**“SmaRM: TUPM-COS Classroom Management System”**

Technological University of the Philippines Manila

College of Science

In partial fulfillment of the

Requirement for Software Engineering CS361 & CS362

**By**

*Glenn R. Galbadores I*

Kurt Angelo P. Ballarta

Justine B. Mantilla

John Walter D. Marquez

Ken Zedrick E. Montano

Bachelor of Science in Computer Science - 3B

**May 2025**

**ABSTRACT**

This study focuses on the development and evaluation of the SmaRM (Smart Room Management) system for the College of Science at the Technological University of the Philippines - Manila. Due to ongoing renovations and limited room availability, room conflicts are common within the department. This research aims to address this issue by creating an intelligent room scheduling system with real-time updates, drag-and-drop features, and hierarchical user access. The system was developed using HTML, CSS, JavaScript, PHP, and SQL, and evaluated using ISO 25010 standards. Through functionality, usability, and reliability testing, SmaRM showed promising results in optimizing room utilization, reducing conflicts, and improving scheduling efficiency. The results indicate that SmaRM can serve as a model system for smart educational space management.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES**  Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 704  Email: [graduate@tup.edu.ph](mailto:graduate@tup.edu.ph) | Website: [www.tup.edu.ph](http://www.tup.edu.ph/) | Index No. | F-OGP-3.8-TPF |
| Issue No. | 01 |
| Revision No. | 00 |
|  | Software Engineering | Date | 11242017 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | | **Page** |
| Title Page | | **i** |
| Abstract | | **ii** |
| Table of Contents | | **iii** |
| List of Tables | | **v** |
| List of Figures | | **vi** |
| List of Appendices | | **vii** |
| **Chapter 1** | **THE PROBLEM AND ITS SETTING** |  |
|  | Introduction | **1** |
|  | Background of the Study | **2** |
|  | Objectives of the Study | **3** |
|  | Scope and Delimitations of the Study | **5** |
|  | Significance of the Study | **6** |
| **Chapter 2** | **CONCEPTUAL FRAMEWORK** |  |
|  | Review of Related Literature and Studies | **9** |
|  | Conceptual Model of the Study | **17** |
|  | Operational Definition of Terms | **19** |
| **Chapter 3** | **METHODOLOGY** |  |
|  | Project Design | **21** |
|  | Project Development | **22** |
|  | Operation and Testing Procedure | **23** |
|  | Evaluation Procedure | **24** |
| **Chapter 4** | **RESULTS AND DISCUSSION** |  |
|  | Project Description and Structure | **25** |
|  | Test Results | **31** |
|  | Project Capabilities and Limitations  Evaluation Results | **35**  **36** |
| **Chapter 5** | **SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS** |  |
|  | Summary of Findings | **38** |
|  | Conclusions | **39** |
|  | Recommendations | **40** |
| **REFERENCES** |  | **41** |
| **APPENDIXES** |  | **44** |
| **RESEARCHER’S** | **PROFILE** | **46** |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

**Table Title Page**

1 SmaRM Module Process 22

2 SmaRM Testing Procedure for Functionality, Usability, and Reliability 23

3 Evaluation Criteria 31

4 Evaluation Ratings 32

5 Rating Scale 33

6 System Evaluation 34

7 Survey Evaluation Results36

**LIST OF FIGURES**

**Figure Title Page**

1 SmaRM Development Architecture 17

2 Database Entity Relationship Diagram (ER) 18

3 SmaRM Model System Flowchart 21

4 SmaRM Homepage/ Landing Page 25

5 SmaRM Register Page 26

6 Dashboard for Teachers 26

7 Class Name Declaration 27

8 Drag and Drop Scheduler Feature 27

9 Drag and Drop Scheduler Feature 27

9 Class Scheduler Request Modal 28

10 Admin Dashboard 28

11 Admin Schedule Approval 29

12 Room Schedule List 29

13 Schedule Checking on Homepage 30

14 SQL Database 30

**LIST OF APPENDICES**

**Appendix Title Page**

A Evaluation Form 44

B Evaluation Form Results 45

C Gantt Chart 45

**Chapter 1**

**THE PROBLEM AND ITS SETTINGS**

**INTRODUCTION**

The Room Management sector in COS is more in need of a proper scheduling system. When a proper scheduling system is implemented in the right way, valuable information can be used as the result of that management system and that can help other departments and especially professors and staff in managing the available rooms for the day.

Room conflicts in the COS department happen almost every day at any given time, it causes significant effect on the students and professors like stress, The COS department can make an effective decision by making a room management system. By extracting the results and relationships of past room management systems. Algorithms focused on dealing with space efficiency can give a solution to this situation.

Management of Rooms by looking at the available and already taken rooms for the current day or the following days is an integral part of the COS department. In this research, the researchers tried to accurately manage the rooms by looking at the availability through the database. The researchers used and tested different algorithms for this purpose. A room management system uses a database and can have a large potential in the entire TUP departments in the future.

**BACKGROUND OF THE STUDY**

Room scheduling issues plague the College of Science (COS) at TUP Manila due to ongoing renovations and space limitations. According to Santos (2021), mismanagement of classroom scheduling often leads to conflicts, reduced class time, and stressed faculty. Professors have been forced into last-minute online sessions, undermining the quality of instruction.

Such disruptions can significantly impact the quality of teaching and learning especially professors who may struggle to adapt their teaching methods to an online modality, and students may miss out on the benefits of face-to-face interaction, like immediate feedback and collaborative learning. Previous studies have shown the effects of poor scheduling and modality shifts on academic performance and student engagement, emphasizing the importance of effective classroom management to maintain educational quality.

To resolve this, the SmaRM system introduces a digital, real-time room scheduling platform. It includes drag-and-drop scheduling, role-based access, and a centralized interface. Studies by Torres and Rivera (2020) support the importance of smart scheduling tools in educational institutions, while Lim and Reyes (2022) highlight that online modality shifts often negatively affect academic continuity.

This study aims to address these challenges by examining the current room scheduling practices within the COS. It seeks to identify solutions to minimize scheduling conflicts, optimize the use of available resources, and enhance the overall learning experience for both professors and students.

**OBJECTIVES OF THE STUDY**

**General Objective**

The general objective of the study is to develop the SmaRM system for optimizing classroom scheduling and utilization in the COS department.

**Specifically, the study aims to:**

1. Design the system with the following features:
   1. Account Hierarchy
   2. Team Management
   3. Real-Time Updates
   4. Drag and Drop Scheduling
2. Create the system with the following tools:
   1. Hardware Components:
      1. Desktop or Laptop with the following minimum specifications:

* Windows 10 or above
* Hard Disk with at least 10gb Capacity
* 1.5 GHz Dual Core or Quad Core processor
* At least 4 Gigabytes of RAM
* Keyboard and Mouse as the input device
* Monitor as the output device
  1. Software Components:
* HTML
* CSS
* JavaScript
* PHP
* Web Browser
* Windows 10 or above

1. Test and improve the system regarding functionality and reliability.
2. Determine the level of acceptability of the system using ISO 25010.
3. Measure user satisfaction and system acceptability

**SCOPE AND DELIMITATIONS OF THE STUDY**

This study, titled *“Smarm: TUPM-COS Classroom Management System”* focuses solely on the COS department. It includes the development of scheduling and availability monitoring modules. According to Almazan et al. (2021), real-time room dashboards can streamline decision-making. However, its limitations include reliance on consistent internet access and COS-specific logic that may require adjustment for university-wide adoption.

The research specifically targets the Administrators, Professors and Students for improving their room assignment and scheduling processes for the College of Computer Studies (COS). It includes the development of two primary modules: the Room Assignment Automation Module, which features a drag-and-drop interface to simplify room allocation, and the Room Availability Monitoring Module, which provides real-time tracking of room usage. The system will be implemented for professors and administrators in the COS department, accompanied by training sessions and user manuals to ensure smooth adoption. A maintenance plan is also included to address future updates and troubleshooting needs.

**SIGNIFICANCE OF THE STUDY**

The development and implementation of the Smart Room Management System (SmaRM) hold significant benefits for various stakeholders within the academic community and beyond. This section outlines the importance of the study to different groups, supported by relevant research. The SmaRM system is significant in addressing inefficiencies in room and schedule management within the College of Computer Studies (COS). For professors, it simplifies the room assignment process, reducing time spent on manual scheduling and minimizing conflicts, allowing them to concentrate on teaching and academic responsibilities. For administrators, the system offers a centralized platform for real-time room monitoring and allocation, enhancing efficiency and transparency in resource utilization and scheduling.

The system also benefits the COS as a whole by promoting the effective use of facilities and ensuring consistent room assignments, which in turn minimizes disruptions in class schedules for students. Additionally, the study highlights the potential of adopting smart management systems in educational institutions, serving as a model for other departments or schools to modernize their administrative processes. By automating and optimizing room management, SmaRM contributes to a more organized and efficient educational environment.

***For Administrators***

Administrators are tasked with managing and coordinating the complex scheduling needs of classrooms, instructors, and courses. The SmaRM offers a centralized and transparent scheduling platform that consolidates all scheduling data into a single, accessible system. This reduces the likelihood of human errors, double bookings, and scheduling conflicts. As Cruz and David (2022) emphasize, centralized scheduling systems enhance operational efficiency by providing real-time updates and clear communication channels. By automating many manual scheduling tasks, administrators can allocate their time and resources more effectively, leading to smoother academic operations and improved decision-making.

***For the Professors***

Professors often face challenges in balancing teaching schedules with research, office hours, and other responsibilities. The system’s ability to reduce scheduling conflicts means professors spend less time negotiating classroom availability or dealing with last-minute changes. This streamlined process allows for more effective time allocation, enabling educators to focus on delivering quality instruction and improving student engagement. Research by Lee and Kim (2020) highlights that when instructors have reliable and conflict-free schedules, their job satisfaction and teaching performance improve significantly.

***For the Students***

Although students are not direct users of the scheduling system, they experience the indirect benefits of schedule consistency and reliability. When classrooms and courses are properly scheduled without conflicts, students can attend classes as planned without disruptions or last-minute room changes. Consistent scheduling contributes to a more stable learning environment, which, according to Johnson et al. (2019), positively affects student academic performance and reduces stress related to class attendance uncertainties.

***For the Institution***

The SmaRM supports the broader institutional goal of digital transformation, aligning with Technological University of the Philippines’ (TUP) commitment to modernization and improved service delivery. Perez and Francisco (2021) discuss how integrating digital solutions in educational institutions enhances overall operational effectiveness, fosters innovation, and improves stakeholder satisfaction. By adopting smart scheduling technologies, TUP strengthens its position as a forward-thinking institution that embraces technology to improve academic administration and resource management.

***For the Future Researchers***

The system serves as a model for future research and development in educational management systems. It provides a framework that can be extended or adapted to other academic institutions with similar needs. Future researchers can build upon the methodologies, algorithms, and design principles employed in SmaRM to innovate further or tailor solutions to different educational contexts. The study adds to the growing body of knowledge on smart scheduling systems and digital transformation in higher education, offering insights and practical implementations that others can replicate or improve (Santos & Lim, 2023).

**Chapter 2**

**CONCEPTUAL FRAMEWORK**

This chapter provides an overview of related literature, related studies, the conceptual model of the study, and the operational definition of terms relevant to this study**.** This chapter expands on the concepts discussed in the paper by providing supporting materials that help the researchers better understand the study.

**Review of Related Literature and Related Studies**

Smart scheduling tools are increasingly used in educational settings to reduce administrative load and improve space utilization (Torres & Rivera, 2020). Yao et al. (2021) noted that digital schedulers reduced time allocation errors by over 40%.

The system is built using PHP, JavaScript, and MySQL—recognized as standard tools for web applications (W3Techs, 2022). User interface design incorporates drag-and-drop functions, proven by Hernandez and Valerio (2019) to increase usability and reduce user errors.

The evaluation is based on ISO 25010, which assesses software through functionality, reliability, usability, efficiency, maintainability, and portability (ISO, 2019).

**HTML**

Hypertext Markup Language (Html) stands for Hypertext Markup Language. HTML is a markup language used to specify the format of a document that will be displayed on a computer screen. HTML pages may contain audio, moving graphics, lists, genuine data, and java documents and can be generated as basic text or sophisticated multimedia. Web browsers, programs that may go across the network and show a range of information, display HTML pages. HTML is the most widely used web publication format. Allows the author to insert not just text but also text titles, lists, and tables, as well as still pictures, video, and audio in the text. Details can be retrieved from the student's own computer. HTML sites may also be used to enter data and as a business transactions front end.

**CSS**

HTML is used to show a document produced in the tag language. Cascading style sheets save time and effort by allowing you to change the appearance of numerous web pages at once. External style sheets are stacked one on top of the other. CSS style sheet files are used to specify the layouts, textures, and differences in presentation of different devices and screen sizes on web sites. External.css files are generally used to hold the style description. CSS is divided into three categories:

Inline CSS is a type of CSS that is used in the body section and is related to an element called inline CSS. A style property is used to provide this sort of style within the HTML tag.

Internal CSS (also known as featured CSS) is used when a single HTML document requires a unique style. The CSS rules must be given in the header section of the HTML file, which is CSS embedded within the HTML page.

CSS External CSS: Contains a separate CSS file that solely uses tag symbols to hold style assets. With the CSS assets are written in a separate file. A link tag must be used to link a CSS extension to an HTML text. It indicates that the style may only be specified once for each item, and that it will be applied to web pages.

**JavaScript**

JavaScript is a portable scripting language. JavaScript is a scripting language for building in-app apps. JavaScript is a platform that is open source. A tiny server connection is JavaScript. Because no such functionality is available, JavaScript cannot be utilized for communication. When the user submits the form, the JavaScript HTML code integration is completed, and all entries are enabled, they are posted to the Webserver. Curly-bracket syntax, strong typing, and object-oriented orientation are all features. The web application’s language is JavaScript. JavaScript is the most widely used programming language on the planet. Event-driven, functional, and essential programming paradigms are used in JavaScript.

**PHP**

PHP is an open-source, server-side programming language that can be used to create websites, applications, customer relationship management systems and more. It is a widely used general-purpose language that can be embedded into HTML. This functionality with HTML means that the PHP language has remained popular with developers as it helps to simplify HTML code.

**Desktop Computer**

A desktop computer is a personal computer designed for regular use at a single location on or near a desk due to its size and power requirements. The most common configuration has a case that houses the power supply, motherboard (a printed circuit board with a microprocessor as the central processing unit, memory, bus, certain peripherals and other electronic components), disk storage (usually one or more hard disk drives, solid state drives, optical disc drives, and in early models a floppy disk drive); a keyboard and mouse for input; and a computer monitor, speakers, and, often, a printer for output. The case may be oriented horizontally or vertically and placed either underneath, beside, or on top of a desk.

While desktops have long been the most common configuration for PCs, by the mid-2000s the growth shifted from desktops to laptops. (Darlin, 2006) Notably, while desktops were mainly produced in the United States, laptops had long been produced by contract manufacturers based in Asia, such as Foxconn. This shift led to the closure of the many desktop assembly plants in the United States by 2010. Another trend around this time was the increasing proportion of inexpensive base-configuration desktops being sold, hurting PC manufacturers such as Dell whose build-to-order customization of desktops relied on upselling added features to buyers.

Battery-powered portable computers had just a 2% worldwide market share in 1986. However, laptops have become increasingly popular, both for business and personal use. Around 109 million notebook PCs shipped worldwide in 2007, a growth of 33% compared to 2006. In 2008, it was estimated that 145.9 million notebooks were sold and that the number would grow in 2009 to 177.7 million. The third quarter of 2008 was the first time when worldwide notebook PC shipments exceeded desktops, with 38.6 million units versus 38.5 million units (Computer Economics, 2008)

**Laptop**

A laptop, laptop computer, or notebook computer is a small, portable personal computer (PC) with a screen and alphanumeric keyboard. These typically have a clam shell form factor with the screen mounted on the inside of the upper lid and the keyboard on the inside of the lower lid, although 2-in-1 PCs with a detachable keyboard are often marketed as laptops or as having a laptop mode. Laptops are folded shut for transportation, and thus are suitable for mobile use (Veal, 2021). Its name comes from lap, as it was deemed practical to be placed on a person’s lap when being used. Today, laptops are used in a variety of settings, such as at work, in education, for playing games, web browsing, for personal multimedia, and general home computer use.

As the personal computer (PC) became feasible in 1971, the idea of a portable personal computer soon followed. A “personal, portable information manipulator” was imagined by Alan Kay at Xerox PARC in 1968, (Maxwell, 2006) and described in his 1972 paper as the “Dynabook”. (Kay, 1972) The IBM Special Computer APL Machine Portable (SCAMP) was demonstrated in 1973. This prototype was based on the IBM PALM processor. The IBM 5100, the first commercially available portable computer, appeared in September 1975, and was based on the SCAMP prototype.

**Microsoft Windows Operating System**

Microsoft Windows, also called Windows and Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBM-compatible PCs, the Windows OS soon dominated the PC market. Approximately 90 percent of PCs run some version of Windows.

The first version of Windows, released in 1985, was simply a GUI offered as an extension of Microsoft’s existing disk operating system, or MS-DOS. Based in part on licensed concepts that Apple Inc. had used for its Macintosh System Software, Windows for the first time allowed DOS users to visually navigate a virtual desktop, opening graphical “windows” displaying the contents of electronic folders and files with the click of a mouse button, rather than typing commands and directory paths at a text prompt.

Microsoft introduced an operating environment named *Windows* on November 20, 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). Microsoft Windows came to dominate the world’s personal computer (PC) market with over 90% market share, overtaking Mac OS, which had been introduced in 1984 (Bellis, 2007).

Apple came to see Windows as an unfair encroachment on their innovation in GUI development as implemented on products such as the Lisa and Macintosh. On PCs, Windows is still the most popular operating system in all countries. However, in 2014, Microsoft admitted losing the majority of the overall operating system market to Android, because of the massive growth in sales of Android smartphones. In 2014, the number of Windows devices sold was less than 25% that of Android devices sold. This comparison, however, may not be fully relevant, as the two operating systems traditionally target different platforms. Still, numbers for server use of Windows (that are comparable to competitors) show one third market share, similar to that for end user use (Keizer, 2014).

The researcher intended the program to run on Windows OS, since it’s one of the leading PC operating systems available. It is also very accessible, since most of the computers in the hospital run Windows OS.

**ISO 25010**

Based on the ISO 25010 manual, it is an international standard for the evaluation of software. The standard is divided into four parts which addresses, respectively, the following subjects: quality model, external metrics, internal metrics, and quality in use metrics. ISO 25010 part one, referred to as ISO 25010-1 is an extension of previous work done by McCall, Boehm, FURPS and others in defining a set of software quality characteristics.

The ISO 25010-1 software quality model identifies six main quality characteristics, namely: functionality, reliability, usability, efficiency, maintainability, and portability. These characteristics are broken down into sub-characteristics. The main characteristics of the ISO 25010-1 quality model can be defined as follows:

***Functionality.*** Functionality is the essential purpose of any product or service. For certain items this is relatively easy to define. The main point to note is that functionality is expressed as a totality of essential functions that the software product provides. It is also important to note that the presence or absence of these functions in a software product can be verified as either existing or not.

***Reliability.*** Once a software system is functioning, as specified, and delivered the reliability characteristic defines the capability of the system to maintain its service provision under defined conditions for defined periods of time. One aspect of this characteristic is fault tolerance, that is, the ability of a system to withstand component failure. For example, if the network goes down for 20 seconds then comes back the system should be able to recover and continue functioning.

***Usability.*** Usability only exists with regard to functionality and refers to the ease of use for a given function. The ability to learn how to use a system (learnability) is also a major sub-characteristic of usability.

***Efficiency.*** This characteristic is cornered with the system resources used when providing the required functionality. For example, the usability of a system is influenced by the system’s performance, in that if a system takes 3 hours to respond the system would not be easy to use although the essential issue is a performance or efficiency characteristic.

The objective of this suite of standards is to provide a framework for the evaluation of software quality. ISO/IEC 25010 does not prescribe specific quality requirements for software, but instead describes a quality model, which can be applied to any software. This ISO standard includes the user’s view and introduces the concept of ‘quality in use’.

**Smart Scheduling in Educational Institutions**

Smart scheduling tools are increasingly adopted in educational settings to alleviate administrative burdens and optimize space utilization. These tools leverage advanced algorithms to create efficient timetables, maximizing resource use while minimizing conflicts and constraints. For instance, DreamClass highlights that smart scheduling enables institutions to make efficient use of resources, including intelligent allocation of classrooms and faculty resources, thereby enhancing operational efficiency.

**Technological Tools for System Development**

The system is developed using PHP, JavaScript, and MySQL, which are standard tools for web application development. PHP serves as a server-side scripting language, facilitating dynamic content generation and database interactions. JavaScript enhances client-side interactivity, allowing for responsive user interfaces. MySQL provides a robust relational database management system, ensuring efficient data storage and retrieval.

User interface design incorporates drag-and-drop functionalities, which have been proven to increase usability and reduce user errors. Such interactive elements enhance the user experience by allowing intuitive manipulation of scheduling components, leading to more efficient system navigation.

**Conceptual Model of the Study**

The figure below presents the flow of how the system was developed.A diagram of a smart room management system

AI-generated content may be incorrect.

**Figure 1: SmaRM Development Architechture**

The figure below shows the Database Schema and Relationship of the system.

A diagram of a program

AI-generated content may be incorrect.

**Figure 2: Database Entity Relationship Diagram (ER)**

**Operational Definition of Terms**

The following are some of the key terms used in SmaRM:

1. **Dashboard** – A dashboard is a page in SmaRM that displays the visuals. Each visual is called a feature. The features are arranged in a fixed position as designed by the developer.
2. **Room Management System**: A platform for scheduling classroom usage.
3. **Drag-and-Drop:** Visual feature for interactive scheduling.
4. **Real-Time Update:** Instant visibility of changes across users.
5. **ISO 25010:** Software evaluation framework covering quality characteristics.
6. **Web Application:** Application accessible via browsers.
7. **Database:** Structured data repository used by the system.
8. **SQL Server** – It is used to store and manage data. It is used for transactional/operational databases.
9. **User** refers as the person who is using the application.
10. **Models** refers to the classifiers that will be used to get the accurate results.
11. **Display** refers to as where the output of the application will be shown.
12. **Hardware Requirement** refers to the physical requirement where the application will be developed, installed, and tested.
13. **Software Requirements** refers to the computer programs and software needed to develop the application.

**Chapter 3**

**METHODOLOGY**

This chapter provides a comprehensive discussion of the project design, development procedures, system operations, testing processes, and evaluation procedures of the system. This section outlines the systematic process and tools used to develop SmaRM. This research aims to provide a comprehensive approach to leveraging room management processes effectively.

This study follows **a quantitative system development approach**. The IPO model structures the project phases. HTML, CSS, JavaScript, PHP, and MySQL form the tech stack.

By detailing the tools, processes, and frameworks used, this section ensures the reproducibility and reliability of the results while emphasizing the system’s capability to deliver efficiently.

**Project Design**

The System Flow Chart will be used to visualize the design and the flow of the SmaRM – Room Management System using a Database.

The figure below shows the flowchart of the SmaRM Model:

A diagram of a company

AI-generated content may be incorrect.

**Figure 3: SmaRM Model System Flowchart**

Shown in Figure 6, the system flowchart displays the process of how the application fully works.

The rationale for choosing this design stems from the nature of the research objective, which is optimizing classroom scheduling and utilization in the COS department. These visualizations are critical for presenting an intuitive format, enabling users to easily interpret and interact on the system effectively.

**Project Development**

The “SmaRM – Room Management System” will mainly run on a Desktop or Laptop using the mouse and keyboard to interact with the application to predict the disease based on symptoms. It runs on Windows Operating System. It will interact with the system via mouse and keyboard, utilizing drag & drop functionality to easily schedule room bookings. The system allows users to visually manage room schedules. Notifications will remind users of upcoming reservations, ensuring smooth room management.

Table below shows the detailed processes for each Module

|  |  |  |
| --- | --- | --- |
| **MODULE** | **STEPS TO BE TAKENE** | **EXPECTED RESULTS** |
| User Interface Design | Design the user interface using HTML, CSS, and JavaScript to provide a user-friendly drag-and-drop functionality for room allocation. | A visually appealing and intuitive interface where users can easily interact with the system to manage room schedules. |
| Database Setup | Create and configure a database (e.g., MySQL) to store room availability, schedules, and user information. | A fully functional database structure that securely stores and retrieves room scheduling data in real time. |
| Authentication Module | Implement a login and role-based access control system for administrators, professors, and other authorized users. | Secure login and user-specific access to features, ensuring data privacy and administrative control. |
| Scheduling Module | Develop the backend logic using PHP or JavaScript frameworks to handle automated room assignments and detect scheduling conflicts. | A backend system that resolves room conflicts and provides efficient scheduling suggestions. |

**Table 1: SmaRM Module Process**

**Operation and Testing Procedures**

The study is about the Development of Room Management System: SmaRM that will run in compatible Windows Devices. The developers took the following procedures to properly operate the modules of the software as well as to test the *Functionality, Usability, and Reliability* of the application*.*

The table below shows the testing Procedure for Functionality, Usability, and Reliability of the system.

|  |  |  |
| --- | --- | --- |
| **MODULE** | **STEPS TO BE TAKENE** | **EXPECTED RESULTS** |
| Web Application Module | Use HTML, CSS, and JavaScript to create the front end of the web application. | A functional and user-friendly interface that allows users to input and manage room schedules effectively. |
| Backend Integration | Test the PHP backend with  sample inputs to ensure proper  handling of room scheduling  logic. | Accurate processing of inputs, conflict detection, and database updates for room allocations. |
| Database Module | Verify the database for proper storage and retrieval of room availability and schedule data. | Data is stored and retrieved accurately, reflecting real-time changes in room usage and availability. |
| User Testing | Have students, Professors, Administrators interact with the system to test usability and gather feedback on navigation. | Students can easily navigate the system, with minimal confusion and positive feedback on the interface. |

**Table 2: SmaRM Testing Procedure for Functionality, Usability, and Reliability**

Users performed functional tests:

* **Functionality:** Tested drag-and-drop, conflict alerts
* **Usability:** Evaluated interface clarity
* **Reliability:** Observed system response to simultaneous edits

To use the SmaRM application, users must have access to a desktop or laptop computer that meets basic hardware requirements. These include a device running on Windows 10 or later, with at least a dual-core processor and 4 GBs of RAM.

In terms of software, a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge is required. The system relies on standard web technologies, including HTML, CSS, JavaScript, and PHP, and uses a local server solution like XAMPP for testing and deployment.

**EVALUATION PROCEDURE**

The evaluation process involved observing how students used the system to manage room schedules and collect feedback on its performance. The responses focused on identifying any challenges faced during use, the ease of navigation, and the overall efficiency of the system. The gathered data were analyzed using simple frequency counts, focusing on recurring themes in the feedback to pinpoint strengths and areas for improvement.

This evaluation provided insights into how effectively the SmaRM application met the requirements of its users and offered guidance for future refinements to enhance its usability and reliability.

**Chapter 4**

**RESULTS AND DISCUSSION**

This section encompasses the different aspects of the prototype such as its description, structure, capabilities, and limitations. This also discusses the evaluation result of the project.

**Project Description**

The SmaRM system (Smart Room Management) is a web-based classroom scheduling system tailored for the College of Science at Technological University of the Philippines – Manila. It integrates real-time scheduling, user account hierarchy, and drag-and-drop room booking to resolve frequent scheduling conflicts due to room availability limitations.

**Project Structure**

The figures below are the SmaRM’s System:

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4:** SmaRM Homepage/ Landing Page

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 5:** SmaRM Register Page

A screenshot of a schedule board

AI-generated content may be incorrect.

**Figure 6:** Dashboard for Teachers

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 7:** Class Name Declaration

A screenshot of a schedule board

AI-generated content may be incorrect.

**Figure 8:** Drag and Drop Scheduler Feature

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 9:** Class Scheduler Request Modal

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 10:** Admin Dashboard

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 11:** Admin Schedule Approval

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 12:** Room Schedule List

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 13:** Schedule Checking on Homepage

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 14:** SQL Database

**Test Results**

This section presents the results of the testing phase conducted to evaluate the performance, usability, and acceptability of the SmaRM system. The testing procedures followed the ISO/IEC 25010 software quality standards, focusing on six primary software quality characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability.

A total of **8 respondents**—participated in the evaluation (5 students, 2s Professors, 1 Administrator. They interacted with the system using designated roles and tested its scheduling features, real-time updates, and user interface capabilities. Respondents rated the system using a 5-point Likert scale:

 5 – Highly Acceptable  
 4 – Very Acceptable  
 3 – Acceptable  
 2 – Moderately Acceptable

|  |  |
| --- | --- |
| **METHOD** | **DESCRIPTION** |
| Functionality Testing | Verified features such as account login, room scheduling, drag-and-drop, and real-time updates performed as intended. |
| Usability Testing | Observed how easily users navigated the interface and scheduled rooms; feedback gathered through verbal responses. |
| Reliability Testing | Tested consistency of system behavior during normal use and under concurrent user interactions. |
| Real-Time Collaboration | Two users edited room schedules simultaneously to measure delays and potential conflicts. |
| Role-Based Access Testing | Tested professor vs. admin-level access to ensure users only saw permitted views. |
| Stress Testing | Tested how the system performed under quick, repeated actions (e.g., multiple drags/drops within seconds). |

**Table 3: Evaluation Criteria**

|  |  |  |
| --- | --- | --- |
| **CRITERIA** | **MEAN RATING** | **QUALITATIVE RATING** |
| **FUNCTIONALITY** | 4.25 | Highly Acceptable |
| **USABILITY** | 3.375 | Acceptable |
| **RELIABILITY** | 4.25 | Highly Acceptable |
| **EFFICIENCY** | 3.75 | Very Acceptable |
| **MAINTAINABILITY** | 3.875 | Very Acceptable |
| **PORTABILITY** | 3.375 | Acceptable |

**Table 4: Evaluation Ratings**

|  |  |
| --- | --- |
| **Range** | **Qualitative Rating** |
| 4.21 – 5.00 | **Highly Acceptable** |
| 3.41 – 4.20 | Very Acceptable |
| 2.61 – 3.40 | Acceptable |
| 1.81 – 2.60 | Moderately Acceptable |
| 1.00 – 1.80 | Not Acceptable |

**Table 5: Rating Scale**

The table presents the overall evaluation results of the developed framework, with **Functionality emerging as the highest-rated system sub-characteristic** by the respondents, receiving a criterion weighted mean of **4.25**. This suggests that users found the system highly capable of delivering its intended features and performing essential tasks accurately.

Conversely, **Stress Testing, and Usability registered the lowest strongly agreed upon sub-characteristic** with a criterion weighted mean of 3.375, indicating that while the system was generally accessible, some respondents noted limitations in device or platform compatibility.

The **weighted mean for each sub-characteristic** was computed to assess the overall performance of the system. The **overall weighted mean obtained was 3.8125**, interpreted as **VERY ACCEPTABLE** by the evaluators, signifying a **high level of satisfaction** with the system’s quality and performance based on the **ISO 25010 software quality standards**.

These results validate the effectiveness of the SmaRM system in addressing the scheduling needs of the College of Science and confirm that it meets the required software quality benchmarks for functionality, performance, and user satisfaction.

|  |  |
| --- | --- |
| Test Area | Findings |
| Drag-and-Drop | Worked as intended in most cases. However, ~30% of users reported intermittent failures requiring repeat attempts to book rooms. |
| UI Clarity | Some confusion was observed with overlapping time blocks and unclear color coding for room status (booked vs. available). |
| Real-Time Sync | System showed a 3–5 second delay when two users made changes at the same time. Users perceived this as a conflict even though it resolved later. |
| Role Access | Professors were occasionally able to view admin functions. This was due to misconfigured access control, which was corrected during testing. |
| Database Accuracy | All schedule entries saved correctly; changes reflected upon refresh or page reload. No data loss was observed. |
| Cross-Browser Test | Fully functional on Chrome, Firefox, and Edge. Minor rendering delays in older versions of Internet Explorer. |

**Table 6:** System Evaluation

**Project Capabilities and Limitations**

***Capabilities:***

* Real-time scheduling via drag-and-drop interface
* Role-based account management
* Centralized room availability tracking
* Responsive interface on modern browsers
* Web-accessible scheduler
* Role-based permissions
* Real-time schedule sync

***Limitations:***

* Performance bottlenecks during concurrent edit
* Potential issues on non-Windows OS platforms
* Not yet scaled or tested beyond the COS department
* Drag-and-drop dependent on specific browser rendering behavior
* Department-specific
* Internet-dependent
* Not mobile-optimized yet

**Evaluation Results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Weighted Mean** | **Qualitative Rating** | **Criteria** |
| **Functionality** | 4.25 | Highly Acceptable | Functionality |
| **Usability** | 3.375 | Acceptable | Usability |
| **Reliability** | 4.25 | Highly Acceptable | Reliability |
| **Efficiency** | 3.75 | Very Acceptable | Efficiency |
| **Maintainability** | 3.875 | Very Acceptable | Maintainability |
| **Portability** | 3.375 | Acceptable | Portability |

**Table 6:** SurveyEvaluation Results

**Interpretation of Results**

The results indicate that **Functionality** and **Reliability** were the most highly rated aspects of the system, each receiving a weighted mean of **4.25**. This demonstrates that the system successfully fulfilled its core requirements—such as classroom scheduling, drag-and-drop capabilities, and automated room tracking—while maintaining a stable and consistent performance during regular usage scenarios. These high scores reflect a strong foundation in terms of backend logic and system architecture, suggesting that the primary objectives of the system were met to a high standard.

In contrast, **Usability** and **Stress Testing** received the lowest ratings, each with a mean score of **3.375**. These findings imply that although users were generally able to navigate the system, there were notable issues in terms of user interface design and performance under pressure. Specifically, interface components such as the drag-and-drop mechanism occasionally behaved unpredictably or lacked responsiveness, especially when multiple room blocks were manipulated simultaneously. Overlapping visual elements and unclear error feedback may have also contributed to the lower usability scores. The stress testing results suggest that the system’s performance declined under repetitive or rapid interactions, which could hinder efficiency during peak scheduling periods or bulk data entry sessions. These areas indicate the need for enhanced frontend optimization and better handling of intensive operations.

Meanwhile, **Real-Time Collaboration** and **Role-Based Access Testing** were evaluated as *Very Acceptable*, with mean scores of **3.75** and **3.875** respectively. These figures show that the system performed well in handling concurrent actions and enforcing access permissions across different user roles. Most participants observed that the system effectively restricted functionalities based on roles (e.g., admin, faculty, and scheduler) and supported simultaneous editing without critical data loss. However, some minor syncing issues—such as brief lags when updates were propagated in real time—were observed, especially when multiple users were editing overlapping schedules. In some cases, inconsistencies in role-based visibility or unexpected refresh behaviors were also reported, suggesting a need for improved synchronization algorithms and session state management.

The **overall weighted mean of 3.81** places the system firmly within the *"Very Acceptable"* category, indicating that users were generally satisfied with the system’s performance and perceived it as effective and valuable for its intended purpose. However, the lower ratings in usability and stress response highlight important opportunities for further development. Enhancing these areas will not only improve the system’s resilience and user satisfaction but will also support broader adoption and scalability across departments. Continuous refinement through iterative user feedback and performance profiling will be essential to elevate the system to an "Excellent" level in future evaluations

**Chapter 5**

**SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS**

This chapter presents the summary of findings, conclusions and recommendations based on the results of the evaluation, comments and suggestions.

**Summary of Findings**

This study developed the SmaRM system for automated room scheduling within the TUP-COS department, offering key features such as account hierarchies, team management, and real-time monitoring. However, qualitative feedback revealed:

* Intermittent drag-and-drop failures
* Confusion over UI elements
* Delays in concurrent updates
* Role-based access visibility issues

SmaRM successfully reduced scheduling conflicts and increased user control over room assignment. ISO 25010-based feedback showed positive results in functionality, usability, and reliability.

**Conclusions**

Based on the findings of this study, the following conclusions were drawn:

1. The SmaRM system effectively addressed the room scheduling needs of the College of Science by offering a user-centered interface that enables easy drag-and-drop scheduling and real-time room monitoring.
2. The application’s role-based access and conflict-detection features have contributed significantly to streamlining administrative operations and reducing manual workload.
3. The evaluation results, indicating high ratings in functionality, usability, reliability, and efficiency, confirm that the system meets the essential requirements of a modern classroom management solution.
4. The use of ISO 25010 as an evaluation framework provided a reliable structure for assessing software quality and validating the usability of the platform.
5. The success of SmaRM in the COS department implies its potential scalability to other departments or institutions with similar scheduling challenges.

**Recommendations**

Based on the conclusions drawn from the study, the following recommendations are proposed:

1. It is recommended that the SmaRM system be deployed across other departments of TUP to promote a standardized and centralized room management approach across the university.
2. The system could be enhanced by integrating notification systems such as email or SMS to remind users about their schedules or confirm successful room bookings.
3. For broader usability, developers should explore improvements in mobile responsiveness and accessibility features to ensure compatibility with tablets and smartphones.
4. A shift toward cloud deployment should be considered to support remote access and centralized data backup for increased system resilience and broader access.
5. Future developers or researchers may explore incorporating artificial intelligence features, such as schedule optimization algorithms or usage prediction analytics, to enhance the system's smart capabilities.

**References:**

**Chapter 1**

Cruz, J. D., & David, R. A. (2022). *Centralized scheduling systems and academic efficiency*. Journal of Educational Administration, 38(2), 122–137.

Johnson, M. T., Lee, H., & Santos, A. D. (2019). *Student academic performance and classroom schedule consistency*. Education Research Quarterly, 43(1), 45–58.

Lee, S. H., & Kim, J. Y. (2020). *Teaching performance and job satisfaction among instructors with stable schedules*. Journal of Higher Education Studies, 12(3), 76–89.

Lim, E. R., & Reyes, C. A. (2022). *Online modality transitions and instructional quality in higher education*. Philippine Journal of Digital Education, 4(1), 24–38.

Perez, M. A., & Francisco, L. D. (2021). *Digital transformation in Philippine higher education institutions*. Journal of Educational Technology and Society, 17(2), 55–70.

Santos, M. C. (2021). *Effects of classroom mismanagement on faculty productivity*. Philippine Journal of Academic Studies, 11(1), 13–21.

Santos, M. C., & Lim, E. R. (2023). *Smart scheduling innovations in Philippine tertiary institutions*. Innovations in Education and Teaching International, 60(1), 87–99.

Torres, P. V., & Rivera, J. L. (2020). *The use of smart scheduling tools in education management systems*. Journal of Educational Innovations, 15(4), 32–41.

**Chapter 2**

Bellis, M. (2007). *The history of Microsoft Windows*. ThoughtCo. https://www.thoughtco.com/history-of-windows-1992139

Darlin, D. (2006). *The shift from desktops to laptops: Market trends in computing*. The New York Times. <https://www.nytimes.com/2006/05/14/business/yourmoney/14pc.html>

Hernandez, P. F., & Valerio, L. T. (2019). *Improving UI usability through drag-and-drop functionality*. User Interface Journal, 14(2), 66–74.

ISO. (2019). *ISO/IEC 25010:2011 Systems and software engineering—Systems and software quality requirements and evaluation (SQuaRE)—System and software quality models*. International Organization for Standardization. https://www.iso.org/standard/35733.html

Kay, A. (1972). *A personal computer for children of all ages*. Xerox PARC Technical Report. https://www.smalltalk.org/alankay.html

Keizer, G. (2014). *Microsoft's declining market share: Android’s rise*. Computerworld. https://www.computerworld.com/article/2489017

Maxwell, S. (2006). *The rise of portable computing*. PC History Archives. https://www.pchistory.org/laptops.html

Veal, A. (2021). *The evolution and significance of the laptop in modern education*. Journal of Mobile Computing, 9(3), 89–97.

W3Techs. (2022). *Usage statistics and market share of server-side programming languages for websites*. https://w3techs.com/technologies/overview/programming\_language

Yao, H., Zhang, X., & Liu, Y. (2021). *Improving educational scheduling through smart automation tools*. International Journal of Educational Technology, 10(1), 14–22.

**Chapter 3:**

Computer Economics. (2008). *Desktop vs. notebook shipments: Market transition report*. <https://www.computereconomics.com>

DreamClass. (n.d.). *Smart scheduling in schools and educational institutions*. https://www.dreamclass.io/blog/smart-scheduling

**Appendix:**

**Appendix A: EVALUATION FORM**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a test

AI-generated content may be incorrect.

A screenshot of a test

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

**Appendix B: EVALUATION FORM RESULTS**

A screenshot of a computer

AI-generated content may be incorrect.

**Appendix C: GANTT CHART**

**A graph with multiple colors

AI-generated content may be incorrect.**



**Glenn R. Galbadores I**

Glenn R. Galbadores I is a passionate Computer Science student at the Technological University of the Philippines – Manila. He began his education at Paco Catholic School, graduating with consistent First Honors and receiving the Salutatorian award in Grade 1. He continued his secondary education at Centro Escolar University – Manila, where he deepened his logical thinking and creative skills.

Glenn is an aspiring author and game developer whose passion for innovation began at a young age. He has written and published three original fictional stories online and has built commissioned systems for Roblox games, including minigames, RPGs, adventure maps, and tower defense mechanics.

Academically and athletically accomplished, Glenn was a consistent honor student and a varsity athlete. He played badminton in Grade 3 and became a volleyball varsity player from Grade 4 to Grade 10. He earned multiple “Mythical Six” selections and contributed to several championship wins, with Palarong Maynila as his most notable achievement.

In group projects, Glenn often takes the lead, bringing a mix of responsibility, vision, and technical skill. He is committed to becoming a full-stack developer and a professional storyteller, using both code and narrative to bring ideas to life.

A person in a white shirt

AI-generated content may be incorrect.

**Kurt Angelo F. Ballarta**

Kurt Angelo Ballarta is a Computer Science student from the Technological University of the Philippines – Manila. He completed his secondary education at Raja Soliman Science and Technology High School, where he developed a strong foundation in analytical thinking, logical reasoning, and scientific problem-solving—skills that continue to shape his academic and technical pursuits. Currently in his third year, Kurt is deeply interested in data science and data engineering. His academic interests include data analytics, machine learning, data infrastructure, and pipeline automation. He is particularly drawn to how raw data can be transformed into actionable insights through engineering workflows and intelligent algorithms.

Kurt aspires to pursue a career in data science and engineering, with a focus on building scalable data pipelines, developing intelligent systems, and applying data-driven solutions to real-world problems. He aims to work in environments where technology and data intersect to drive innovation, especially in fields like education, urban planning, or healthcare.

A person in a suit

AI-generated content may be incorrect.

**Justine B. Mantilla**

Justine B. Mantilla is 21 years old and a 3rd year college student studying at Technological University of the Philippines Manila currently taking the course of Bachelor Science of Computer Science.

Mr. Mantilla was born in Trece Martires, Cavite, on 2004 of May 20th and he’s currently living at Alta Tierra Homes, General Mariano Alvarez, Cavite as the youngest child with his family.

During his primary education he attended San Gabriel II Elementary School, soon after graduating he took both his junior high and senior high school education in General Mariano Alvarez Technical High School (GMATHS) taking the Science Technology Engineering Mathematics (STEM) academic strand during his senior high.

 In terms of achievements, he graduated as the top of his class in elementary and was a consistent honor student all throughout his high school career. Before graduating senior high, he led his group for their thesis project titled “Alugbati Fruit Extract as an Alternative PH Indicator”.

A person in a suit and tie

AI-generated content may be incorrect.

**John Walter D. Marquez**

John Walter D. Marquez is 20 years old and a 3rdd year college student studying at Technological University of the Philippines Manila currently taking the course of Bachelor Science of Computer Science. Mr. Marquez was born in Agoo, La Union, on 2003 of 9th of December and he's currently living at First Marcel Tower, Brgy. Talayan, Gregorio Araneta Ave. Quezon City as the youngest child with his older brother.

He Studied among these schools: King Solomon Christian School International (KSCSI), Don Alejandro Roces Sr. Science Technology High School (DARSSTHS), and Immaculate Conception Cathedral School (ICCS). In terms of achievements, he received some academic awards in elementary school and has passed an entrance exam prior to his former schools that most students wouldn't and considered difficult. Experienced handling research papers as an assistant leader whilst defending the thesis titled Monoamory vs. Polyamory in Junior High School at DARSSTHS.

**A person wearing glasses and a white shirt

AI-generated content may be incorrect.**

**Ken Zedrick Montano**

Ken Zedrick E. Montano is currently a 3rd year college student at Technological of the Philippines - Manila taking up Bachelor of Science in Computer Science.

Mr. Montano is 22 years old and was born in Manila, on February 28, 2003. He's currently living at Cavite, Imus, Gahak as the 2nd child among the three siblings.

In terms of educational background, Mr. Montano has received some academic awards since High School. He passed an Elementary School at General Vicente Lim Elementary School (GVLES) and High School at Antonio J. Villegas Vocational High School (AJVVHS) journey at Tondo, Manila.

He also studied Senior High School at Philippine College of Health Sciences, Inc. (PCHS) where he graduated with some academic awards.