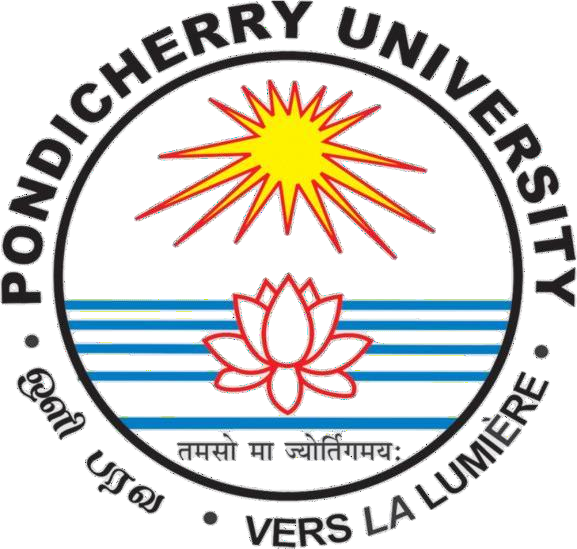
PONDICHERRY UNIVERSITY

**(***A Central University***)**



**SCHOOL OF ENGINEERING AND TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE**

### MASTER OF COMPUTER APPLICATIONS

NAME : MONIKA D

REG. NO. : 22352038

SEMESTER : IV

SUBJECT : PROJECT REPORT

GUIDED BY : Dr. M. SATHYA

# Student Information and Marks Management System

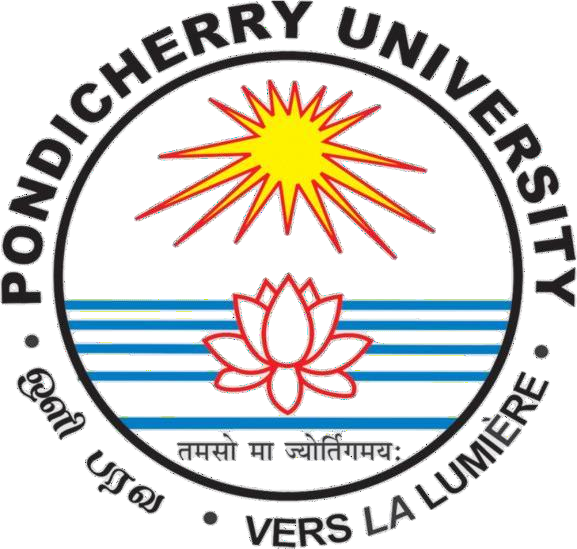
by

**MONIKA D**

(Registration Number: 22352038)

#### Project report submitted in partial fulfillment of the requirements for the award of the degree of

**MASTER OF COMPUTER APPLICATIONS**



## DEPARTMENT OF COMPUTER SCIENCE

## SCHOOL OF ENGINEERING & TECHNOLOGY

## PONDICHERRY UNIVERSITY

## PUDUCHERRY – 605 014

## INDIA

**May 2024**

## BONAFIDE CERTIFICATE

This is to certify that this project work entitled **“Student Information and Marks Management System** is a bonafide record of work done by **Ms. D. Monika** (Register Number: 22352038) in the partial fulfillment for the degree of Master of Computer Applications of Pondicherry University.

This work has not been submitted elsewhere for the award of any other degree to the best of our knowledge.

**INTERNAL GUIDE HEAD OF THE DEPARTMENT**

#### Dr. M. Sathya Dr. S.K.V. Jayakumar

Assistant Professor Professor & Head

Department of Computer Science Department of Computer Science

School of Engineering & Technology School of Engineering & Technology Pondicherry University Pondicherry University

Pondicherry – 605 014 Pondicherry – 605 014

Submitted for the Viva-Voce Examination held on:

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

I am greatly thankful to all those who helped me in making this project a successful one.

I express my heartfelt gratitude to my Project Guide **Dr. M. Sathya, Assistant Professor, Department of Computer Science, Pondicherry University, Pondicherry**, for her whole - hearted assistance and direction not only for the duration of the project but for the entire duration of the course. I will always remain grateful to her and whose constant care about me has provided a new direction to work.

I express my gratitude to **Dr. S. K. V. JAYAKUMAR, Professor and Head, Department of Computer Science, Pondicherry University, Puducherry**, and **Dr. S. Sivasathya, Dean, School of Engineering & Technology, Pondicherry University, Puducherry** for providing us the facilities to work with the project.

Finally, I would like to express my regards for all the faculty members of Department of Computer Science and others involved in this project, directly or indirectly. I am also grateful to my parents for their continuous support and encouragement.

Last but not least, the blessings of **God** enabled me to complete the project successfully.

(**MONIKA D)**

## SYNOPSIS

The Student Information and Academic Management System (SIAMS) project is dedicated to revolutionizing the management of student information and academic records within universities. Its primary objective is to introduce a comprehensive digital platform accessible to university administrators and staff, aiming to streamline data entry, storage, and retrieval processes. SIAMS focuses on several key goals to achieve this aim. Firstly, it seeks to enhance data management by efficiently storing and managing student personal details and academic records. Additionally, the system aims to streamline academic record-keeping by facilitating the management of semester-wise core and additional subject details, including marks, grades, and points. Furthermore, SIAMS prioritizes improving data accessibility and transparency by providing detailed academic performance reports for individual students, ensuring transparency and ease of access. To maintain the integrity and security of the data, robust mechanisms are implemented within the system. Finally, SIAMS aims to facilitate decision-making processes by offering comprehensive and accurate records to aid in academic planning and decision-making. Overall, SIAMS represents a significant advancement in academic administration, leveraging technology to streamline processes and enhance efficiency, transparency, and reliability in managing student information and academic records.

**TABLE OF CONTENTS**

**TITLE PAGE NO.**

**ACKNOWLEDGEMENT**

**ABSTRACT**

1. **INTRODUCTION**
   1. ABOUT THE PROJECT 1
   2. PROJECT PLAN 2
2. **PROBLEM DEFINITION & FEASIBILITY ANALYSIS**
   1. PROBLEM DEFINITION 3
   2. EXISTING SYSTEM 3
   3. PROPOSED SYSTEM 3
   4. FEASIBILITY STUDY
      1. TECHNICAL FEASIBILITY 4
      2. OPERATIONAL FEASIBILITY 4
      3. ECONOMIC FEASIBILITY 4
3. **SOFTWARE REQUIREMENT** 
   1. HARDWARE REQUIREMENTS 5
   2. SOFTWARE REQUIREMENTS 5
   3. SYSTEM REQUIREMENTS 6
4. **SYSTEM DESIGN**
   1. MODULE DESCRIPTION 7
   2. USE CASE DIAGRAM 8
   3. ACTIVITY DIAGRAM 9
   4. SEQUENCE DIAGRAM 10
5. **IMPLEMENTATION** 11
   1. STUDENT INFORMATION ENTRY
   2. SEMESTER SELECTION
   3. HARDCORE SUBJECT ENTRY
   4. SOFTCORE SUBJECT ENTRY
   5. MARKSHEET GENERATION
   6. PDF EXPORT
   7. DATABASE MANAGEMENT
6. **SYSTEM TESTING** 
   1. SYSTEM IMPLEMENTATION 12
   2. TESTING
      1. UNIT TESTING 12
      2. VALIDATION TESTING 13
      3. FUNCTIONAL TESTING 13
7. **CONCLUSION**

APPENDIX I: SCREENSHOTS 13

APPENDIX II: OUTPUTS 14

1. **BIBLIOGRAPHY**

# Student Information and Marks Management System

### Abstract

The Student Information and Academic Management System (SIAMS) is a transformative project aimed at modernizing the management of student information and academic records in university settings. Through the introduction of a comprehensive digital platform accessible to university administrators and staff, SIAMS seeks to streamline data entry, storage, and retrieval processes. Key goals of the project include enhancing data management efficiency, streamlining academic record-keeping, improving data accessibility and transparency, ensuring data integrity and security, and facilitating informed decision-making. By achieving these goals, SIAMS aims to reduce administrative overhead, enhance data accuracy, and improve overall efficiency in managing student academic records, ultimately leading to better academic administration and decision-making.

**1. Introduction**

**1.1 About the project**

The Student Information and Academic Management System (SIAMS) project is a pioneering initiative aimed at revolutionizing the management of student information and academic records within university environments. Recognizing the need for an efficient and transparent system, SIAMS introduces a comprehensive digital platform accessible to university administrators and staff. This platform simplifies the processes of data entry, storage, and retrieval, facilitating seamless management of student details and academic records. SIAMS is designed to achieve several key goals, including enhancing data management efficiency, streamlining academic record-keeping, improving data accessibility and transparency, ensuring data integrity and security, and facilitating informed decision-making. Through robust mechanisms and user-friendly interfaces, SIAMS aims to reduce administrative overhead, enhance data accuracy, and improve overall efficiency in managing student academic records. By providing comprehensive and accurate records, SIAMS empowers administrators to make informed decisions, ultimately leading to better academic management and planning. Overall, SIAMS represents a significant advancement in academic administration, leveraging technology to enhance efficiency, transparency, and reliability in managing student information and academic records.

**1.2 Project Plan**

The project plan for the Student Information and Academic Management System (SIAMS) encompasses several key phases and activities aimed at achieving the project's goals effectively and efficiently. The plan typically includes the following components:

**1. Initiation Phase**: This phase involves defining the project scope, objectives, and stakeholders. Key activities include conducting a feasibility study, establishing project governance, and securing necessary approvals and resources.

**2. Requirements Gathering**: In this phase, the project team collaborates with university stakeholders to gather detailed requirements for the SIAMS platform. This includes identifying user needs, functional requirements, and technical specifications.

**3. System Design**: Based on the requirements gathered, the project team develops a comprehensive system design for SIAMS. This includes designing the user interface, database structure, system architecture, and integration points with existing university systems.

**4. Development and Testing**: The development phase involves building the SIAMS platform according to the approved design. Concurrently, rigorous testing is conducted to ensure that the system meets quality standards and functional requirements. This includes unit testing, integration testing, and user acceptance testing.

**5. Implementation and Deployment**: Once development and testing are complete, the SIAMS platform is deployed in a phased approach. This involves installing the system, configuring settings, migrating data from legacy systems, and training users.

**6. Rollout and Adoption**: The SIAMS platform is gradually rolled out to different departments and campuses within the university. Change management strategies are implemented to promote user adoption and ensure a smooth transition to the new system.

**7. Monitoring and Evaluation**: After deployment, the project team monitors the performance and usage of the SIAMS platform. Feedback is collected from users, and system performance is evaluated against key metrics to identify areas for improvement.

**8. Maintenance and Support**: Ongoing maintenance and support activities are conducted to ensure the continued functionality and reliability of the SIAMS platform. This includes applying updates, addressing user issues, and providing technical assistance as needed.

Throughout the project lifecycle, effective communication, stakeholder engagement, and project management practices are essential to ensure the successful delivery of the SIAMS platform on time and within budget. Regular progress reporting and milestone reviews help track project progress and address any issues or risks that may arise.

### Problem Defintion And Feasibility Analysis

### 

### 2.1 Problem Definition

Educational institutions often rely on manual and paper-based systems to manage student grades and academic information. This method is not only time-consuming and labor-intensive but also prone to errors and inconsistencies. The process involves manually entering student grades, maintaining physical records, and generating reports and transcripts by hand, which can lead to significant inefficiencies. The manual system poses challenges in data retrieval, accessibility, and accuracy, impacting both administrative staff and students. There is a pressing need for a streamlined, automated solution to enhance the efficiency and accuracy of managing student grades and academic records.

**2.2 Existing System**

The current system employed by many educational institutions involves manual data entry and paper-based record-keeping for student grades and academic information. Administrative staff must enter student details and grades into physical registers or spreadsheets, which are then stored in filing cabinets or computer files with limited access. This approach leads to several issues: the risk of human error during data entry, difficulties in maintaining and updating records, challenges in retrieving specific student information promptly, and limited accessibility for authorized personnel. Generating reports and transcripts is a tedious task, often requiring extensive manual effort and time. The existing system is not scalable, lacks real-time accessibility, and provides no automated way to ensure data accuracy and consistency.

**2.3 Proposed System**

The proposed Student Grades Management System aims to replace the manual, paper-based processes with an automated, web-based solution. This system will allow for efficient data entry, storage, and retrieval of student grades and academic information. Key features include a user-friendly interface for entering student details and grades, validation checks to ensure data accuracy, and a structured database for secure and organized data storage. The system will enable quick and easy retrieval of student records and generate automated reports and transcripts, which can be printed or exported as PDF documents. Additionally, it will provide remote access for authorized personnel, enhancing accessibility and collaboration. The proposed solution is designed to be scalable, ensuring it can handle increasing volumes of data over time, and will significantly reduce the time and effort required for managing student grades while improving accuracy and reliability.

**2.4 Feasibility Study**

The feasibility study for implementing the Student Grades Management System involves assessing various aspects to determine the viability and potential success of the project. The study encompasses technical, economic, operational, and schedule feasibility considerations.

**2.4.1. Technical Feasibility**

The technical feasibility assesses whether the proposed system can be developed with the current technology and within the existing technical infrastructure. The system will be developed using widely used web technologies such as PHP, MySQL, JavaScript, HTML, and CSS. These technologies are robust, well-documented, and supported by a large community of developers, ensuring that any technical issues can be resolved efficiently. The system will be hosted on a local server using XAMPP, a reliable and easy-to-configure solution for local development. Given the availability of skilled developers and the maturity of the chosen technologies, the project is technically feasible.

**2.4.2. Operational Feasibility**

Operational feasibility focuses on whether the system will function effectively within the organization and meet user needs. The proposed system is designed to be user-friendly, with an intuitive interface that requires minimal training for users to become proficient. The system will streamline data entry, storage, and retrieval processes, significantly improving operational efficiency. It will also provide secure access to data for authorized personnel, ensuring that student information is protected. The ability to generate automated reports and transcripts will save time and reduce errors, enhancing the overall effectiveness of the institution's administrative processes. Feedback from potential users indicates strong support for the system, confirming its operational feasibility.

**2.4.3 Economic Feasibility**

The technical feasibility assesses whether the proposed system can be developed with the current technology and within the existing technical infrastructure. The system will be developed using widely used web technologies such as PHP, MySQL, JavaScript, HTML, and CSS. These technologies are robust, well-documented, and supported by a large community of developers, ensuring that any technical issues can be resolved efficiently. The system will be hosted on a local server using XAMPP, a reliable and easy-to-configure solution for local development. Given the availability of skilled developers and the maturity of the chosen technologies, the project is technically feasible.

1. **Software Requirement Specification**

**3.1 Hardware Requirements:**

1. Server:

- Processor: Intel Core i5 or equivalent

- RAM: Minimum 8GB

- Storage: Minimum 500GB HDD/SSD

- Network Interface: Ethernet or Wi-Fi for internet connectivity

- Operating System: Linux or Windows Server

2. Client Devices:

- Desktop/Laptop:

- Processor: Intel Core i3 or equivalent

- RAM: Minimum 4GB

- Storage: Minimum 250GB HDD/SSD

- Network Interface: Ethernet or Wi-Fi for internet connectivity

- Operating System: Windows 10, macOS, or Linux

- Mobile Devices (Optional):

- Smartphones/Tablets with Android or iOS operating systems

- Wi-Fi or mobile data connectivity

**3.2 Software Requirements:**

1. Server-Side:

- Web Server: Apache, Nginx, or Microsoft IIS

- Database Management System: MySQL, PostgreSQL, or MongoDB

- Programming Languages: PHP or Node.js for server-side scripting

- Frameworks: Laravel, Django, or Express.js for web application development

- Operating System: Linux (e.g., Ubuntu Server) or Windows Server

2. Client-Side

- Web Browser: Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge

- JavaScript Framework: React.js, Angular, or Vue.js for client-side interactivity

- HTML5, CSS3: for front-end design and layout

- Operating System: Windows, macOS, Linux, Android, or iOS

* 1. **System Requirements**
     1. **Functional Requirements**
* Student Management: Enter, modify, and view student details.
* Course and Subject Management: Manage core and additional subjects for different semesters.
* Grades Management: Record, edit, delete, and view grades for core and additional subjects.
* Report Generation: Generate and export grade statements as PDF documents.

Search and Filter: Search for students and filter records based on various parameters.

**3.3.2** **Non-Functional Requirements**

* **Performance**: Ensure fast response times and high system availability.
* **Security**: Implement authentication, authorization, and data encryption mechanisms.
* **Usability**: Design an intuitive and user-friendly interface for easy navigation.
* **Reliability**: Ensure system reliability and data integrity through backups and error handling.
* **Compatibility**: Support compatibility with various devices, browsers, and operating systems.

**3.3.3. System Integration**

* Integration with university databases for user authentication and data synchronization.
* Integration with email or SMS services for automated notifications.
* Integration with third-party APIs for additional functionalities (optional).

By meeting these hardware, software, and system requirements, the Student Information and Marks Management System (SIAMS) can be effectively deployed and operated to streamline the management of student grades within the educational institution.

1. **System Design**

**4.1 Module Description**

**1. Student Management Module**

- **Description**: This module facilitates the management of student records and details. It allows administrators to add, edit, and delete student information.

- **Features**:

- Add/Edit/Delete student details (name, register number, date of birth, session, programme, specialization).

- View and search student records.

- Import/export student data.

**2. Grade Entry Module:**

- **Description**: This module enables faculty members to enter and update grades for core and additional subjects. It ensures accuracy and efficiency in recording student performance.

**- Features:**

- Enter and update grades for core subjects.

- Enter and update grades for additional subjects.

- Validation of grade entries.

**3. Grade Reporting Module:**

**- Description**: This module generates grade reports for students based on their performance in core and additional subjects. It provides a summary of grades in a printable format.

**- Features:**

- Generate grade reports for individual students.

- Download grade reports in PDF format.

- Print grade reports.

**4.2 Use Case Diagram**

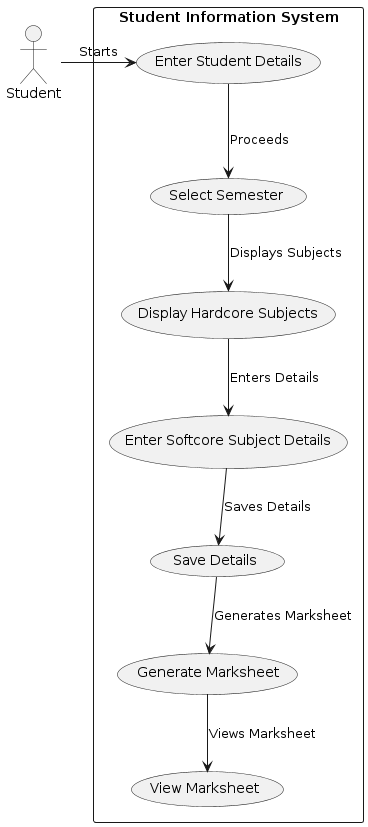


FIGURE 4.1 : Use Case Diagram

**Use case diagram:** The Student Information System is designed to streamline the process of managing student academic details and generating marksheets. The system begins with the student entering their personal details, such as name, registration number, date of birth, session, programme, and specialization. Once the student details are entered, the student selects their current semester, which triggers the display of hardcore subjects specific to that semester. The student then proceeds to enter the details for any softcore subjects they have opted for, including grades, marks, and points. These details are subsequently saved into the system. After all relevant data is entered and saved, the system generates a comprehensive marksheet for the student, which includes both hardcore and softcore subjects along with their respective grades and marks. Finally, the student can view and, if needed, print the generated marksheet. This streamlined process ensures accurate and efficient handling of student academic records, facilitating easy access and management of academic information.

**4.3 Activity Diagram**

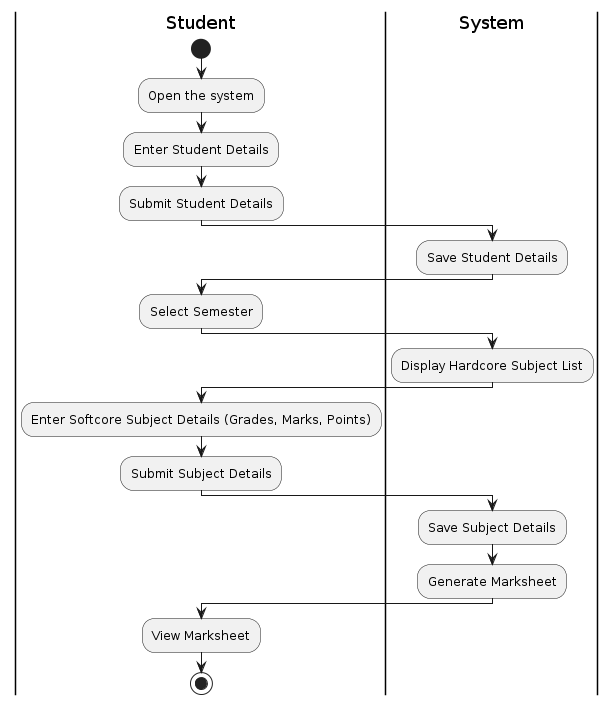
****

FIGURE 4.2 : Activity Diagram

**Activity Diagram:** The system process begins when a student accesses the application. Initially, the student enters their personal details, such as name, registration number, date of birth, session, program, and specialization. Upon submitting this information, the system saves the details into the database. The student then selects the relevant semester from a list. Based on the chosen semester, the system displays a predefined list of hardcore subjects. The student proceeds to enter details for softcore subjects, including grades, marks, and points. After submitting these subject details, the system saves the information. Finally, the system generates a comprehensive marksheet that includes both hardcore and softcore subjects along with their respective grades, marks, and points. The student can view this generated marksheet, providing a complete overview of their academic performance for the selected semester. This streamlined process ensures efficient data entry and accurate marksheet generation.

**4.4 Sequence Diagram**

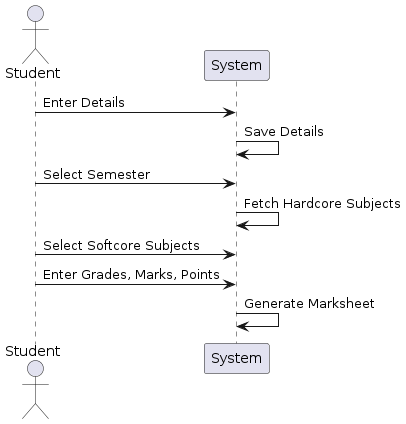
****

FIGURE 4.3 : Sequence Diagram

**Sequence Diagram**: The sequence diagram outlines the interaction between the student and the system within the academic grading system. Initially, the student enters their personal details, including their name, registration number, and other pertinent information. Following this, they select the semester for which they wish to view their grades. Upon selection, the system retrieves a list of core subjects associated with the chosen semester. Subsequently, the student selects their preferred softcore subjects from the available options. After finalizing the subject selection, the student inputs their grades, marks, and points for each chosen subject. Once all necessary information is provided, the system proceeds to generate the marksheet, incorporating the entered data.

1. **Implementation**

**5. 1. Student Information Entry**

**- Description :** This module allows students to input their personal details and academic information, such as name, registration number, date of birth, session, program, and specialization.

**- Implementation :**

- Develop a user-friendly form using HTML and CSS.

- Use PHP to handle form submission and validation.

- Insert the collected data into the database using SQL queries.

**5.2. Semester Selection**

**- Description**: This module enables students to select their current semester from a predefined list.

**- Implementation:**

- Create a dropdown menu with semester options using HTML.

- Use JavaScript to handle user selection and display relevant information.

- Ensure the selected semester is stored and processed correctly for further operations.

**5.3. Hardcore Subject Entry**

**- Description:** Displays a list of predefined hardcore subjects for the selected semester and allows students to view and confirm them.

**- Implementation:**

- Use PHP to fetch hardcore subjects from the database based on the selected semester.

- Display the subjects in a table format using HTML and CSS.

- Store and manage the list of hardcore subjects in the database.

**5.4. Softcore Subject Entry**

**- Description:** Allows students to select additional softcore subjects and input grades, marks, and points for these subjects.

**- Implementation:**

- Provide a form for students to select and enter details for softcore subjects.

- Use JavaScript for dynamic form handling, allowing multiple subjects to be added.

- Validate and insert the entered data into the database using PHP and SQL.

**5.5. Marksheet Generation**

**- Description:** Generates and displays a marksheet based on the entered hardcore and softcore subjects, along with their respective grades, marks, and points.

**- Implementation:**

- Use PHP to retrieve and compile the entered academic data from the database.

- Generate an HTML table to display the marksheet.

- Incorporate CSS for styling and formatting the marksheet for clarity and readability.

**5.6. PDF Export**

**- Description:** Provides functionality to export the generated marksheet as a PDF document.

**- Implementation:**

- Integrate TCPDF library to convert the HTML marksheet to PDF.

- Implement a "Print PDF" button that triggers the PDF generation process.

- Use PHP to handle the PDF creation and download functionality.

**5.7. Database Management**

**- Description:** Manages all database interactions, including inserting, updating, and retrieving student and academic data.

**- Implementation:**

- Design and implement database schemas for storing student information, subject details, and grades.

- Write SQL queries to handle data manipulation tasks.

- Ensure database security and data integrity through proper validation and error handling mechanisms.

By implementing these modules, the academic grading system will efficiently handle student data entry, semester selection, subject management, and marksheet generation, ultimately providing a seamless and user-friendly experience for students and administrators.

**6. System Testing**

* 1. **System Implementation**

System implementation testing evaluates the overall functionality and performance of the PCCS system in its deployed environment.

It involves testing the system as a whole to ensure that all modules and components work together as intended and meet the specified requirements.

Test cases cover end-to-end scenarios, user workflows, system interactions, and system performance under normal and peak load conditions.

**6.2 Testing**

**6.2.1 Unit Testing**

Unit testing focuses on verifying the functionality of individual components in isolation to ensure they work correctly. This involves testing form validation functions to ensure that user inputs are correctly validated, checking the CRUD operations for the database to ensure data is correctly created, read, updated, and deleted, validating the logic that displays subjects based on the selected semester, and ensuring that the functions calculating grades, marks, and points work as expected. By isolating each component, any issues can be identified and resolved early in the development process.

**6.2.2 Validation Testing**

Validation testing is aimed at ensuring that all user inputs meet the specified requirements and constraints. This involves verifying that the student details form correctly validates inputs such as names, registration numbers, and dates of birth. It also includes validating the entry of subject details, ensuring that grades, marks, and points are within acceptable ranges and correctly formatted. Additionally, validation testing ensures that the semester selection process is accurate and that all boundary cases are handled appropriately, preventing invalid data from being processed by the system.

**6.2.3 Functional Testing**

Functional testing is conducted to verify that the system's functionalities work as specified in the requirements. This includes ensuring that the system can capture and store student information accurately, display the correct list of hardcore subjects for the selected semester, and allow users to add, edit, or delete softcore subjects along with their corresponding grades, marks, and points. It also involves generating an accurate marksheet based on the entered data and ensuring that it can be exported as a PDF file. Furthermore, functional testing verifies that the system maintains data persistence, ensuring that information is not lost across sessions and that all functionalities operate seamlessly together.

1. **Conclusion**

In conclusion, the PU Civil Complaints System (PCCS) represents a significant advancement in the management of maintenance-related complaints within Pondicherry University premises. By leveraging technology to streamline processes, enhance communication, and improve accountability, the PCCS aims to address the challenges associated with maintenance management and create a more efficient and transparent system for users and administrative staff alike.

Through the implementation of various modules such as user management, complaint submission, tracking, reporting, and administration, the PCCS provides a comprehensive solution for addressing maintenance needs effectively. The system enables users to submit complaints easily, track their status, and receive timely updates on resolution progress. Meanwhile, administrative staff can prioritize and assign complaints efficiently, monitor resolution efforts, and generate reports to inform decision-making and resource allocation.

Furthermore, the rigorous testing procedures, including unit testing, integration testing, system implementation testing, validation testing, and functional testing, ensure the reliability, usability, and performance of the PCCS system. By validating the system against user requirements, acceptance criteria, and business objectives, the PCCS aims to deliver a solution that meets the needs and expectations of stakeholders and enhances the overall experience for users within Pondicherry University.

In summary, the PU Civil Complaints System (PCCS) project endeavors to optimize the management of maintenance-related complaints, foster transparency and accountability, and ultimately contribute to creating a safer, more functional, and aesthetically pleasing campus environment for all members of the university community.