CENTRAL MINDANAO UNIVERSITY STUDENT ORGANIZATION EVENT MONITORING AND ACCOUNTABILITIES MANAGEMENT SYSTEM

QUEEN R. BAHAY

JERRICHO B. PAMISA

ALPHALYN P. QUINTERO

UNDERGRADUATE CAPSTONE PROJECT SUBMITTED TO THE FACULTY OF INFORMATION TECHNOLOGY, COLLEGE OF INFORMATION SCIENCES

AND COMPUTING, CENTRAL MINDANAO UNIVERSITY, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

MAY 2023

TABLE OF CONTENTs

Table of Contents

[CHAPTER I 4](#_Toc151665300)

[1.1 Project Context 4](#_Toc151665301)

[1.2 Statement of the Problem 6](#_Toc151665302)

[1.3 Objectives of the Study 6](#_Toc151665303)

[1.4 Scope and Limitations 6](#_Toc151665304)

[1.5 Significance of the Study 7](#_Toc151665305)

[CHAPTER II 8](#_Toc151665306)

[2.1 Review of Related Concept 8](#_Toc151665309)

[1.1. Student Organizations 8](#_Toc151665310)

[1.2. QR-Code 8](#_Toc151665311)

[1.3. Accountabilities 9](#_Toc151665312)

[2.2. Review of Related System 10](#_Toc151665315)

[i. QR-Code Based Attendance System 10](#_Toc151665316)

[ii. Scanning and Verification System 12](#_Toc151665317)

[iii. Event Management System 13](#_Toc151665318)

[iv. Membership Management System 13](#_Toc151665319)

[CHAPTER III 15](#_Toc151665320)

[FRONT-END FRAMEWORK 15](#_Toc151665321)

[BACK-END FRAMEWORK 15](#_Toc151665322)

[CHAPTER IV 16](#_Toc151665323)

[4.1 Conceptual Framework 16](#_Toc151665324)

[4.2 Systems Development Methodology 17](#_Toc151665325)

[a. Requirement Analysis 17](#_Toc151665326)

[b. Design 18](#_Toc151665327)

[c. Develop/Implementation 19](#_Toc151665328)

[d. Testing/Verification 19](#_Toc151665329)

[i. Beta Testing 19](#_Toc151665330)

[e. Review 20](#_Toc151665331)

[ii. System Usability Scale (SUS) Survey 20](#_Toc151665332)

[f. Deployment and Maintenance 21](#_Toc151665333)

[4. 3 System Analysis 23](#_Toc151665334)

[4.3.1. Flowcharts 23](#_Toc151665345)

[4.3.2. Entity Relationship Diagram 24](#_Toc151665346)

[4.3.3. Use Case Diagram 25](#_Toc151665347)

[4.3.4. Database Schema 25](#_Toc151665348)

[4.3.5. Data Dictionary 26](#_Toc151665349)

[4.4 System Requirements Specification 27](#_Toc151665350)

[I. Functional Requirements 28](#_Toc151665351)

[II. Non-Functional Requirements 29](#_Toc151665352)

[4.5 Gantt Chart 31](#_Toc151665353)

[CHAPTER V 32](#_Toc151665354)

[5.1. Results 32](#_Toc151665355)

[5.2 System Process 32](#_Toc151665356)

[5.3. Student Organization Module 34](#_Toc151665357)

[5.3.1. Login Page 34](#_Toc151665358)

[5.3.2. Student Organization’s Roles 34](#_Toc151665359)

[5.3.3. Student Organization Dashboard 35](#_Toc151665360)

[5.3.4. Attendance Schedule 36](#_Toc151665361)

[5.3.5. Record Attendance 37](#_Toc151665362)

[5.4. Student Member Module 38](#_Toc151665363)

[5.5. Repository Data 39](#_Toc151665364)

[5.6. Beta Testing 40](#_Toc151665365)

[5.7. System Usability Scale Survey 41](#_Toc151665366)

[5.8. SUS Survey Score 42](#_Toc151665367)

# CHAPTER I

**INTRODUCTION**

## Project Context

On almost every college campus, there are student groups that represent a wide range of student interests, including community service, political engagement, Greek life, intercultural projects, the fine arts, and many more. These events honor the diversity, abilities, and interests of the students that attend each university. These groups also provide a "secret curriculum" that helps students excel in organizational settings while simultaneously benefiting their communities, (Honors et al., 2011).

The Central Mindanao University has a wide variety of student organizations which can help colleges to be easily identified. In the school year 2022-2023, Central Mindanao University has 105 registered student organizations and 10 student councils that are affiliated with the 10 colleges. These student organizations play a vital role in organizing and coordinating activities for their respective members. However, this has also brought about a challenge in maintaining an organized attendance record and has resulted in a time-consuming process of checking accountabilities.

The increasing number of student organizations and their activities has made tracking attendance and checking accountabilities complex and time-consuming. Central Mindanao University should prioritize addressing these challenges by implementing digital solutions or streamlined systems to enhance efficiency, accountability, and organization within the student organizations. This will create a more productive environment for the members.

Another problem here is that addressing the queries of a hundred students is hard. Showing to the students that they have their accountabilities is difficult when you don’t have the accurate documents. According to (Usman & Dangara, 2016), among the problem areas that highlight the sector's poor performance are the inadequate school product quality, weak administrative procedures, and a lack of accountability in the educational system, political hiring and appointment of school leaders, ineffective oversight, and a faulty quality assurance and control method.

A QR code is a 2-dimensional matrix type barcode that can store data. It is a popular method for information dissemination that can be decoded using smartphones, digital tablets, and other electronic devices. QR codes have a high storage capacity and fast readability, which makes them useful for a variety of applications such as attendance tracking, product labeling, and marketing campaigns (Muhammad et al., 2020). Using a QR code-based attendance system can save lecturing time and enhance the educational process. The system is based on a QR code that is displayed for students during or at the beginning of each lecture. Students can scan the QR code using their smartphones, which captures their facial image and communicates the information collected to confirm attendance. The whole process takes less than a minute for any student as well as for the whole class to complete their attendance confirmation. This system can help teachers to take attendance quickly and efficiently, allowing them to focus more on teaching and less on administrative tasks (Masalha et al., 2014).

The proposed project aimed to create a web-based system for managing events and accountabilities, providing features such as event management, attendance tracking, resource management, communication tools, task tracking, and reporting capabilities. This would have enabled student organizations to streamline operations, improve communication and collaboration, promote accountability, and gain valuable insights.

## Statement of the Problem

This proposed capstone project intended to simplify how student organizations established their events and monitored the attendance of their respective members, and the processing of their accounts. The project also addressed the following problems:

1. Lack of transparency on accountabilities.
2. Time consuming monitoring and checking of attendance.
3. Lack of centralized event monitoring.
4. Struggle of receiving evaluation and feedback on events.

## Objectives of the Study

The main objective of this proposed project was to create, develop, and deploy the Central Mindanao University Student Organization Event Monitoring and Accountabilities Management System to make the new system of the CMU student organization convenient. Specifically, it aimed to:

1. Develop a system that ensured transparency in managing and tracking financial accountabilities of student organization events.
2. Create a system for recording attendance at student group events that made use of QR code technology to simplify and automate the attendance monitoring process.
3. Create a system for comprehensive event monitoring of every student organization.
4. Develop an integrated evaluation and feedback mechanism within the system to facilitate the collection and analysis of feedback from event attendees and participants.
   1. Scope and Limitations

The proposed project focused on developing a student organization events and accountability management system specifically for Central Mindanao University. It aimed to address the challenges faced by student groups by providing a solution for managing events, tracking attendance, ensuring financial accountability, organizing activities, and collecting evaluation and feedback from their members on events. The project would have involved the creation of a web application as the interface for the system, integration with university departments, usability assessment, and user training. However, the implementation of network devices was not within the scope of this project.

The overall objective of the study was to enhance event management, accountability, and efficiency within the student groups of Central Mindanao University. By implementing this system, the project sought to streamline processes, improve organization and communication, and promote responsible handling of resources. The project's scope was limited to the university's student organizations, and the intended outcomes aimed to benefit both the groups and the university community as a whole.

* 1. Significance of the Study

The proposed project aimed to implement a student organization events and accountability management system at Central Mindanao University. This system would have assisted student organizations in managing their daily transactions and improving efficiency. It promoted the use of technology and provided a streamlined platform for activity management, access to information, compliance monitoring, and feedback collection. By implementing this system, student organizations would have had a centralized platform to manage events, track attendance, maintain financial accountability, and organize activities. This would have enhanced overall administration, accountability, and effectiveness among student groups at the university.

By incorporating security features into the system, student organizations, university administrators, and event attendees could have had confidence in the confidentiality and reliability of their data. This would have helped mitigate the risks associated with data breaches, identity theft, and fraudulent practices.

This proactive approach to fraud prevention would have enhanced the overall integrity of the system and contributed to maintaining a trustworthy environment for all stakeholders involved. In summary, the proposed project would have prioritized security measures to protect against potential risks such as data breaches and fraud. By implementing robust security protocols and incorporating fraud prevention mechanisms, the system would have provided a secure and reliable platform for student organizations, university administrators, and event attendees to conduct their transactions and activities.

# CHAPTER II

**REVIEW OF RELATED LITERATURE**



## Review of Related Concept

### Student Organizations

Student organizations continue to play an important role in the lives of many students today. Student organizations are groups of students who meet to pursue a common interest or goal. They are frequently affiliated with schools or universities and can focus on a variety of activities such as academic subjects, cultural activities, sports, community service, and so on. However, the nature of student organizations and the types of organizations that exist have evolved to reflect students' changing needs and interests.

According to Ebede (2015), it has been demonstrated that participation in clubs and organizations correlates positively with several areas of psychosocial development. Specifically, College juniors who are members of student organizations outperform nonmembers on factors such as educational involvement.

Several studies on extracurricular participation in college student organizations concluded that, on average, students who participated in these organizations were more likely to enjoy their college experience than those who did not. Furthermore, these student organizations appeared to offer a wide range of opportunities to their members (Ivey, 2016).

Overall, student organizations continue to play an important role in the lives of many students today, providing opportunities for personal and professional development, community engagement, and other activities. They have, however, evolved to reflect the changing needs and interests of students in the twenty-first century.

### QR-Code

A QR code is a 2-dimensional matrix type barcode that can store data. It is a popular method for information dissemination that can be decoded using smartphones, digital tablets, and other electronic devices. QR codes have a high storage capacity and fast readability, which makes them useful for a variety of applications such as attendance tracking, product labeling, and marketing campaigns (Muhammad et al., 2020). It is an optical machine-readable label that can be attached.

to items and record information related to the item. The QR code system has become popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes (Masalha et al., 2014).

QR codes can be used in a variety of security-related contexts, such as protecting the integrity of online examinations. It is possible that QR codes could also be used in authentication processes to provide an additional layer of security, but further research would be needed to determine the specifics of how this could be implemented (Petrova et al., 2016).

### Accountabilities

Student organizations manage their own funds and are given the autonomy and responsibility to spend these funds as the organization sees fit within the confines of university policies. A healthy financial structure is an important factor in student organization success and sustainability. Solid fiscal management creates a framework that allows organizations to work proactively to accomplish their goals and serve their communities with financially sustainable events and programs (Tan, J., & Pinca- Legaspi, K. 2016).

Since the 1970s, perceptions and actions related to responsibility in education have changed. We look at the evolution of accountability from earlier forms that prevailed in the post-World War II bureaucratic welfare state to more contemporary forms that appeared with the advent of the "audit society" and the post-bureaucratic state. primarily on accountability procedures that employ performance information obtained from standardized tests. Performative accountability in education, as several researchers have demonstrated, may have a variety of adverse and perverse repercussions on teachers' work and students' learning. With the growth of educational data infrastructures, this kind of accountability has been incorporated more deeply into school systems (Lingard et al., 2017).

Accountability in the context of the education system is often related with the responsible administration of limited educational resources. The goal is to guarantee that available resources are used wisely to achieve the desired educational objectives. Incorporating accountability into education management procedures can result in enhanced service delivery and organizational discipline, leading in higher system efficiency. Administrators must recognize that responsibility is a critical feature that is

associated with subordinates inside the organizational framework (Usman & Dangara, 2016).

Effective financial management is critical to the school's overall success. Accounting for funds is one of the most important aspects of managing finances in any firm. As technology has advanced in recent years, we have entered a new era of Web- based solutions. These systems, also known as inquiry systems, provide the capacity to submit data with a guaranteed response time and the certainty that the information is correct and timely (Nyondo & Lameck, n.d.).



## Review of Related System

### QR-Code Based Attendance System

The Teacher Attendance Monitoring System uses QR codes to monitor teacher attendance. Teachers are given a unique QR code that they scan when they arrive at school and again when they enter the classroom. The system then records this information in real-time, allowing principals and attendance officers to monitor teacher attendance remotely (Amirulloh et a.,l 2020). A Students Attendance System Using QR Code, the system requires a simple login process by the class instructor through its Server Module to generate an encrypted QR code with specific information.

During the class, or at its beginning, the instructor displays an encrypted QR code to the students. The students can then scan the displayed QR code using the system Mobile Module, provided to them through the smartphone market by the university. Along with the student’s facial image captured by the mobile application at the time of the scan, the Mobile Module will then communicate the information collected to the Server Module to confirm attendance. The whole process should take less than a minute for any student as well as for the whole class to complete their attendance confirmation (Masalha et al., 2014).

A related system like the High Cliff City Diskominfo has developed a Meeting Attendance System Design using a web-based QR Code with PHP programming language and MySQL database. The purpose of this system is to simplify the attendance process for agency employees and enable High Cliff Diskominfo to monitor absenteeism effectively. The system incorporates a QR Code scanning feature that can be utilized through a PC's webcam. By implementing this QR Code system, the agency anticipates an improved attendance system that streamlines meeting absences for its employees (Nasution, M., & Hanum, L. 2022). Students' attendance taking and tracking are important for monitoring their performance in class. However, tracking them manually can be tedious and time consuming. To address this issue, a unique and secure attendance tracking system is proposed that automates most of the steps involved in tracking students' attendance.

To address the issue of signature forgery, a secret code using MD5 hashing algorithm is implemented to prevent signature forgery amongst students. Implementation of the system shows that the time taken to track students' attendance using this system can be significantly reduced and the secret code is able to prevent signature forgery amongst students (Zhi et al., 2014).

According to Casunuran et al., (2020), a QR Code Attendance System with SMS Location Tracker to improve manual/traditional attendance and provide a Global Positioning System (GPS) that can track the location of students. The main purpose of the study was to design a QR Code Attendance System to improve manual/traditional attendance and provide a Global Positioning System (GPS) that can track the location of the students. The design project was tested and evaluated by 50 users and 10 experts, and based on the series of testing the system can provide an easier and more convenient recording and checking of attendance using the QR Code Scanner, as well as providing information about attendance by sending a text message and providing location by requesting on the Android Application.

The integration of Quick Response (QR) codes in classrooms has been identified as an important tool in promoting active and distributed learning, especially in higher education. A questionnaire was used to collect data from 200 students, based on a purposive sampling technique with Partial Least Squares Structural Equation Modeling (PLS-SEM).

Initial results suggested that students had very positive attitudes towards QR code utilization for course related activities, but their positive attitude was anchored on their perceived usefulness and easiness towards QR code. The social influence variable was not a predictive factor of students’ perception towards the usefulness of QR codes. The study provided practical examples of how QR codes can be integrated in teaching and recommended future experimental research into QR code effectiveness as well as instructor acceptance if the technology is to be integrated on a wider national scale (Rabu, S. et al., 2018). The benefits of using QR codes for student attendance include improved processing time and accuracy. The QR code contains time information and is generated and displayed at the lecturer’s presentation. Students can scan the code using their smartphones, which sends the information to the server for attendance processing. This eliminates the need for manual attendance taking, which can be time-consuming and prone to errors.

### Scanning and Verification System

QR codes have been used to reduce the spread of COVID-19. This article shares an innovation using QR codes to replace paper information leaflets, which is contactless and preferred to reduce viral transmission. The findings demonstrate that QR codes are a familiar, easy-to-use system and a preferred tool for delivering patient information over paper leaflets (Sharara, S., & Radia, S. 2021).

The two-dimensional quick response (QR) codes can be misleading due to their difficulty in distinguishing a genuine one from a malicious one. To evaluate the vulnerabilities and propose countermeasures, a simulated experiment was conducted where a malicious QR code directed a user to a phishing site. The results showed that hackers could easily leverage QR codes into phishing attack vectors targeting smartphone users, even bypassing web browsers’ safe browsing feature. QRCS (Quick Response Code Secure) is a universal efficient and effective solution focusing exclusively on the authenticity of the originator and the integrity of QR code by using digital signatures (Mavroeidis, V., & Nicho, M. (2017).

According to Schultz, M. (2013), This case study explored the perceptions of patrons of QR codes at Ryerson University Library and the Museum of Inuit Art. Observations and 56 patron and staff interviews were conducted to obtain data on usage, knowledge, reactions and expectations regarding QR codes in these institutions. It was found that QR code usage was low, but there was potential for use, with patrons' reactions being generally positive. Three themes were identified from an analysis of the results: an assumption that young people and smartphone owners use QR codes, that QR codes are only used for one-way provision of information, not to initiate a conversation, and that QR codes can be used to personalize a visit to an institution. Libraries and museums are advised to tailor these initiatives to their wants and needs.

According to Casunuran et al., (2020), based on the series of testing the system can provide an easier and more convenient recording and checking of attendance using the QR Code Scanner, it is also capable of providing the information about attendance by sending a text message and can provide location by requesting on the Android Application.

### Event Management System

According to Mishra, V., et al. (2016), that their online event management system software project serves the functionality of an event manager. It allows registered users to manage various types of events and provides most of the basic functionality required for an event, including selecting from a list of event types. However, it does not provide a detailed evaluation of the effectiveness of such systems in streamlining event planning and execution processes.

As stated by Jeong et al. (2016), the Event Management System provides basic functionality for managing events, such as allowing users to select from a list of event types, setting up a profile with details such as location and email, and modifying or changing the profile at any stage. It also allows administrators to interact with users as per their requirements. This proposed web application is designed to assist in strategic planning and ensure organizations are equipped with the right level of information and details of future goals.

### Membership Management System

A membership management system is a software solution that helps organizations manage their membership data, automate membership-related tasks, and streamline communication with members. It typically includes features such as a member database, event registration, payment processing, email marketing, and reporting (MemberClicks, n.d). In recent years, as the country has paid more and more attention to education, informatization of student management has become more and more important.

In student management, the simplification of information can effectively improve the efficiency of student management, so the reconstruction of student management information based on association rule mining has become very important (Xiang, Y., et al. 2022).

As explained by Hai-li, W. (2011), having a system will greatly improve the efficiency of student management, saves a lot of human and material resources, realizes the networking of college student management and evaluation, and manages and calculates students' information data more conveniently. From the point of view of Liu, Z., et al (2010), a fully functional, flexible and convenient application and friendly interface provide a good guarantee for student information management.

# CHAPTER III

**TECHNICAL BACKGROUND**

The proposed student organization events and accountabilities management system will use various web development programming languages and scripts, libraries, development tools and an external device will be used to meet the project's objectives as well.

## FRONT-END FRAMEWORK

In this proposed project, HTML, CSS, Bootstrap, Sass, and Vue.js are all important tools for creating a functional, responsive, and visually appealing Student Organization Event Monitoring and Accountabilities Management System. Each tool serves a specific purpose in the web development process and can help to streamline the development process and improve the user experience. HTML will be used to create the overall layout of the web pages, such as the navigation, header, and footer, as well as the content areas for displaying information about events and attendance.CSS is used to add style and formatting to web pages, such as font types, colors, and layout. Bootstrap can be used to create a responsive design that adjusts to different screen sizes, such as desktop, tablet, and mobile. Sass can be used to make the CSS code more organized and easier to maintain, which can help to streamline the development process.Vue.js can be used to create dynamic and interactive elements on the web pages, such as pop-ups, modals, and forms, without having to reload the page. This can help to improve the user experience and make it easier for users to navigate and interact with the system.

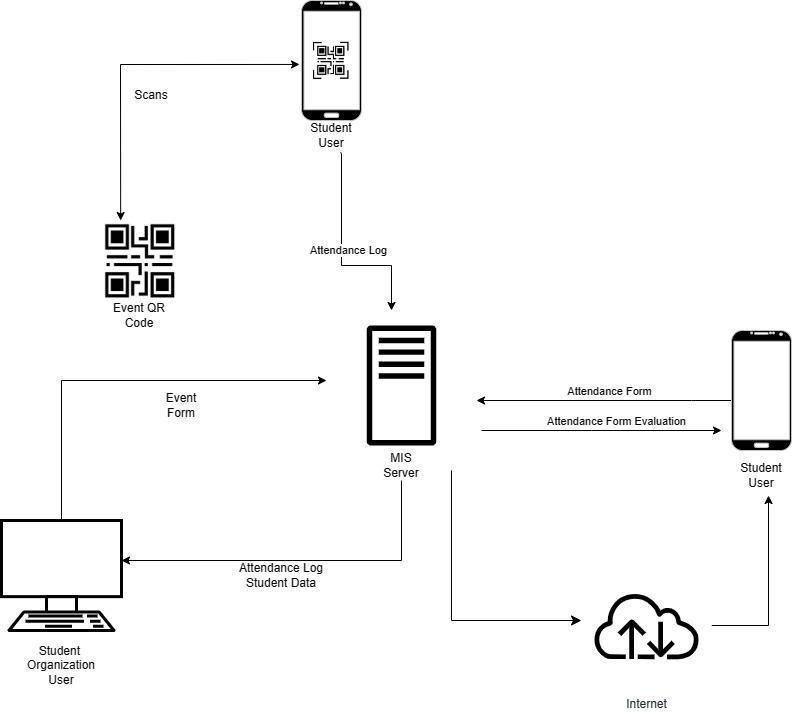
## BACK-END FRAMEWORK

By combining Laravel, JavaScript, and PHP, the proponents of this proposed project develop a powerful and effective Student Organization Event Monitoring and Accountabilities Management System. JavaScript and PHP are necessary for building dynamic, interactive web pages and managing server-side processing, while Laravel offers a robust framework for controlling the system's back end. This method satisfies the requirements of both student organizations and the school administration while being simple to use and effective.

# CHAPTER IV

**METHODOLOGY**

## 4.1 Conceptual Framework



*Figure 1. Conceptual Framework of the Student Organization Events and Accountabilities Management System*

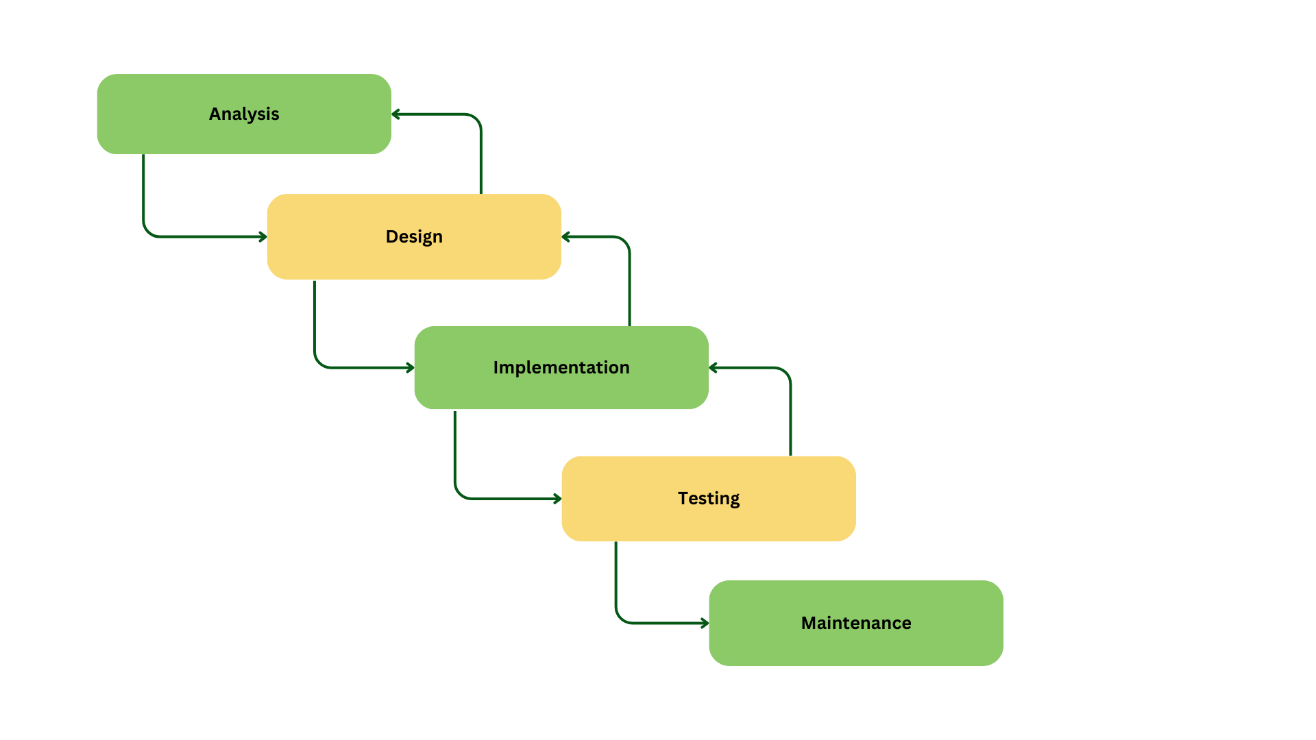
The concept of the Student Organization Events and Accountabilities Management System was to improve the monitoring of attendance as well as checking of accountabilities of each student organization and allow students to view the status of their accountabilities, promoting an efficient and paperless process.

With the use of desktops, laptops, and mobile devices with installed web browsers connected through the local network of Central Mindanao University, student organization officers were able to post events, manage event rules such as attendance checking, announcement of events, and validation of attendance through online evaluation forms.

The students could view their status on their accounts and see approaching events by visiting the system using their devices. All data generated by the system was stored in a local server in CMU, preferably together with the MIS server.

## 4.2 Systems Development Methodology

The student organization event and accountabilities management system will be developed using the Waterfall methodology.



*Figure 2. Waterfall Methodology*

The Waterfall method was suitable for developing the Student Organization Event Monitoring and Accountabilities Management System because the system had well-defined requirements, scope, and clients. The system development methodology comprised 5 different stages.

### Requirement Analysis

The initial phase of the system development methodology focused on gathering information regarding the functioning of the current attendance tracking system and obtaining details about the responsibilities related to student organization events. To achieve this, in Table 1, survey questionnaires were distributed to students through Google Forms to gain insights into how they processed their attendance and were monitored by their respective officers, how event information was shared across platforms, and how their organization-related accountabilities were established.

Additionally, any available high-ranking member of a student organization, preferably the chairperson, was interviewed based on the questions at Table 2, to gain insights of their organizational management. To determine the requirements of the system, the researchers also gathered data of what devices the users, student organizations and students used, through the questionnaire. After this phase of gathering data, the proponents then proceeded to the next phase of the system design.

### Design

During the design phase, the system's design was created, keeping in mind the identified requirements from the previous phase. The design aimed to be responsive, meaning it would adapt and work effectively on various devices such as mobile devices and web browsers. This included implementing user interfaces specifically tailored for these devices. Database designs created the structure and organization of a system's database, while web frames represented the visual representation of user interface elements, defining the overall look and feel of the system's user interfaces.

The designed system flowcharts introduced changes in the system process. This was illustrated by presenting the old process(es) first, followed by the new system flowchart(s) for each relevant old process. Following the flowcharts were the list of functional requirements, detailing all functions and features that the system would provide. Furthermore, these functional requirements were presented in the use case diagrams for consistency. The use case diagrams presented the possible scenarios of how the users would interact with the system case by case. The use case diagram also presented the boundaries of every user, and which features and/or functions were available to certain users.

The entity-relationship diagram (ERD) was used to illustrate the different entities and their respective attributes to be stored in the system’s database. This included the type of data that would be stored, and to which entity shall that particular data belong, and whether entities would share data with one another through a unique key. This diagram was then used as a reference for the construction of the database design.

Including the entities and their attributes in the database design, the proponents also set up the database users, their credentials, and their respective roles in order to limit the scope of their activities when using the database. For a more comprehensive approach in the description of the database design, the proponents created a data dictionary based on the database design. Aside from the software requirements detailed out in the previous paragraphs above, the proponents also expounded the hardware requirements for the system, including but not limited to: the hardware and system specifications of the client’s computer in order to optimally use the system, and the RFID scanner that must be used compatible to the system.

### Develop/Implementation

During this phase, the designs were translated into code. The proponents utilized the technologies introduced in Chapter II in order to accomplish the objectives of the project. In addition, the proponents simultaneously conducted alpha testing sessions in order to verify the accomplishment of functional requirements or use cases that were designed in the previous stage, and whether the step-by-step processes and/or logic were followed in the system flowcharts.

The database used in the system was set up according to the database design from the previous stage. This included the establishment of database users and roles to restrict the activities of users and the scope of their activities when using the database.

### Testing/Verification

In the testing phase, several testing methods were utilized in order to measure the quality of the website. Beta testing was the only testing that was conducted in this phase.

### Beta Testing

The system had to be fully developed before conducting beta testing, specifically closed beta testing, which involved participants chosen by the proponents and stakeholders. The participants, who served as beta testers, were selected from 50% of the currently registered student organizations, including all members of the selected organizations. These participants represented 10% of the marginal population of the student organizations.

To determine the number of participants for their proposed project, the proponents utilized a sample size calculator from SurveyMonkey by Momentive. They used this calculator to estimate the minimum recommended sample size needed to achieve the desired level of precision or confidence for the project. The proponents likely entered parameters like population size, margin of error, confidence level, and response distribution into the calculator.

### Review

After concluding the beta testing phase, the proponents requested that participants provide feedback on their user experience with the student organization event and accountability management system. To do this, participants were asked to complete the SUS survey, which was described below.

The System Usability Scale (SUS) survey consists of 10 statements that participants rated on a scale of 1 (strongly disagree) to 5 (strongly agree). Questions alternated between positive and negative statements about the system's usability, such as "I thought the system was easy to use" or "I found the system unnecessarily complex."

By administering the SUS survey, the proponents aimed to quantify the system's overall usability and collect qualitative feedback. The results provided insights into areas for improvement, as well as what aspects worked well. This user feedback was valuable for optimizing the system prior to full implementation. Completing the SUS provided participants a chance to share their experience and have a voice in the system's ongoing development.

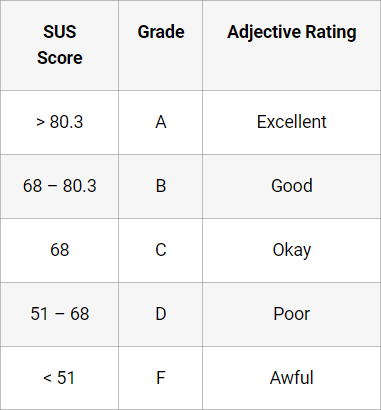
### System Usability Scale (SUS) Survey

After the beta testing, the beta testers were encouraged to answer a form that was seen in Appendix D, Tables 3 and 4, which asked for their experience when using the web application. They were also welcome to provide additional short comments regarding the web application.

This part of the beta test was called the SUS Survey. It was a questionnaire used by the proponents to measure the usability perception of the users after they used the web application. The questionnaire contained 10 questions where participants were given a scale of 1-5 to determine their level of agreement with the statements for each number. 1 meant strongly disagree while 5 meant strongly agree.

For odd-numbered questions with a positive tone, assigning 10 points to a "strongly agree" response was preferred. Conversely, a "strongly disagree" answer warranted 0 points. By subtracting 1 from each odd-numbered question's response, the minimum score was guaranteed to be 0. To ensure the maximum score for each question was 10, the resulting score was multiplied by 2.5.

On the other hand, for even-numbered questions employing a negative tone, assigning 0 points to a "strongly agree" response was optimal. Similarly, a "strongly disagree" response also earned 0 points. Subtracting the score for each question from 5 assured a minimum score of 0. By multiplying the resulting score by 2.5, the maximum score for each question was guaranteed to be 10. The resulting number was then compared to the scale represented by the provided figure.



*Figure 3. SUS Table Score*

The application would only proceed to the final implementation stage of the web app if the SUS result of the beta test was grade B or above. If the SUS result was lower, another round of beta testing would be conducted until the SUS result reached grade B or A.

### Deployment and Maintenance

Once the closed beta version had undergone testing and been approved by stakeholders, the system was deployed by the proponents. Initially, it was remotely hosted on a web hosting site, and subsequently, it was deployed on the CMU MIS server. Individuals could access both versions of the system through designated links.

The closed beta version of the system went through rigorous testing by selected users to identify any remaining issues. Feedback from this beta testing was incorporated to optimize system performance and usability.

After the closed beta testing phase, stakeholders, likely including project advisors and department leadership, evaluated the system and provided final approval for deployment. Their sign-off marked the end of development testing. With this approval, the proponents deployed the first live version of the system remotely on a web hosting platform. This allowed broader access for additional testing.

Later, the system was deployed on CMU's MIS server, indicating its complete transition to production use. Throughout this process, authorized users were granted access via specified links, allowing the proponents to control the deployment rollout. The staged release enabled systematic progress toward full implementation.

## 4. 3 System Analysis



### Flowcharts

Appendix Figure 4.a. It illustrated Organizations could customize their profiles to promote individuality and uniqueness, communicate their mission, goals, events, and initiatives, and provide valuable insights for college administration and stakeholders. This feature empowered organizations to maintain an updated and tailored representation, reinforcing their connection and relevance to their college.

Appendix 5.a, the organization had the capability to manage roles based on its members' positions, with three roles currently in place. These roles signified different responsibilities within the organization. The organization had the authority to assign and modify these roles according to its specific needs. By managing roles, the organization could grant appropriate access rights and permissions to individuals based on their positions, facilitating efficient collaboration and effective task execution.

Appendix 6.a, illustrated the import process allowed organizations to add multiple students' information in bulk, saving time and effort compared to manually entering individual student details. The imported student list typically included essential information such as student names, IDs, contact information, and any additional relevant data.

Appendix Figure 7.a, Figure 8, and Figure 9, illustrated that the organizations had the ability to add, edit, and delete events within the system, providing them with full control over their event management process. When adding an event, organizations could input relevant details such as the event name, description, date, time, location, and other pertinent information. When editing events, organizations could make adjustments to event details, ensuring that event information remained accurate and up to date. Finally, when an event needed to be canceled or was no longer relevant, organizations could remove the event from the system to ensure attendees were properly notified and any associated resources or registrations were appropriately managed.

Appendix Figure 10.a and Figure 11.a, allowed the organizations to have the ability to add, edit, and delete events evaluation forms to gather feedback, opinions, and assessments from event participants. The form could cover various aspects of the event, such as organization, event planning, content quality, speaker performance, logistics, and overall participant satisfaction. Additionally, organizations had the option to delete the evaluation form if it became obsolete or no longer relevant.

Appendix 12.a, The organization utilized the attendance tracking feature during events to monitor participant presence and identify individuals with fines. This functionality helped the organization keep track of financial obligations related to event attendance and manage fine collection efficiently. By recording attendance, the organization could match individuals' presence with the event's fine policy, which was established based on predetermined criteria. This allowed the organization to accurately identify individuals who had incurred fines based on their attendance records.

Appendix 13.a. illustrated that the organization's system had an automated fine calculation feature that automatically calculated fines based on attendance records and predefined fine policies. This functionality streamlined the process of determining fines associated with event attendance and ensured accurate and consistent calculations.

Appendix Figure 14.a, the system offered students the ability to view their accountabilities, access announcements, and evaluate events within the organization. Students could easily see their financial responsibilities, stay informed about important updates, and provide feedback on events. These features promoted transparency, engagement, and accountability, enhancing the student experience within the organization.

Appendix Figure 15.a, the system supported login capabilities for three roles: admin, student, and attendance checker. Admins had elevated privileges and managed the system, while students accessed their individual accounts to view accountabilities, evaluate events, and receive announcements. Attendance checkers focused on managing attendance during events. Each role had specific permissions and functionalities, and the login process involved entering credentials, validating them, and granting access. The system enabled secure access and streamlined task performance for each user group.

### Entity Relationship Diagram

In Appendix Figure 16.b, the system included several tables, such as Roles, User\_Orgs, Organizations, Events, Attendance, Users, and Accountabilities, which stored relevant data. Each table served a specific purpose in organizing and managing student organization events and accountabilities.

The Roles table stored different roles within the system, such as "admin" or "student". The User\_Orgs table established relationships between users, organizations, and roles. The Organizations table held information about various organizations, including their names and descriptions. The Events table stored details about the events being organized, including event names, descriptions, dates, and locations.

The Attendance table tracked the attendance of users at events, recording the event ID, user ID, and date. The Users table contained information about system users, such as their usernames, passwords, and names. The Accountabilities table included details about student accountabilities, such as accountability names and associated amounts.

The system allowed the attendance checker to monitor attendance by accessing the Attendance table. Students could view their accountabilities through the Accountabilities table. Additionally, the admin had additional privileges, such as managing events, creating and editing evaluation forms, and calculating fines for students.

### Use Case Diagram

Appendix B. Figure 17, the use case diagram illustrates the interactions between three actors (Organization Admin, Student, and Attendance Checker) and the system. Each actor has specific use cases that define their actions and responsibilities within the system.

### Database Schema

### Data Dictionary

Table #. Data Dictionary of tbl\_users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_users** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| name | Varchar (191) | Not Null |  |  |
| email | Varchar (191) | Not Null |  |  |
| password | Varchar (191) | Not Null |  |  |

Table #. Data Dictionary of tbl\_roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_roles** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| role\_id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| name | Varchar (191) | Not Null |  |  |

Table #. Data Dictionary of tbl\_user\_organization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_user\_organization** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| student\_org\_id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| student\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| role\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| year\_level | Varchar (191) | Not Null |  |  |

Table #. Data Dictionary of tbl\_organizations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl \_organizations** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| org\_id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| name | Varchar (191) | Not Null |  |  |
| description | text | Not Null |  |  |

Table#. Data Dictionary of tbl\_events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_events** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| event\_id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| name | Varchar (191) | Not Null |  |  |
| location | Varchar (191) | Not Null |  |  |
| description | Text | Not Null |  |  |
| require\_attendance | Tinyint (4) | Not Null |  |  |
| attendance\_count | Int (11) | Not Null |  |  |
| evaluation\_status | Tinyint (1) | Not Null |  |  |
| attendance\_status | Tinyint (1) | Not Null |  |  |
| org\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |

Table #. Data Dictionary of tbl\_attendance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_attendance** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| user\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| org\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| event\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| officer\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| session | Int (11) | Not Null |  |  |

Table #. Data Dictionary of tbl\_evaluation\_form\_answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_evaluation\_form\_answer** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| event\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| user\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| org\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |

Table #. Data Dictionary of tbl\_attendance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_events** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| user\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| ord\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |
| event\_id | Bigint (20) | Foreign Key, Not Null, Unsigned |  |  |

Table #. Data Dictionary of tbl\_failed\_jobs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_failed\_jobs** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| uuid | Varchar (191) | Not Null |  |  |
| connection | Bigint (20) | Not Null |  |  |
| queue | Text | Not Null |  |  |
| paylaod | Long Text | Not Null |  |  |
| exception | Long Text | Not Null |  |  |

Table #. Data Dictionary of tbl\_fines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_fines** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| Student\_id | Bigint (20) | Not Null, Unsigned |  |  |
| Student\_org\_id | Bigint (20) | Not Null, Unsigned |  |  |
| Amount | Double | Not Null |  |  |

Table #. Data Dictionary of tbl\_organization\_accountabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_organization\_accountabilities** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| org\_id | Bigint (20) | Not Null, Unsigned |  |  |
| accountability\_name | Varchar (191) | Not Null, Unsigned |  |  |
| amount | Double | Not Null |  |  |

Table #. Data Dictionary of tbl\_paid\_accountabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tbl\_paid\_accountabilities** | | | | |
| **Field Name** | **Type** | **Constraints** | **Description** | **Reference Table** |
| id | Bigint (20) | Primary Key, Not Null, Unsigned |  |  |
| student\_id | Bigint (20) | Not Null, Unsigned |  |  |
| student\_org\_id | Bigint (20) | Not Null, Unsigned |  |  |
| accountability\_name | Varchar (191) | Not Null |  |  |
| amount | Double | Not Null |  |  |

## 4.4 System Requirements Specification

The proposed system is a desktop/laptop software application to help student organizations manage events, communications, evaluations, and accountability. It will provide a comprehensive solution to enhance organization efficiency and effectiveness. The system will be developed using the Waterfall methodology with Laravel, Xampp, Bootstrap, and Vue.js. Based on the project description and methodology using the Waterfall methodology approach, below is the system requirements specification:

**Functional Requirements**

|  |  |
| --- | --- |
| Requirement  No. 1 | Login |
| Priority | High |
| Purpose | Allows users to access the system through their designated accounts. |
| Input | The both student organization officer and student user will input their username and password. |
| Operations | After submitting the login form, the system will first check whether the user exists or not inside the system by checking the username. If it does, it will then check the password if it is correct or not. |
| Output | If a user does not exist, it will output an error. If the password is also not correct, an error will also be given. If all credentials are correct, the user will be redirected to the main page of each module. |

|  |  |
| --- | --- |
| Requirement  No. 2 | Create Event |
| Priority | High |
| Purpose | Allows student organization to create an event with complete details. |
| Input | The student organization will input the event name, start date, end date, event time from start to end, location, description, and requiring of attendance. |
| Operations | Every change in the input will be checked individually. Once all inputs are done and the form is submitted, it will then be saved into the database and it will be shown on the announcement menu. |
| Output | Once the event created has been saved, the system will then notify that the event has been successfully created and can now be modified. |

|  |  |
| --- | --- |
| Requirement  No. 3 | Edit Event |
| Priority | Medium |
| Purpose | This will allow the student organization to edit an event if the details are wrong or lacking. |
| Input | The student organization will be able to edit the event name, start date, end date, event time from start to end, location, description, and requiring of attendance. |
| Operations | Every change in the input will be checked individually. Once all inputs are done and the form is submitted, it will then be saved into the database and it will be shown on the announcement menu. |
| Output | Once the edited event has been saved, the system will then notify that the event has been successfully edited and can now be modified. |

|  |  |
| --- | --- |
| Requirement  No. 4 | Delete Event |
| Priority | Low |
| Purpose | Allows the student organization to delete an event. |
| Input | The student organization will choose what event to delete. The student organization will then click the button delete. |
| Operations | Completely deletes the event selected. |
| Output | Once the student organization have deleted a certain event, the system will then notify that event has been successfully deleted. |

|  |  |
| --- | --- |
| Requirement  No. 5 | Record Attendance |
| Priority | High |
| Purpose | Allows the student organization to record attendance using QR code scanner. |
| Input | The student organization will scan the QR code of the students |
| Operations | The student organization will utilize QR code scanning to track attendance. Each student will be assigned a unique QR code generated from their ID number. Once, the student organization will start the attendance, students that will arrive at the event, they can scan their personal code to register their presence. The system will automatically save these attendance records to the database. The student organization can then view attendance statistics through their administration portal, while students can access their individual attendance history in their attendance menu. |
| Output | Once the student organization’s end time has met, they will not be allowed to scan. |

|  |  |
| --- | --- |
| Requirement  No. 6 | Download Recorded Attendance |
| Priority | Medium |
| Purpose | Allow the student organization to download a copy of the recorded attendance. |
| Input | Once the student organization is done with the checking of attendance, the student organization will then download a copy. |
| Operations | The student organization can download attendance reports for events. Students’ check-in by scanning QR codes, which automatically records participation data. The system tallies checked-in students and generates summary reports showing event participation numbers. Organizers can conveniently access these downloadable reports. |
| Output | The student organization can download event attendance reports summarizing the number of students who signed in via QR code scans. |

|  |  |
| --- | --- |
| Requirement  No. 7 | Print Recorded Attendance |
| Priority | Medium |
| Purpose | Allow the student organization to print a copy of the recorded attendance. |
| Input | Once the student organization is done with the checking of attendance, the student organization will then print a copy. |
| Operations | The student organization can print attendance reports for events. Students’ check-in by scanning QR codes, which automatically records participation data. The system tallies checked-in students and generates summary reports showing event participation numbers. Organizers can conveniently access these printable reports. |
| Output | The student organization can print event attendance reports summarizing the number of students who signed in via QR code scans. |

|  |  |
| --- | --- |
| Requirement  No. 8 | Add Student |
| Priority | Medium |
| Purpose | Allow the student organization to manually add a student. |
| Input | The student organization can manually add a student’s name, ID, and year level once they will click the “Add Student” button. |
| Operations | The system allows the student organization to manually add a student then they will save it. |
| Output | The system confirms successful addition of student details and saves the updated student list in the database. |

|  |  |
| --- | --- |
| Requirement  No. 9 | Upload Student List |
| Priority | High |
| Purpose | Allow the student organization to upload a list of enrolled students. |
| Input | The student organization will upload a list of enrolled student members. |
| Operations | The system allows the student organization to bulk upload student members in a provided file format to update enrollment records. |
| Output | Once the student organization has uploaded the list of enrolled students, it will then notify successfully uploaded. |

|  |  |
| --- | --- |
| Requirement  No. 10 | Print Evaluation Results |
| Priority | Medium |
| Purpose | Allows the student organization to print results from the event’s evaluation form. |
| Input | The student organization can print summary reports of event evaluation form responses to review attendee feedback and identify improvements for future events. |
| Operations | The student organization will be able to close the evaluation form from a specific time. Once the student organization is satisfied with the numbers of students who have answered the form, they will then be able to print a copy of results. |
| Output | The student organization will be able to print evaluation results with event details directly to a printer. |

|  |  |
| --- | --- |
| Requirement  No. 11 | Set Accountabilities |
| Priority | High |
| Purpose | This allows the student organization to add accountabilities and description. |
| Input | The student organization can assign varying numbers of accountabilities for different events, enabling students to determine their specific responsibilities for each event. |
| Operations | The student organization can easily monitor accountability fulfillment for each event, and students can easily view any fines incurred for their responsibilities across different events. |
| Output | Once the student organization have set the accountability for a certain event, it will then notify it has been successfully saved. The system will then tally the total accountability fulfillment for each student. |

|  |  |
| --- | --- |
| Requirement  No. 12 | Accountabilities Lists |
| Priority | High |
| Purpose | This page allows the student organization to track students’ accountabilities. |
| Input | The student organization can track students’ accountabilities easily. |
| Operations | When the student organization sets the time for paying of accountabilities, the student organization will search for the name of the student then it will show the student’s accountability. As for the students, the students will be able to determine their accountabilities easily by visiting their profile. |
| Output | Once the student organization have set the accountability for a certain event, it will then notify it has been successfully saved. The system will then tally the total accountability fulfillment for each student. |

|  |  |
| --- | --- |
| Requirement  No. 13 | Announcement |
| Priority | Medium |
| Purpose | This will allow the student organization to create an announcement after the event details have been created. |
| Input | The student organization will input the event details. |
| Operations | The student organization publicizes upcoming events by creating event details and announcements. Once the event details such as date, time, location, and description are finalized, they are posted in the student announcement menu. This allows students to conveniently see what student organization events they need to participate in. |
| Output | Once the student organization finalizes the event details, the information is saved in the system and automatically published in the student announcement menu. |

|  |  |
| --- | --- |
| Requirement  No. 14 | Evaluation |
| Priority | Medium |
| Purpose | This will allow the student organization to start an evaluation automatically after an event. |
| Input | After each event concludes, the student organization conducts an evaluation using a standardized form from the Office of Student Affairs (OSA). |
| Operations | After each event, the student organization initiates the standardized online evaluation form for students to provide feedback. This digital form, following guidelines from the Office of Student Activities (OSA), is automatically generated - eliminating the need for manual creation. Students can conveniently access the evaluation by logging into their profile and fill it up. |
| Output | Once the student have succ |

### Non-Functional Requirements

Non-functional requirements for the Organization Admin, Student, and Attendance Checker roles could include:

1. Performance:

* The system should respond to user actions quickly and provide a seamless user experience.
* The system should be able to handle a large number of events, users, and evaluations without significant performance degradation.

1. Security:

* User authentication and authorization mechanisms should be in place to ensure that only authorized individuals can access and perform actions within the system.
* The system should implement secure data transmission protocols (e.g., HTTPS) to protect sensitive information during communication.

1. User Reliability:

* The system should be available and accessible to users with minimal downtime or disruptions.
* Adequate backup and recovery mechanisms should be in place to protect against data loss and ensure system availability.
* The system should have error handling and fault tolerance mechanisms to gracefully handle errors and recover from failures.

1. Scalability:

* The system should be able to accommodate a growing number of users, events, and evaluations without significant performance degradation.
* The system should be designed to scale horizontally or vertically to meet increased demands.

1. Usability:

* The user interfaces should be intuitive, user-friendly, and easy to navigate.
* Clear instructions and feedback should be provided to guide users through different actions and processes.
* The system should support multiple devices and screen sizes to ensure accessibility from various platforms.

1. Maintainability:

* The system should be modular, well-organized, and follow coding best practices to facilitate future enhancements, bug fixes, and maintenance.
* Documentation should be provided to assist administrators and developers in understanding and maintaining the system.
* Passwords and other sensitive data should be securely stored using encryption techniques.

1. Integration:

* The system should be able to integrate with other existing systems or APIs, such as email services or notification systems, to enable seamless communication and data exchange.

These non-functional requirements ensure that the system is efficient, secure, reliable, user-friendly, scalable, maintainable, and capable of integrating with other systems as required.

## **Gantt Chart**

# CHAPTER V

**RESULTS AND DISCUSSION**



## Results

The objective of this capstone project is to create, develop, test, and deploy the Central Mindanao University Student Organization Event Monitoring and Accountabilities Management System. Specifically, it aimed to: (1) develop a system that ensures transparency in managing and tracking financial accountabilities of the student organization; (2) create a system for recording attendance using QR-code; (3) create a system for comprehensive event monitoring; (4) develop an integrated evaluation to facilitate the collection and analysis of feedback from attendees.

## System Process

The table below lists the advantages and disadvantages of the general system processes between the current and the developed system in terms of time and transparency.

Table. # Advantages and disadvantages of the current system’s process in terms of time and transparency

|  |  |  |
| --- | --- | --- |
| **Transaction** | **Current System** | |
|  | Advantages | Disadvantages |
| Event Announcement | The organization announced events on its Facebook group page and via Facebook messenger. | The user struggled to find the needed event post on the organization's Facebook page amidst an influx of unrelated content. The high volume of posts made it hard to locate key details about the specific event. |
| Accountabilities | None | The current system lacked transparency and was difficult to track due to numerous forms needing verification. |
| Attendance | Easy to monitor. | Checking attendance was time-consuming and finding the person in charge was a hassle. |
| Evaluation | Evaluation is done online. | The large volume of posts and indistinct format made it difficult to distinguish between different events on the organization's Facebook page. |

Table # Advantages and disadvantages of the CMU STORE-AMS’s system process in terms of time and transparency

|  |  |  |
| --- | --- | --- |
| **Transaction** | **CMU STORE-AMS** | |
|  | Advantages | Disadvantages |
| Event Announcement | The organization will create dedicated event pages with details that serve as announcements. | To schedule recurring events, the student organization must create separate but identical event pages in the online system. Internet access is needed to use the system. |
| Accountabilities | The system transparently tracks fees, fines, and attendance, enabling students to easily check payment status and missed events. | Internet access is required to use the system. |
| Attendance | The new system simplifies attendance tracking. Students are issued QR codes which the organization scans for automated check-in, reducing hassle. | Internet access is required. Without their QR code, students must be manually checked in by ID number, which is more cumbersome. |
| Evaluation | With CMU STORE-AMS, the student organization can enable the evaluation form after the event for students to provide feedback. | Requiring internet access could be a hindrance to getting high response rates on evaluation forms compared to traditional paper forms. |

## Student Organization Module



## Login Page

Figure # Login Page Student Organization User Interface

The figure shows the login interface for the student organization module of CMU STORE-AMS. Users must input their username and password to access the main page. The system will display an alert for incorrect login details. If a session exists or the user has logged in recently, they will be redirected to the main page automatically.

## Student Organization’s Roles

Figure # Student Organization’s Role

The figure illustrates the student organization officer's ability to select which page to access in the system as an admin of their student organization. The student organization officer can access the admin side of the system for their organization, and they can also access the student side of the system if they are involved in multiple organizations. This allows the officer to toggle between the admin view for their specific organization and the general student view to manage responsibilities across multiple organizations.

## Student Organization Dashboard

Figure #. Student Organization Dashboard User Interface

The figure depicts the landing page for a student organization officer who has logged into the admin side of the system. It displays the event timeline, announcement section, and calendar where upcoming events are also visible. This centralized dashboard allows the officer to easily view and manage the organization's schedule, announcements, and other activities.

## Attendance Schedule

Figure #. Attendance Schedule User Interface

As shown in the figure, the Attendance menu has two options: Schedule and Record. The Schedule page displays when it is selected, showing scheduled event cards that contain the event details. A calendar is also provided on the Schedule page, allowing student organization officers to easily see and track which events are coming up. The calendar enables at-a-glance tracking of planned events.

## Record Attendance

Figure #. Record Attendance User Interface

The figure shows the Record page from the Attendance menu. On this page, attendance records are displayed via cards for each event held. The student organization officer opens attendance recording when an event starts, and it closes automatically at the set time. The cards list details like the event name, date, and attendance count. This gives the officer easy access to attendance statistics for previous events.

## Student Member Module

## Login Page

Figure # Login Page for Student Member User Interface

The figure shows the login interface for the student member module of CMU STORE-AMS. Users must input their username and password to access the main page. The system will display an alert for incorrect login details. If a session exists or the user has logged in recently, they will be redirected to the main page automatically.

## Repository Data

## Beta Testing

The CMU STORE-AMS beta testing was divided into two categories: student organization testing and student member testing. The student organization testing focused on evaluating the system's capabilities for managing student organizations. Four respondents participated in this category.

The student member testing focused on evaluating the system's capabilities related to individual student members. Forty-four respondents participated in this category.

Table #. Student Beta Testing Category Respondents.

|  |  |
| --- | --- |
| **STUDENT BETA TESTING CATEGORY- RESPONDENTS** | |
| College of Information Sciences and Computing | 44 Students |

## System Usability Scale Survey

Table #. Count and Standard Deviation of Values in the SUS Survey of the Current System under the Student Category

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **QUESTIONS** | **COUNT** | | | | | | **STANDARD DEVIATION** |
|  | **SD** | **D** | **N** | **A** | | **SA** |  |
| 1 | 7 | 6 | 5 | | 15 | 11 | 3.71 |
| 2 | 6 | 2 | 15 | | 11 | 10 | 4.44 |
| 3 | 4 | 8 | 12 | | 10 | 10 | 2.71 |
| 4 | 4 | 5 | 14 | | 11 | 10 | 3.76 |
| 5 | 3 | 3 | 14 | | 14 | 10 | 4.96 |
| 6 | 4 | 6 | 10 | | 14 | 10 | 3.49 |
| 7 | 4 | 1 | 12 | | 17 | 10 | 5.70 |
| 8 | 3 | 6 | 7 | | 15 | 13 | 4.49 |
| 9 | 5 | 7 | 5 | | 18 | 9 | 4.83 |
| 10 | 4 | 8 | 11 | | 14 | 7 | 3.43 |

## SUS Survey Score