText("\"Use this format for Faculty Teacher 0000-00-00-000@LASTNAME\"");**

**jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(32, 151, 364, -1));**

**javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());**

**getContentPane().setLayout(layout);**

**layout.setHorizontalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 408, Short.MAX\_VALUE)**

**);**

**layout.setVerticalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 374, Short.MAX\_VALUE)**

**);**

**pack();**

**setLocationRelativeTo(null);**

**}// </editor-fold>**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**try {**

**// Create the SQL query with placeholders**

**String getQuery = "SELECT \* FROM log\_in";**

**String insertQuery = "INSERT INTO log\_in( username, password) VALUES ( ?, ?)";**

**// Create a connection**

**ConnectionProvider dbc = new ConnectionProvider();**

**String jdbcDriver = dbc.getJdbcDriver();**

**String dbConnectionURL = dbc.getDbConnectionURL();**

**String dbUsername = dbc.getDbUsername();**

**String dbPassword = dbc.getDbPassword();**

**Class.forName(jdbcDriver);**

**Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);**

**// Create the PreparedStatement**

**PreparedStatement statement = connection.prepareStatement(insertQuery);**

**statement.setString(1, username\_field.getText());**

**statement.setString(2, password\_field.getText());**

**statement.executeUpdate();**

**this.username\_field.setText("");**

**this.password\_field.setText("");**

**// Display a success message**

**JOptionPane.showMessageDialog(null, "Registration Successful!", "Success", JOptionPane.INFORMATION\_MESSAGE);**

**System.out.println("Add Successfully!");**

**} catch (ClassNotFoundException | SQLException e) {**

**JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);**

**e.printStackTrace();**

**}**

**}**

**private boolean isValidUsername(String username) {**

**// Define the expected pattern using regular expression**

**String pattern = "\\d{4}-\\d{4}-\\d{2}@\\w+";**

**// Check if the entered username matches the pattern**

**return username.matches(pattern);**

**}**

**private void password\_fieldActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**}**

**private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {**

**// TODO add your handling code here:**

**// TODO add your handling code here:**

**Login log = new Login();**

**log.setVisible(true);**

**log.pack();**

**log.setLocationRelativeTo(null);**

**this.dispose();**

**}**

**private void username\_fieldActionPerformed(java.awt.event.ActionEvent evt) {**

**// Get the entered username**

**String username = username\_field.getText();**

**// Define the expected pattern using regular expression**

**String pattern = "\\d{4}-\\d{4}-\\d{2}@\\w+";**

**// Check if the entered username matches the pattern**

**if (username.matches(pattern)) {**

**// Valid username format**

**System.out.println("Username is in the correct format: " + username);**

**} else {**

**// Invalid username format**

**System.err.println("Error: Invalid username format. Please use the format 1046-2020-22@FULLNAME");**

**// You can also show a message dialog or set an error label to provide feedback to the user**

**}**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Register().setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JButton jButton2;**

**private javax.swing.JLabel jLabel1;**

**private javax.swing.JLabel jLabel2;**

**private javax.swing.JLabel jLabel3;**

**private javax.swing.JLabel jLabel4;**

**private javax.swing.JLabel jLabel5;**

**private javax.swing.JLabel jLabel6;**

**private javax.swing.JPanel jPanel1;**

**private javax.swing.JTextField password\_field;**

**private javax.swing.JTextField username\_field;**

**// End of variables declaration**

**}**

**PORTAL Codes**

import General.ConnectionProvider;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

import javax.swing.table.DefaultTableModel;

import javax.swing.Timer;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Random;

import java.util.UUID;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.JTextField;

public class PORTAL extends javax.swing.JFrame {

private String POSITION;

private String ID\_NUMBER;

private String FULL\_NAME;

private String TIME\_IN;

private String TIME\_OUT;

private String DATE;

private String PC\_NUMBER;

private String SUBJECT;

private boolean idSet;

private String UNI\_CODE;

private String FACULTY\_USER;

private String USERNAME;

private boolean componentsInitialized = false;

/\*\*

\* Creates new form PORTAL

\*/

public PORTAL(String username) {

if (!componentsInitialized) {

initComponents();

componentsInitialized = true; // Set the flag to true after initializing components

dt();

times();

uni();

}

jLabel6.setText(username);

}

public void dt() {

// Use java.util.Date instead of Date() constructor

java.util.Date d = new java.util.Date();

// Use SimpleDateFormat with "yyyy-MM-dd" pattern

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

// Format the date and set it to the label

String dd = sdf.format(d);

jLabel12.setText(dd);

}

public void uni(){

Random rn = new Random();

uni\_field.setText(Integer.toString(rn.nextInt(100000000)));

}

public void times() {

Timer t = new Timer(1000, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Date dt = new Date(System.currentTimeMillis());

SimpleDateFormat st = new SimpleDateFormat("hh:mm:ss a");

String tt = st.format(dt);

jLabel13.setText(tt); // Assuming you have a label l\_time to display the time

}

});

t.start();

}

// ======================================================= Start Function =========================================================//

public void showportal(){

try {

// Create the SQL query with placeholders

String getQuery = "SELECT \* FROM portal";

// Create a connection

ConnectionProvider dbc = new ConnectionProvider();

String jdbcDriver = dbc.getJdbcDriver();

String dbConnectionURL = dbc.getDbConnectionURL();

String dbUsername = dbc.getDbUsername();

String dbPassword = dbc.getDbPassword();

Class.forName(jdbcDriver);

Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);

// Create the PreparedStatement

PreparedStatement statement = connection.prepareStatement(getQuery);

statement.executeQuery();

ResultSet resultSet = statement.executeQuery();

// Process the result set as needed

if (resultSet.next()) {

this.POSITION = resultSet.getString("POSITION");

this.ID\_NUMBER = resultSet.getString("ID\_NUMBER");

this.TIME\_IN = resultSet.getString("TIME\_IN");

this.TIME\_OUT = resultSet.getString("TIME\_OUT");

this.DATE = resultSet.getString("DATE");

this.PC\_NUMBER = resultSet.getString("PC\_NUMBER");

this.SUBJECT = resultSet.getString("SUBJECT");

this.UNI\_CODE= resultSet.getString("UNI\_CODE");

this.FACULTY\_USER= resultSet.getString("FACULTY\_USER");

// Display the retrieved data

System.out.println("POSITION: " + this.POSITION);

System.out.println("ID\_NUMBER: " + this.ID\_NUMBER);

System.out.println("FULL\_NAME: " + this.FULL\_NAME);

System.out.println("TIME\_IN: " + this.TIME\_IN);

System.out.println("TIME\_OUT: " + this.TIME\_OUT);

System.out.println("DATE: " + this.DATE);

System.out.println("PC\_NUMBER: " + this.PC\_NUMBER);

System.out.println("SUBJECT: " + this.SUBJECT);

System.out.println("UNI\_CODE: " + this.UNI\_CODE);

System.out.println("FACULTY\_USER: " + this.FACULTY\_USER);

// Create a table model to store data

DefaultTableModel tableModel = new DefaultTableModel();

jTable1.setModel(tableModel);

// Get column names and add them to the table model

ResultSetMetaData metaData = resultSet.getMetaData();

int columnCount = metaData.getColumnCount();

for (int i = 1; i <= columnCount; i++) {

String columnName = metaData.getColumnName(i);

if (columnName.equals("POSITION") || columnName.equals("ID\_NUMBER") || columnName.equals("FULL\_NAME") ||

columnName.equals("TIME\_IN") || columnName.equals("TIME\_OUT") || columnName.equals("DATE")

|| columnName.equals("PC\_NUMBER") || columnName.equals("SUBJECT")|| columnName.equals("UNI\_CODE")|| columnName.equals("FACULTY\_USER")) {

tableModel.addColumn(columnName);

}

}

// Add rows to the table model

do {

Object[] rowData = new Object[columnCount];

for (int i = 1; i <= columnCount; i++) {

rowData[i - 1] = resultSet.getObject(i);

}

tableModel.addRow(rowData);

} while (resultSet.next());

} else {

System.out.println("No matching records found.");

JOptionPane.showMessageDialog(null, "No record found!");

}

// Close resources

resultSet.close();

statement.close();

connection.close();

System.out.println("Retrieved Successfully!");

} catch (ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

// </editor-fold>

//===========================================================================================//

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

position\_combo\_box = new javax.swing.JComboBox<>();

jLabel3 = new javax.swing.JLabel();

id\_number\_field = new javax.swing.JTextField();

full\_name\_field = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

time\_in\_field = new javax.swing.JTextField();

time\_out\_field = new javax.swing.JTextField();

date\_field = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

pc\_field = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

subject\_field = new javax.swing.JTextField();

jPanel3 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

searchTF = new javax.swing.JTextField();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

jButton10 = new javax.swing.JButton();

uni\_field = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(0, 102, 102));

jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 102));

jLabel1.setText("AMS PORTAL");

jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 30, -1, -1));

jLabel2.setBackground(new java.awt.Color(255, 255, 255));

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("POSITION");

jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(23, 87, -1, -1));

position\_combo\_box.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "TEACHER", "FACULTY", "STUDENT" }));

position\_combo\_box.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

position\_combo\_boxActionPerformed(evt);

}

});

jPanel1.add(position\_combo\_box, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 84, 161, -1));

jLabel3.setForeground(new java.awt.Color(255, 255, 255));

jLabel3.setText("ID. NO.");

jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(37, 127, -1, -1));

id\_number\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

id\_number\_fieldActionPerformed(evt);

}

});

jPanel1.add(id\_number\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 124, 161, -1));

full\_name\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

full\_name\_fieldActionPerformed(evt);

}

});

jPanel1.add(full\_name\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 164, 161, -1));

jLabel4.setForeground(new java.awt.Color(255, 255, 255));

jLabel4.setText("FULLNAME");

jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(15, 167, -1, -1));

time\_in\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

time\_in\_fieldActionPerformed(evt);

}

});

jPanel1.add(time\_in\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 204, 161, -1));

time\_out\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

time\_out\_fieldActionPerformed(evt);

}

});

jPanel1.add(time\_out\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 245, 161, -1));

date\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

date\_fieldActionPerformed(evt);

}

});

jPanel1.add(date\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 286, 161, -1));

jLabel8.setForeground(new java.awt.Color(255, 255, 255));

jLabel8.setText("PC.NO.");

jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(36, 330, -1, -1));

pc\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

pc\_fieldActionPerformed(evt);

}

});

jPanel1.add(pc\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 327, 161, -1));

jLabel9.setForeground(new java.awt.Color(255, 255, 255));

jLabel9.setText("SUBJECT");

jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(23, 367, -1, -1));

subject\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

subject\_fieldActionPerformed(evt);

}

});

jPanel1.add(subject\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(101, 367, 161, -1));

jPanel3.setBackground(new java.awt.Color(204, 204, 204));

jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"POSITION", "ID. NO.", "FULLNAME", "TIME IN", "TIME OUT", "DATE", "PC NO.", "SUBJECT", "UNI\_CODE", "IN CHARGE"

}

));

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jTable1MouseClicked(evt);

}

});

jScrollPane1.setViewportView(jTable1);

jPanel3.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 990, 470));

jPanel1.add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 52, 990, 469));

jButton1.setBackground(new java.awt.Color(0, 102, 102));

jButton1.setForeground(new java.awt.Color(255, 255, 255));

jButton1.setText("ADD");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 450, 229, -1));

jButton2.setBackground(new java.awt.Color(0, 102, 102));

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("UPDATE");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jPanel1.add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 490, 229, -1));

jButton4.setBackground(new java.awt.Color(0, 102, 102));

jButton4.setForeground(new java.awt.Color(255, 255, 255));

jButton4.setText("RETURN TO LOGIN");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jPanel1.add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(1100, 530, 140, -1));

jButton5.setBackground(new java.awt.Color(0, 102, 102));

jButton5.setForeground(new java.awt.Color(255, 255, 255));

jButton5.setText("SEARCH");

jButton5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton5ActionPerformed(evt);

}

});

jPanel1.add(jButton5, new org.netbeans.lib.awtextra.AbsoluteConstraints(1040, 10, 145, -1));

jButton6.setBackground(new java.awt.Color(0, 102, 102));

jButton6.setForeground(new java.awt.Color(255, 255, 255));

jButton6.setText("EXIT");

jButton6.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton6ActionPerformed(evt);

}

});

jPanel1.add(jButton6, new org.netbeans.lib.awtextra.AbsoluteConstraints(1190, 10, -1, -1));

searchTF.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

searchTFActionPerformed(evt);

}

});

jPanel1.add(searchTF, new org.netbeans.lib.awtextra.AbsoluteConstraints(830, 10, 200, -1));

jLabel10.setForeground(new java.awt.Color(255, 255, 255));

jLabel10.setText("DATE");

jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 530, -1, -1));

jLabel11.setForeground(new java.awt.Color(255, 255, 255));

jLabel11.setText("TIME");

jPanel1.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(700, 530, -1, -1));

jLabel12.setForeground(new java.awt.Color(255, 255, 255));

jLabel12.setText("0");

jPanel1.add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 530, 130, -1));

jLabel13.setForeground(new java.awt.Color(255, 255, 255));

jLabel13.setText("0");

jPanel1.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(760, 530, 110, -1));

jButton7.setBackground(new java.awt.Color(0, 102, 102));

jButton7.setForeground(new java.awt.Color(255, 255, 255));

jButton7.setText("TIME IN");

jButton7.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton7ActionPerformed(evt);

}

});

jPanel1.add(jButton7, new org.netbeans.lib.awtextra.AbsoluteConstraints(7, 204, 82, -1));

jButton8.setBackground(new java.awt.Color(0, 102, 102));

jButton8.setForeground(new java.awt.Color(255, 255, 255));

jButton8.setText("TIME OUT");

jButton8.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton8ActionPerformed(evt);

}

});

jPanel1.add(jButton8, new org.netbeans.lib.awtextra.AbsoluteConstraints(7, 245, -1, -1));

jButton9.setBackground(new java.awt.Color(0, 102, 102));

jButton9.setForeground(new java.awt.Color(255, 255, 255));

jButton9.setText("DATE");

jButton9.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton9ActionPerformed(evt);

}

});

jPanel1.add(jButton9, new org.netbeans.lib.awtextra.AbsoluteConstraints(6, 286, 83, -1));

jButton10.setBackground(new java.awt.Color(0, 102, 102));

jButton10.setForeground(new java.awt.Color(255, 255, 255));

jButton10.setText("RESET");

jButton10.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton10ActionPerformed(evt);

}

});

jPanel1.add(jButton10, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 410, 229, -1));

uni\_field.setBackground(new java.awt.Color(0, 102, 102));

uni\_field.setForeground(new java.awt.Color(0, 102, 102));

uni\_field.setBorder(null);

uni\_field.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

uni\_fieldActionPerformed(evt);

}

});

jPanel1.add(uni\_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 10, 170, 30));

jLabel6.setForeground(new java.awt.Color(255, 255, 255));

jLabel6.setText("jLabel6");

jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 10, 260, -1));

jLabel5.setForeground(new java.awt.Color(255, 255, 255));

jLabel5.setText("IN-CHARGE DUTY:");

jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 10, -1, -1));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 1288, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 558, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void intComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

position\_combo\_box = new javax.swing.JComboBox<>();

jLabel3 = new javax.swing.JLabel();

id\_number\_field = new javax.swing.JTextField();

full\_name\_field = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jButton7 = new javax.swing.JButton();

time\_in\_field = new javax.swing.JTextField();

time\_out\_field = new javax.swing.JTextField();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

date\_field = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

pc\_field = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

subject\_field = new javax.swing.JTextField();

jPanel3 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

searchTF = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(0, 102, 102));

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N

jLabel1.setText("PORTAL");

jLabel2.setText("POSITION");

position\_combo\_box.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "TEACHER", "FACULTY", "STUDENT" }));

jLabel3.setText("ID. NO.");

jLabel4.setText("FULLNAME");

jButton7.setText("TIME IN");

jButton8.setText("TIME OUT");

jButton9.setText("DATE");

jLabel8.setText("PC.NO.");

jLabel9.setText("SUBJECT");

jPanel3.setBackground(new java.awt.Color(204, 204, 204));

jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"POSITION", "ID. NO.", "FULLNAME", "TIME IN", "TIME OUT", "DATE", "PC NO.", "SUBJECT"

}

));

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jTable1MouseClicked(evt);

}

});

jScrollPane1.setViewportView(jTable1);

jPanel3.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, -3, 600, 470));

jButton1.setBackground(new java.awt.Color(0, 204, 51));

jButton1.setText("ADD");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(153, 153, 0));

jButton2.setText("UPDATE");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jButton4.setBackground(new java.awt.Color(102, 204, 255));

jButton4.setText("BACK TO LOGIN");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jButton5.setBackground(new java.awt.Color(204, 102, 0));

jButton5.setText("search");

jButton5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton5ActionPerformed(evt);

}

});

jButton6.setText("EXIT");

jButton6.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton6ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(101, 101, 101)

.addComponent(jLabel1))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(23, 23, 23)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 229, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel4)

.addComponent(jButton7)

.addComponent(jButton8)

.addComponent(jButton9)

.addComponent(jLabel8)

.addComponent(jLabel9))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(position\_combo\_box, 0, 161, Short.MAX\_VALUE)

.addComponent(id\_number\_field)

.addComponent(full\_name\_field)

.addComponent(time\_in\_field)

.addComponent(time\_out\_field)

.addComponent(date\_field)

.addComponent(pc\_field)

.addComponent(subject\_field)))

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 229, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(searchTF, javax.swing.GroupLayout.PREFERRED\_SIZE, 200, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 145, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(43, 43, 43)

.addComponent(jButton6)

.addGap(29, 29, 29))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 142, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(205, 205, 205))

)));

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap(10, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton6)

.addComponent(searchTF, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton5))

.addGap(18, 18, 18)

.addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 469, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(24, 24, 24)

.addComponent(jLabel1)

.addGap(31, 31, 31)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(position\_combo\_box, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addComponent(id\_number\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel4)

.addComponent(full\_name\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton7)

.addComponent(time\_in\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton8)

.addComponent(time\_out\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton9)

.addComponent(date\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel8)

.addComponent(pc\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel9)

.addComponent(subject\_field, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1)

.addGap(18, 18, 18)

.addComponent(jButton2)

.addGap(8, 8, 8)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton4))

.addGap(17, 17, 17))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(24, 24, 24))

);

pack();

}// </editor-fold>

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here

// Validate ID number

String idNumber = id\_number\_field.getText().trim();

String idValidationPattern = "^\\d{1,}(-\\d+)\*$";

if (!idNumber.matches(idValidationPattern)) {

// Invalid ID number format

JOptionPane.showMessageDialog(this, "Invalid ID number format. Please enter a valid ID number.", "Error", JOptionPane.ERROR\_MESSAGE);

return; // Stop execution if the ID is invalid

}

// Validate FULL\_NAME

// Validate FULL\_NAME

String fullName = full\_name\_field.getText().trim();

String nameValidationPattern = "^[a-zA-Z\\s.]+$";

if (!fullName.matches(nameValidationPattern)) {

// Invalid FULL\_NAME format

JOptionPane.showMessageDialog(this, "Invalid name format. Please enter a valid name.", "Error", JOptionPane.ERROR\_MESSAGE);

return; // Stop execution if the FULL\_NAME is invalid

}

String timeIn = time\_in\_field.getText().trim(); // Get the input text and remove leading/trailing spaces

String timeOut = time\_out\_field.getText().trim(); // Get the time-out input text

// Define a regular expression pattern for the allowed format (12-hour clock with AM/PM)

String pattern = "^(0[1-9]|1[0-2]):[0-5][0-9]\\s(?:AM|PM)$";

// Check if the time-in input matches the pattern

if (timeIn.matches(pattern)) {

// Parse the hours from the time-in input

int hoursIn = Integer.parseInt(timeIn.split(":")[0]);

// Check if the time-in is in the morning (AM) or afternoon/evening (PM)

if ((hoursIn >= 1 && hoursIn <= 11) && timeIn.endsWith("?:AM|PM")) {

// Invalid input for time-in (morning time with PM)

System.out.println("Invalid time in: " + timeIn);

JOptionPane.showMessageDialog(this, "Invalid time format for time in. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_in\_field.setText(""); // Clear the time-in input field

return;

}

} else {

// Invalid input for time-in

System.out.println("Invalid input for time in: " + timeIn);

JOptionPane.showMessageDialog(this, "Invalid time format for time in. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_in\_field.setText(""); // Clear the time-in input field

return;

}

// Check if the time-out input matches the pattern

if (timeOut.matches(pattern)) {

// Parse the hours from the time-out input

int hoursOut = Integer.parseInt(timeOut.split(":")[0]);

// Check if the time-out is in the morning (AM) or afternoon/evening (PM)

if ((hoursOut >= 1 && hoursOut <= 11) && timeOut.endsWith("?:AM|PM")) {

// Invalid input for time-out (morning time with PM)

System.out.println("Invalid time out: " + timeOut);

JOptionPane.showMessageDialog(this, "Invalid time format for time out. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_out\_field.setText(""); // Clear the time-out input field

} else {

// You can add further processing or leave it as is

}

} else {

// Invalid input for time-out

System.out.println("Invalid input for time out: " + timeOut);

JOptionPane.showMessageDialog(this, "Invalid time format for time out. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_out\_field.setText(""); // Clear the time-out input field

}

// Validate date

String userEnteredDate = date\_field.getText();

if (!isValidDateFormat(userEnteredDate) || !isCurrentDate(userEnteredDate)) {

JOptionPane.showMessageDialog(this, "Please enter the current date in the format yyyy-MM-dd", "Incorrect Date", JOptionPane.ERROR\_MESSAGE);

return;

}

// Validate PC\_NUMBER

String pcText = pc\_field.getText().trim();

if (!pcText.matches("^[0-9\\p{Punct}]+$")) {

// Invalid input for PC\_NUMBER

JOptionPane.showMessageDialog(this, "Invalid input for PC\_NUMBER. Please enter only numeric characters.", "Error", JOptionPane.ERROR\_MESSAGE);

pc\_field.setText(""); // Clear the pc\_field input field

return;

}

// Validate SUBJECT

String subjectText = subject\_field.getText().trim();

if (!subjectText.matches("^[a-zA-Z0-9\\p{Punct}\\s]+$")) {

// Invalid input for PURPOSE

JOptionPane.showMessageDialog(this, "Invalid input for SUBJECT. Please enter only letters and special characters.", "Error", JOptionPane.ERROR\_MESSAGE);

subject\_field.setText(""); // Clear the purpose\_field input field

return;

}

String existingLabel6Text = jLabel6.getText();

try {

// Create the SQL query with placeholders

String insertQuery = "INSERT INTO portal (POSITION, ID\_NUMBER, FULL\_NAME, TIME\_IN, TIME\_OUT, DATE, PC\_NUMBER, SUBJECT, UNI\_CODE, FACULTY\_USER) VALUES (?, ?, ?, ?, ?, ?, ? , ?, ?, ?)";

// Create a connection

ConnectionProvider dbc = new ConnectionProvider();

String jdbcDriver = dbc.getJdbcDriver();

String dbConnectionURL = dbc.getDbConnectionURL();

String dbUsername = dbc.getDbUsername();

String dbPassword = dbc.getDbPassword();

Class.forName(jdbcDriver);

Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);

// Create the PreparedStatement

PreparedStatement statement = connection.prepareStatement(insertQuery);

// Assuming student\_id\_field represents a string student ID

statement.setString(1, String.valueOf(position\_combo\_box.getSelectedItem()));

statement.setString(2, id\_number\_field.getText());

statement.setString(3, full\_name\_field.getText());

statement.setString(4, time\_in\_field.getText()); // Corrected line

statement.setString(5, time\_out\_field.getText());

statement.setString(6, date\_field.getText());

statement.setString(7, pc\_field.getText());

statement.setString(8, subject\_field.getText());

statement.setString(9, uni\_field.getText());

statement.setString(10, jLabel6.getText());

statement.executeUpdate();

// Clear the input fields

this.position\_combo\_box.setSelectedItem("FACULTY");

this.id\_number\_field.setText("");

this.full\_name\_field.setText("");

this.time\_in\_field.setText("");

this.time\_out\_field.setText("");

this.date\_field.setText("");

this.pc\_field.setText("");

this.subject\_field.setText("");

jLabel6.setText(this.USERNAME);

// Set the text of jLabel6 back to its original content

jLabel6.setText(existingLabel6Text);

// Display a success message

JOptionPane.showMessageDialog(null, "Added Successful!", "Success", JOptionPane.INFORMATION\_MESSAGE);

DefaultTableModel tableModel = new DefaultTableModel();

jTable1.setModel(tableModel);

showportal();

System.out.println("Add Successfully!");

} catch (ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

uni();

}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

DefaultTableModel RecordTable = (DefaultTableModel) jTable1.getModel();

// Check if the table is editable, and show a message if not

if (!isTableEditable(RecordTable)) {

JOptionPane.showMessageDialog(this, "TABLE IS NOT EDITED", "Warning", JOptionPane.WARNING\_MESSAGE);

return;

}

int SelectedRows = jTable1.getSelectedRow();

position\_combo\_box.setSelectedItem(RecordTable.getValueAt(SelectedRows, 0).toString());

id\_number\_field.setText(RecordTable.getValueAt(SelectedRows, 1).toString());

full\_name\_field.setText(RecordTable.getValueAt(SelectedRows, 2).toString());

time\_in\_field.setText(RecordTable.getValueAt(SelectedRows, 3).toString());

time\_out\_field.setText(RecordTable.getValueAt(SelectedRows, 4).toString());

date\_field.setText(RecordTable.getValueAt(SelectedRows, 5).toString());

pc\_field.setText(RecordTable.getValueAt(SelectedRows, 6).toString());

subject\_field.setText(RecordTable.getValueAt(SelectedRows, 7).toString());

uni\_field.setText(RecordTable.getValueAt(SelectedRows, 8).toString());

jLabel6.setText(RecordTable.getValueAt(SelectedRows, 9).toString());

}

// Function to check if the table is editable

private boolean isTableEditable(DefaultTableModel tableModel) {

return tableModel.isCellEditable(0, 0); // Change the indices based on your requirements

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try {

// Create the SQL query for update

String updateQuery = "UPDATE portal SET TIME\_OUT=? WHERE ID\_NUMBER=?";

// Trim input to remove leading/trailing whitespaces

String idNumber = id\_number\_field.getText().trim();

// Create a connection

ConnectionProvider dbc = new ConnectionProvider();

String jdbcDriver = dbc.getJdbcDriver();

String dbConnectionURL = dbc.getDbConnectionURL();

String dbUsername = dbc.getDbUsername();

String dbPassword = dbc.getDbPassword();

Class.forName(jdbcDriver);

Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);

// Use PreparedStatement

try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {

preparedStatement.setString(1, time\_out\_field.getText());

preparedStatement.setString(2, idNumber);

// Execute the UPDATE query

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

// Clear the input fields

this.position\_combo\_box.setSelectedItem("FACULTY");

this.id\_number\_field.setText("");

this.full\_name\_field.setText("");

this.time\_in\_field.setText("");

this.date\_field.setText("");

this.pc\_field.setText("");

this.subject\_field.setText("");

this.uni\_field.setText("");

this.jLabel6.setText("");

JOptionPane.showMessageDialog(this, "Record has been successfully updated");

} else {

JOptionPane.showMessageDialog(this, "No record found with the given ID\_NUMBER.");

}

// Refresh the table with updated data

DefaultTableModel tableModel = new DefaultTableModel();

jTable1.setModel(tableModel);

showportal();

System.out.println("Update Successfully!");

} catch (SQLException e) {

e.printStackTrace();

} finally {

// Close the connection

if (connection != null) {

connection.close();

}

}

} catch (ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

uni();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try {

// Create the SQL query with placeholders

String searchQuery = "SELECT \* FROM portal WHERE ID\_NUMBER LIKE ? OR POSITION LIKE ? OR TIME\_IN LIKE ? OR TIME\_OUT LIKE ? OR DATE LIKE ? OR PC\_NUMBER LIKE ? OR SUBJECT LIKE ? OR UNI\_CODE LIKE ? OR FACULTY\_USER LIKE ?";

// Create a connection

ConnectionProvider dbc = new ConnectionProvider();

String jdbcDriver = dbc.getJdbcDriver();

String dbConnectionURL = dbc.getDbConnectionURL();

String dbUsername = dbc.getDbUsername();

String dbPassword = dbc.getDbPassword();

Class.forName(jdbcDriver);

Connection connection = DriverManager.getConnection(dbConnectionURL, dbUsername, dbPassword);

// Create the PreparedStatement

PreparedStatement statement = connection.prepareStatement(searchQuery);

// Set search parameters

String searchKeyword = "%" + searchTF.getText() + "%";

for (int i = 1; i <= 9; i++) {

statement.setString(i, searchKeyword);

}

// Execute the query

ResultSet resultSet = statement.executeQuery();

// Process the result set as needed

if (resultSet.next()) {

this.POSITION = resultSet.getString("POSITION");

this.FULL\_NAME = resultSet.getString("FULL\_NAME");

this.TIME\_IN = resultSet.getString("TIME\_IN");

this.TIME\_OUT = resultSet.getString("TIME\_OUT");

this.DATE = resultSet.getString("DATE");

this.PC\_NUMBER = resultSet.getString("PC\_NUMBER");

this.SUBJECT = resultSet.getString("SUBJECT");

this.UNI\_CODE = resultSet.getString("UNI\_CODE");

this.ID\_NUMBER = resultSet.getString("ID\_NUMBER");

this.USERNAME = resultSet.getString("FACULTY\_USER");

// Display the retrieved data

System.out.println("POSITION: " + this.POSITION);

System.out.println("FULL\_NAME: " + this.FULL\_NAME);

System.out.println("TIME\_IN: " + this.TIME\_IN);

System.out.println("TIME\_OUT: " + this.TIME\_OUT);

System.out.println("DATE: " + this.DATE);

System.out.println("PC\_NUMBER: " + this.PC\_NUMBER);

System.out.println("SUBJECT: " + this.SUBJECT);

System.out.println("UNI\_CODE: " + this.UNI\_CODE);

System.out.println("ID\_NUMBER: " + this.ID\_NUMBER);

System.out.println("FACULTY\_USER: " + this.USERNAME);

jLabel6.setText(this.USERNAME);

// Create a table model to store data

DefaultTableModel tableModel = new DefaultTableModel();

jTable1.setModel(tableModel);

// Get column names and add them to the table model

ResultSetMetaData metaData = resultSet.getMetaData();

int columnCount = metaData.getColumnCount();

for (int i = 1; i <= columnCount; i++) {

String columnName = metaData.getColumnName(i);

if (columnName.equals("POSITION") || columnName.equals("FULL\_NAME") ||

columnName.equals("TIME\_IN") || columnName.equals("TIME\_OUT") || columnName.equals("DATE")

|| columnName.equals("PC\_NUMBER") || columnName.equals("SUBJECT")|| columnName.equals("UNI\_CODE")|| columnName.equals("FACULTY\_USER") || columnName.equals("ID\_NUMBER")) {

tableModel.addColumn(columnName);

}

}

// Add rows to the table model

do {

Object[] rowData = new Object[columnCount];

for (int i = 1; i <= columnCount; i++) {

rowData[i - 1] = resultSet.getObject(i);

}

tableModel.addRow(rowData);

} while (resultSet.next());

} else {

System.out.println("No matching records found.");

JOptionPane.showMessageDialog(null, "No record found!");

}

// Close resources

resultSet.close();

statement.close();

connection.close();

} catch (ClassNotFoundException | SQLException e) {

e.printStackTrace();

}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Login log = new Login();

log.setVisible(true);

log.pack();

log.setLocationRelativeTo(null);

this.dispose();

}

private void id\_number\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// TODO add your handling code here:

String input = id\_number\_field.getText().trim(); // Get the input text and remove leading/trailing spaces

// Define a regular expression pattern for the allowed format

String pattern = "^\\d{1,}(-\\d+)\*$";

// Check if the ID has been set

if (!idSet) {

// If not set, check the validity of the input

if (input.matches(pattern)) {

System.out.println("Valid input: " + input);

idSet = true; // Set the flag to true, indicating that ID has been set

id\_number\_field.setEditable(false); // Make the field non-editable

} else {

System.out.println("Invalid input: " + input);

JOptionPane.showMessageDialog(this, "Invalid ID format. Please enter a valid ID number in the specified format.", "Error", JOptionPane.ERROR\_MESSAGE);

id\_number\_field.setText(""); // Clear the input field

}

} else {

// If ID has already been set, show an error message

System.out.println("Attempted to update ID: " + input);

JOptionPane.showMessageDialog(this, "ID number cannot be updated.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void full\_name\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String fullName = full\_name\_field.getText().trim(); // Get the input text and remove leading/trailing spaces

// Define a regular expression pattern for the allowed format

String pattern = "^[a-zA-Z\\s.]+$";

// Check if the input matches the pattern

if (fullName.matches(pattern)) {

// Valid input

System.out.println("Valid input: " + fullName);

// You can add further processing or leave it as is

} else {

// Invalid input

System.out.println("Invalid input: " + fullName);

// Display an error message or handle it as per your requirements

JOptionPane.showMessageDialog(this, "Invalid name format. Please enter a valid name.", "Error", JOptionPane.ERROR\_MESSAGE);

full\_name\_field.setText(""); // Clear the input field

}

}

private void pc\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String inputText = pc\_field.getText().trim();

// Validate that the input contains only numeric characters

if (!inputText.matches("^[0-9\\p{Punct}]+$")) {

// Invalid input (contains non-numeric characters)

JOptionPane.showMessageDialog(this, "Invalid input. Please enter only numeric characters.", "Error", JOptionPane.ERROR\_MESSAGE);

pc\_field.setText(""); // Clear the input field

} else {

// Valid input (contains only numeric characters)

// You can add further processing or leave it as is

}

}

private void subject\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String inputText = subject\_field.getText().trim();

// Validate that the input contains only special characters and letters (no numbers)

if (!inputText.matches("^[a-zA-Z0-9\\p{Punct}\\s]+$")) {

// Invalid input (contains numbers or other characters)

JOptionPane.showMessageDialog(this, "Invalid input. Please enter only letters and special characters.", "Error", JOptionPane.ERROR\_MESSAGE);

subject\_field.setText(""); // Clear the input field

} else {

// Valid input (contains only letters and special characters)

// You can add further processing or leave it as is

}

}

private void date\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// Get the user-entered date from the text field

String userEnteredDate = date\_field.getText();

// Validate the entered date format

if (!isValidDateFormat(userEnteredDate)) {

JOptionPane.showMessageDialog(this, "Please enter the date in the format yyyy-MM-dd",

"Invalid Date Format", JOptionPane.ERROR\_MESSAGE);

return;

}

// Check if the entered date is the current date

if (!isCurrentDate(userEnteredDate)) {

JOptionPane.showMessageDialog(this, "Please enter the current date",

"Incorrect Date", JOptionPane.ERROR\_MESSAGE);

return;

}

// If the format and date are valid, you can proceed with your logic

// Your code here...

}

private boolean isValidDateFormat(String date) {

try {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

sdf.setLenient(false);

sdf.parse(date);

return true;

} catch (ParseException e) {

return false;

}

}

private boolean isCurrentDate(String date) {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String currentDate = sdf.format(new Date());

return currentDate.equals(date);

}

private void time\_out\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

String timeOut = time\_out\_field.getText().trim(); // Get the input text and remove leading/trailing spaces

// Validate TIME\_OUT only if a value is provided

if (!timeOut.isEmpty()) {

String timeValidationPattern = "^(0[1-9]|1[0-2]):[0-5][0-9]\\s(?:AM|PM)$";

if (!timeOut.matches(timeValidationPattern)) {

// Invalid TIME\_OUT format

JOptionPane.showMessageDialog(this, "Invalid time format. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_out\_field.setText(""); // Clear the input field

return; // Stop execution if the TIME\_OUT is invalid

}

// Parse the hours from the TIME\_OUT input

int hoursOut = Integer.parseInt(timeOut.split(":")[0]);

// Check if the TIME\_OUT is in the morning (AM) or afternoon/evening (PM)

if ((hoursOut >= 1 && hoursOut <= 11) && timeOut.endsWith("?:AM|PM")) {

// Invalid input for TIME\_OUT (morning time with PM)

JOptionPane.showMessageDialog(this, "Invalid time format for TIME\_OUT. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_out\_field.setText(""); // Clear the input field

return;

} // Validate that TIME\_OUT is after TIME\_IN

String timeIn = time\_in\_field.getText().trim();

if (!isTimeAfter(timeIn, timeOut)) {

// Invalid input

JOptionPane.showMessageDialog(this, "Invalid time. TIME\_IN should be ahead of TIME\_OUT.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_out\_field.setText(""); // Clear the input field

return; // Stop execution if TIME\_OUT is not ahead of TIME\_IN

}

}

// If the time\_out\_field is empty, set TIME\_OUT to NULL or an appropriate default value

String timeOutValue = timeOut.isEmpty() ? "NULL" : "?";

}

// Helper method to check if time1 is after time2

private boolean isTimeAfter(String time1, String time2) {

try {

SimpleDateFormat format = new SimpleDateFormat("hh:mm a");

java.util.Date date1 = format.parse(time1);

java.util.Date date2 = format.parse(time2);

return date1.after(date2);

} catch (ParseException e) {

e.printStackTrace();

return false;

}

}

private void time\_in\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

String timeIn = time\_in\_field.getText().trim(); // Get the input text and remove leading/trailing spaces

// Define a regular expression pattern for the allowed format (12-hour clock with AM/PM)

String pattern = "^(0[1-9]|1[0-2]):[0-5][0-9]\\s(?:AM|PM)$";

// Check if the input matches the pattern

if (timeIn.matches(pattern)) {

// Parse the hours from the TIME\_IN input

int hoursIn = Integer.parseInt(timeIn.split(":")[0]);

// Check if the TIME\_IN is in the morning (AM) or afternoon/evening (PM)

if ((hoursIn >= 1 && hoursIn <= 11) && timeIn.endsWith("?:AM|PM")) {

// Invalid input for TIME\_IN (morning time with PM)

System.out.println("Invalid input: " + timeIn);

JOptionPane.showMessageDialog(this, "Invalid time format. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_in\_field.setText(""); // Clear the input field

return;

}

// You can add further processing or leave it as is

} else {

// Invalid input

System.out.println("Invalid input: " + timeIn);

// Display an error message or handle it as per your requirements

JOptionPane.showMessageDialog(this, "Invalid time format. Please enter a valid time in the 00:00 AM/PM format.", "Error", JOptionPane.ERROR\_MESSAGE);

time\_in\_field.setText(""); // Clear the input field

}

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Get the current time

Date currentTime = new Date(System.currentTimeMillis());

SimpleDateFormat timeFormat = new SimpleDateFormat("hh:mm a");

String formattedTime = timeFormat.format(currentTime);

// Set the formatted time to the time\_in\_field

time\_in\_field.setText(formattedTime);

// Add focus listener to time\_in\_field

time\_in\_field.addFocusListener(new java.awt.event.FocusAdapter() {

private boolean errorShown = false;

public void focusGained(java.awt.event.FocusEvent evt) {

// Display an error message only if it hasn't been shown yet

if (!errorShown) {

JOptionPane.showMessageDialog(null, "Time In field is not editable", "Error", JOptionPane.ERROR\_MESSAGE);

errorShown = true;

}

// Reset the focus to the component you want (e.g., jButton7)

jButton7.requestFocusInWindow();

}

});

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

Date currentTime = new Date(System.currentTimeMillis());

SimpleDateFormat timeFormat = new SimpleDateFormat("hh:mm a");

String formattedTime = timeFormat.format(currentTime);

// Set the formatted time to the time\_out\_field

time\_out\_field.setText(formattedTime);

// Add focus listener to time\_out\_field

time\_out\_field.addFocusListener(new java.awt.event.FocusAdapter() {

private boolean errorShown = false;

public void focusGained(java.awt.event.FocusEvent evt) {

// Display an error message only if it hasn't been shown yet

if (!errorShown) {

JOptionPane.showMessageDialog(null, "Time Out field is not editable", "Error", JOptionPane.ERROR\_MESSAGE);

errorShown = true;

}

// Reset the focus to the component you want (e.g., jButton8)

jButton8.requestFocusInWindow();

}

});

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

// Get the current date and time

Date currentDate = new Date(System.currentTimeMillis());

// Format the date using SimpleDateFormat

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

String formattedDate = dateFormat.format(currentDate);

// Set the formatted date to the date\_field

date\_field.setText(formattedDate);

// Make the date\_field not editable

date\_field.setEditable(false);

date\_field.setText(formattedDate);

// Set the formatted date to the date\_field

date\_field.setText(formattedDate);

// Make the date\_field not editable

date\_field.setEditable(false);

// Add focus listener to date\_field

date\_field.addFocusListener(new java.awt.event.FocusAdapter() {

private boolean errorShown = false;

public void focusGained(java.awt.event.FocusEvent evt) {

// Display an error message only if it hasn't been shown yet

if (!errorShown) {

JOptionPane.showMessageDialog(null, "Date field is not editable", "Error", JOptionPane.ERROR\_MESSAGE);

errorShown = true;

}

// Reset the focus to jButton9

jButton9.requestFocusInWindow();

}

});

}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

position\_combo\_box.setSelectedItem("");

this.id\_number\_field.setText("");

this.full\_name\_field.setText("");

this.time\_in\_field.setText("");

this.time\_out\_field.setText("");

this.date\_field.setText("");

this.pc\_field.setText("");

this.subject\_field.setText("");

clearFields();

uni();

}

private void uni\_fieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

uni();

}

private void searchTFActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void position\_combo\_boxActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void clearFields() {

// Clear editable text fields

clearTextField(time\_out\_field);

clearTextField(time\_in\_field);

clearTextField(date\_field);

clearTextField(pc\_field);

clearTextField(full\_name\_field);

clearTextField(id\_number\_field);

clearTextField(subject\_field);

}

private void clearTextField(JTextField textField) {

// Check if the text field is editable before setting its text to an empty string

if (textField.isEditable()) {

textField.setText("");

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(PORTAL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(PORTAL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(PORTAL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PORTAL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

}

});

}

// Variables declaration - do not modify

private javax.swing.JTextField date\_field;

private javax.swing.JTextField full\_name\_field;

private javax.swing.JTextField id\_number\_field;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton10;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JButton jButton6;

private javax.swing.JButton jButton7;

private javax.swing.JButton jButton8;

private javax.swing.JButton jButton9;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

public javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField pc\_field;

private javax.swing.JComboBox<String> position\_combo\_box;

private javax.swing.JTextField searchTF;

private javax.swing.JTextField subject\_field;

private javax.swing.JTextField time\_in\_field;

private javax.swing.JTextField time\_out\_field;

private javax.swing.JTextField uni\_field;

// End of variables declaration }

**CURRICULUM**

**VITAE**



## 

## Personal Information

Name: Herbert Tomi-as Acedre

Contact Number: 09854443162

Email Address: [acedreherbert6417@gmail.com](mailto:acedreherbert6417@gmail.com)

Address: Sitio Romblon, Inayawan, Cebu City, Cebu 6000, Phil.

Sex: Male

Civil Status: Married

Religion: Catholic

Age: 24

Birthday: October 25, 1998

Birthplace: Manlapay Dumanjug, Cebu

Height: 5’6 ft

Weight: 78 kg

Father’s Name: Robert Carbajosa Acedre

Mother’s Name: Hermenia Tomi-as Acedre

**EDUCATIONAL BACKGROUND**

Elementary: Inayawan Elementary School (2005 - 2011)

High School: Pardo National High School (2011 - 2015)

College: Talisay City College (2020 - 2024)



**Personal Information**

Name: Jhuztin Nino S. Obenza

Contact Number: 09334327607

Email Address: [Jhuztinobenza@gmail.com](mailto:Jhuztinobenza@gmail.com)

Address: Deca Homes Tungkil Minglanilla City, Cebu

Sex: Male

Civil Status: Single

Religion: Roman Catholic

Age: 20

Birthday: June 9, 2002

Birthplace: Cebu Maternity

Height: 5’6 ft.

Weight: 54 kg.

Father’s Name: Anonymous

Mother’s Name: Jean Marie Obenza

**EDUCATIONAL BACKGROUND**

Elementary: Lipata Central School (2008-2014)

High School: St. Paul College Foundation Inc. (2014-2018)

Senior High: St. Paul College Foundation Inc. (2018-2020)

College: Talisay City College (2020-2024)



**Personal Information**

Name: Vanessa P. Dayondon

Contact Number: 09224067857

Email Address: [sunshinez143.09978418462@gmail.com](mailto:sunshinez143.09978418462@gmail.com)

Address: Linao Talisay City

Sex: Female

Civil Status: Single

Religion: Roman Catholic

Age: 26

Birthday: January 27, 1997

Birthplace: CCMC

Height: 5’2 ft.

Weight: 48 kg.

Father’s Name: Rolando Secuya dayondon

Mother’s Name: Lucia Gingone Panogalinog

**EDUCATIONAL BACKGROUND**

Elementary: Tabunok Elementary School (2006-2011)

High School: Cansojong National High School (2011-2014)

College: Talisay City College (2015- 2024)



**Personal Information**

Name: Mariel D. Bazar

Contact Number: 09605058946

Email Address: marieldimolbazar18@gmail.com

Address: Tanke, Talisay City Cebu

Sex: Female

Civil Status: Single

Religion: Catholic

Age: 22

Birthday: July 18, 2001

Birthplace: Tanke, Talisay City Cebu

Height: 5’4 ft.

Weight: 60 kg.

Father’s Name: Rogelio Bazar

Mother’s Name: Marilyn Bazar

**EDUCATIONAL BACKGROUND**

Elementary: San Roquez Elementary School (2008-2014)

High School: Cansojong National High School (2014-2018)

College: Talisay City College (2020-2024)



**Personal Information**

Name: Rasbel Perez Sebial

Contact Number: 09562297139

Email Address: rasbelsebial22@gmail.com

Address: Dumlog , Talisay City Cebu

Sex: Male

Civil Status: Single

Religion: Roman Catholic

Age: 21

Birthday: July 2, 2001

Birthplace: Dumlog, Talisay City

Height: 5’6 ft.

Weight: 55 kg.

Father’s Name: Domingo Sebial

Mother’s Name: Lilibeth Sebial

**EDUCATIONAL BACKGROUND**

Elementary: Talisay City Central School (2008-2014)

High School: Talisay City National High School (2014-2018)

College: Talisay City College (2020-2024)

**OPERATIONAL MANUAL**