

ATENEO DE ZAMBOANGA UNIVERSITY
College of Science, Information Technology, and Engineering
Computer Science Department
Academic Year 2025 — 2026



Social Development Unit Project Monitoring System

In final fulfillment of the requirements in
Database Management Systems (CIT.010)

Jerome P. Arawangsa

Khan O. Fernandez

Danna Mishna I. Lledo

Kenneth Clyde A. Que

Jed Edison I. Tenorio

CIT.010-A

BS Computer Science - III

December 2025

ACKNOWLEDGEMENT

We, the developers of the *Social Development (SDU) Unit Project Monitoring System*, respectfully extend our gratitude to the individuals and offices whose support and cooperation were integral to the successful completion of this project.

We express our sincere appreciation to the **Social Development Unit (SDU) of Ateneo de Zamboanga University**, particularly **Mrs. Aurora Gonzales**, Unit Director, for granting us the opportunity to examine their existing processes and for providing essential information that informed the development of our system. Her willingness to collaborate and share valuable insights significantly contributed to the accuracy and relevance of our work.

We also extend our deepest thanks to the **Program Coordinator, Office Directors, and staff of the SDU's six offices**; Ateneo Center for Culture and the Arts (ACCA), Ateneo Center for Environment and Sustainability (ACES), Ateneo Center for Leadership and Governance (ACLG), Ateneo Learning and Teaching Excellence Center (ALTEC), Ateneo Peace Center (APC), and the Center for Community Extension Services (CCES). Their thoughtful feedback regarding current reporting practices played a crucial role in defining system requirements and ensuring that our proposed solution reflects the operational needs of the unit.

Our profound gratitude goes to our instructors, **Mrs. Precious T. Opinion, MSCS** (Database Management Systems – CIT.009) and **Ms. Clairizza V. Arcilias** (Web Systems and Technologies – CIT.010), whose guidance, technical expertise, and consistent support enabled us to apply core concepts effectively throughout the development process. Their professionalism and dedication greatly enriched our learning experience.

We also acknowledge our classmates, peers, and mentors for their valuable feedback, and our families and friends for their patience and unwavering support throughout this project.

This undertaking reflects the collective effort of all who contributed their time, expertise, and support, inspiring us to pursue excellence not only in academics but in all that we do — *magis*

TABLE OF CONTENTS

| | |
|---|-----------|
| PLANNING | 3 |
| A. Project Context | 3 |
| I. About the organization | 3 |
| II. Problems, issues, and Challenges identified | 4 |
| III. Significance of the System to be developed | 5 |
| IV. Planning Method | 6 |
| B. Enterprise Modeling | 8 |
| I. Data Entity-Business Function Matrix | 8 |
| II. Current System Analysis | 10 |
| Analysis | 13 |
| A. Proposed System Analysis | 13 |
| B. Conceptual Data Modeling | 16 |
| I. Entity Relationship Diagram | 16 |
| II. Business Rules | 19 |
| C. Query Statements | 20 |
| I. System Functions | 20 |
| II. SQL Syntax | 23 |
| DESIGN | 31 |
| A. User Interface | 31 |
| I. System-Wide | 31 |
| a. Log-in Page | 31 |
| b. Search Component Page | 32 |
| c. Messaging Component Page | 34 |
| II. Unit Director | 35 |
| a. Overview Dashboard | 35 |
| b. Reports Dashboard | 36 |
| c. Calendar Dashboard Page | 38 |
| d. Settings Dashboard | 39 |
| 1. Program Management Sub-Dashboard | 40 |
| 2. User Management Sub-Dashboard | 40 |
| c. Messages Management Sub-Dashboard | 41 |
| e. Help and Documentation Dashboard | 42 |
| III. Offices | 43 |

| | |
|-------------------------------------|-----------|
| a. Overview Dashboard | 43 |
| b. Projects Dashboard | 45 |
| 1. Project Detail Modal View | 46 |
| 2. Project Form | 46 |
| c. Help and Documentation Dashboard | 47 |
| II. Report Design | 49 |
| A. Single Project/Activity Report | 49 |
| B. Consolidated Report | 50 |
| IV. TESTING | 52 |
| I. Testing Method | 51 |
| II. Testing Results | 52 |
| III. Training of User | 53 |
| V. IMPLEMENTATION | 55 |
| I. User Support Services | 55 |
| II. Deployment Documentation | 57 |
| A. Single Project/Activity Report | 57 |
| B. Consolidated Report | 59 |
| A. Single Project/Activity Report | 61 |
| REFERENCES | 65 |

PLANNING

This section defines the system by outlining the SDU's structure and mission. It analyzes the problems of the current reporting process, identifying the need for automation and standardization. Finally, it details the Planning Method, utilizing Enterprise Modeling to align the system design with organizational roles.

A. Project Context

I. About the organization

The Social Development Unit (SDU) of Ateneo de Zamboanga University (AdZU), established in 2007 under the presidency of Fr. Antonio Moreno, SJ, serves as the university's central arm for advancing the common good through transformative social development work. Guided by its core commitments to community engagement, issue-based advocacy, and fostering dialogue on critical societal themes, the SDU implements its mission through a range of programs carried out by its specialized offices. These initiatives are deliberately aligned with the Seventeen (17) Sustainable Development Goals (SDGs), which constitute a universal set of goals, targets, and indicators that member states are expected to use in shaping their development agendas and policy priorities over a fifteen-year period, according to the United Nations (2025).

Over time, the SDU's structure has evolved, and under the current leadership of Unit Director Mrs. Aurora Gonzales, the unit now operates six offices: the Ateneo Center for Culture and the Arts (ACCA), Ateneo Center for Environment and Sustainability (ACES), Ateneo Center for Leadership and Governance (ACLG), Ateneo Learning and Teaching Excellence Center (ALTEC), the Ateneo Peace Center (APC), and the Center for Community Extension Services (CCES). Each office serves a distinct yet interconnected role—ranging from community empowerment and artistic development to environmental advocacy, governance education, peace promotion, and capacity building through research and training.

Together, these centers embody the SDU's overarching purpose: to cultivate meaningful partnerships, promote social justice, and uphold AdZU's role as a catalyst for sustainable and inclusive development in Western Mindanao.

II. Problems/ Issues/ Challenges identified

In project management and social development practice, formal documentation of projects and activities is considered a standard for ensuring accountability, transparency, and efficiency according to Funds for NGOs (2024). Organizations that fail to implement structured reporting mechanisms often experience delays, errors, and challenges in monitoring performance effectively.

In alignment with this standard, the Social Development Unit (SDU) of Ateneo de Zamboanga University serves as the central body for six (6) offices, each responsible for implementing multiple programs and its associated projects or activities. While, according to its staff, a centralized digital repository exists, the reporting and documentation process remains largely manual. Each office submits reports in its own format, often using spreadsheets, word-processing documents, or other basic file types. Locating and reviewing these reports requires manually searching the repository, which leads to inconsistencies in terminology, structure, and completeness. As highlighted by Henderson (2025), cross-office comparison becomes complicated, and the consolidation process is slowed, increasing the likelihood of errors such as missing or duplicate data.

In addition, the current process relies heavily on informal digital communication, such as group chats or messaging apps with its manual reporting. Because these channels are unstructured, reminders, updates, or notifications are easily missed or overlooked, leading to inconsistent coordination and delayed reporting. This is evident as per Rao (2025) that organizations that still rely on informal workflows and standalone tools report increased inefficiencies, administrative burden, and fragmented data. Since offices follow their own schedules and staff are frequently occupied, the lack of an integrated communication and reporting system reduces accountability and complicates monitoring. Studies by Cavicchi & Vagnoni (2022) on nonprofit organizations show that structured

digital information systems enhance internal accountability, improve coordination, and support timely, accurate reporting. Without real-time visibility or automated notifications, the Unit Director cannot quickly detect overdue reports or emerging issues, making oversight reactive rather than proactive and increasing the risk of errors or omissions.

These challenges underscore the need for a structured intervention in the form of a project monitoring system. A centralized, automated system would standardize report submission, provide timely notifications and updates, enhance coordination between the Unit Director and offices, and enable real-time tracking of projects or activities. This aligns with the findings of Luther (2024), who reported that automated reporting systems improve accuracy, reduce errors, enhance accountability, and streamline coordination across multiple reports. Implementing such a system would therefore directly address existing inefficiencies, strengthen accountability, minimize errors, and improve the overall operational capacity of the SDU.

III. Significance of the System to be developed

The development of a web-based centralized project monitoring system holds significant value for the SDU and its stakeholders by addressing the existing challenges in documentation, coordination, and oversight. The stakeholders are the following:

1. ***Unit Director*** – the system provides comprehensive, real-time visibility of all offices and their respective projects or activities. It allows reports to be accessed either as a consolidated report for the month or individually. This flexibility simplifies monitoring and evaluation, tracks submission statuses, and records the alignment of projects with the SDGs. The organized overview enhances strategic decision-making, facilitates timely interventions, and enables the preparation of well-structured reports for both internal review and presentation to external stakeholders.

2. ***Offices*** – For each of the six (6) SDU offices, the system offers a standardized and secure platform for submitting project or activity reports, ensuring consistency in terminology, format, and structure. By reducing the time and effort spent on

organizing or correcting data, office staff can focus on implementing and improving programs. Additionally, the system's dashboard allows offices to track project timelines, monitor progress, check the status of their projects, and plan future activities effectively, ensuring coherent and accurate contributions to the centralized system.

3. ***Social Development Unit*** – improved operational efficiency through the centralization of information. Automation reduces manual errors, prevents duplicated or misplaced files, and enforces standardized reporting practices. In addition, the system provides an integrated communication platform focused specifically on projects and activities, allowing offices to coordinate, share updates, and address issues efficiently. Together, these improvements strengthen the SDU's ability to coordinate, monitor, and evaluate initiatives, ensuring that programs align with organizations' and university's goals, and the SDGs.
4. ***Ateneo de Zamboanga University*** – from the availability of well-organized, standardized, and verified reports generated by the SDU. The system supports the university's internal planning, strategic reviews, and documentation requirements while facilitating compliance with external stakeholders, such as accrediting bodies and government regulators. By providing transparent, accurate, and comprehensive reports, the system enhances institutional accountability and enables the university to demonstrate its social development efforts effectively to both internal and external audiences.

IV. Planning Method

The project adopted an *iterative development approach*, in which the system was developed in repeated cycles, allowing for continuous refinement based on stakeholder feedback. The development process began with gathering initial requirements from the SDU to identify the challenges and limitations of the existing reporting process.

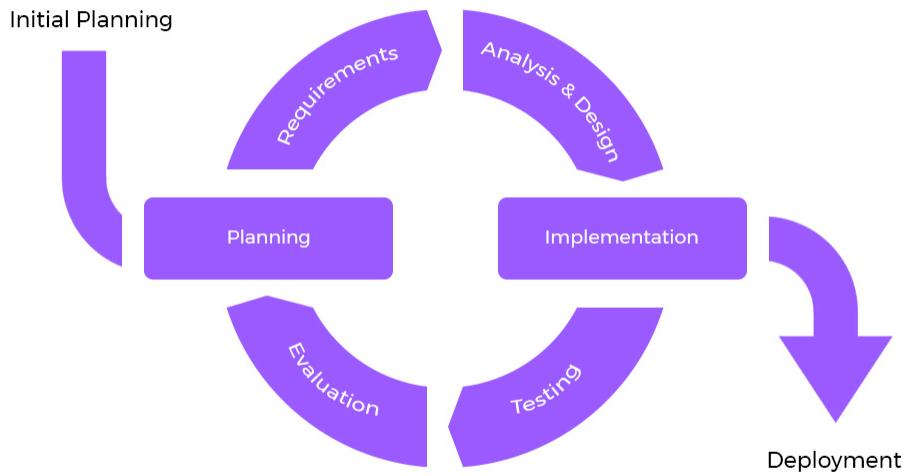


Figure 1. Iterative Development Approach Diagram.

Based on these requirements, the team created a preliminary version of the system, which included the basic reporting forms and centralized repository features. This version was then presented to the SDU for evaluation, where stakeholders provided feedback on functionality, usability, and alignment with organizational workflows. Subsequent iterations incorporated these suggestions, gradually enhancing system features, standardizing report formats, improving data validation, and integrating visual dashboards. This cycle of *develop* → *present* → *review* → *refine* was repeated approximately three (3) to four (4) times until the system met all project objectives and aligned with the SDU's operational needs.

a. Scope Management

The project scope was clearly defined to establish boundaries, objectives, and deliverables for the system. Key components of the scope included:

- Development of standardized web-based reporting forms for all six (6) SDU offices.
- Submission tracking and monitoring to identify completed and pending report submissions.
- Data validation rules to ensure consistency and accuracy in submissions.
- Visual dashboards to provide an overview of submission statuses and deadlines.

By establishing the scope at the outset, the team focused on features that addressed reporting inefficiencies without overcomplicating the system with storage or database management functionalities

b. Requirements Management

Requirements were gathered and managed using the following project management techniques:

- Stakeholder interviews and consultations to capture operational needs, expectations, and reporting challenges.
- Document analysis of existing reports, spreadsheets, and repository files to identify gaps and inconsistencies.
- Iterative validation through successive prototype reviews, allowing stakeholders to verify deliverables and provide corrective feedback.

This approach ensured that the system requirements were continuously validated and aligned with SDU objectives.

B. Enterprise Modeling

I. Data Entity-Business Function Matrix

A Kimball matrix, according to Leduc (2021), is a planning tool for data warehouses that maps business processes (rows) to dimensions (columns) to ensure consistent, integrated, and prioritized data mart development. The Data Entity-Business Function Matrix (or Responsibility Assignment Matrix - RAM)

shown in Figure 2 explicitly links the planned system's core Business Processes (rows) with the responsible Organizational Roles (columns). This matrix serves two primary functions: (1) to validate that all key stakeholders are integrated into the system design, and (2) to clearly define the access and responsibilities required for system functionality, which will inform the final security and user

| Process | Unit Director | Office Director | Activity | Notification | Program | Office Message |
|---|---------------|-----------------|----------|--------------|---------|----------------|
| <i>Monitor All Office Programs and Activities</i> | ✓ | | | | | |
| <i>Monitor Office Programs and Activities</i> | ✓ | ✓ | | | ✓ | |
| <i>Edit Submitted Programs and Activities</i> | | ✓ | | | | |
| <i>Monitor Office Programs and Activities</i> | ✓ | ✓ | ✓ | | ✓ | |
| <i>Remove Office Activities</i> | | ✓ | ✓ | | | ✓ |
| <i>Create Office Activities</i> | | ✓ | ✓ | | | ✓ |
| <i>View Activity Timeline</i> | ✓ | ✓ | ✓ | | | ✓ |
| <i>Manage Budgets</i> | ✓ | ✓ | | | | ✓ |
| <i>Generate Summary Reports (Monthly)</i> | | ✓ | | | | |
| <i>Update Status or Submitted Programs and Activities</i> | ✓ | | | | | |
| <i>Receive Notifications</i> | ✓ | ✓ | ✓ | ✓ | | ✓ |
| <i>Create Notifications</i> | | ✓ | ✓ | | | ✓ |
| <i>Create Message</i> | ✓ | ✓ | | | | ✓ |
| <i>Receive Message</i> | ✓ | ✓ | | | | ✓ |

role design.

Figure 2. Responsibility Assignment Matrix of SDU Project Monitoring System.

II. Current System Analysis

Grounded Theory (GT) is a qualitative methodology designed to develop a theory directly from systematically gathered and analyzed empirical data, without being constrained by pre-existing hypotheses (Shrestha, 2025). This approach employs iterative coding, constant comparison, and theoretical sampling to identify core patterns and relationships in the data, resulting in a context-sensitive, participant-grounded model (Akkaya, 2023; Abamonga, 2023). GT is particularly suitable for analyzing complex

organizational processes, such as the SDU's project monitoring system, where observed inefficiencies are not fully explained by existing, formal frameworks.

For this analysis, information was collected from SDU internal documentation (brochures, existing reports) and through semi-structured interviews with the Unit Director, Office Directors, and staff responsible for project reporting and monitoring.

Core Phenomenon and Causal Conditions

Applying the Grounded Theory method to this data allowed for a comprehensive examination of the SDU's operations, workflows, and recurring inefficiencies.

1. Open and Axial Coding (Identifying the Systemic Problem)

The analysis revealed three critical characteristics of the current workflow: (1) Fragmentation across the six specialized offices, (2) Lack of Standardization in reporting formats, and (3) Heavy Reliance on Informal Communication Channels (e.g., messaging apps). These factors interact to create a self-reinforcing cycle of inefficiency. Reports are independently prepared and submitted by different offices using varied formats (spreadsheets, word-processing documents), which, according to Alvarez-Sánchez et al. (2023), results in inconsistent content, gaps in data, and an increased risk of errors. Additionally, the submission of critical project reports through informal digital channels further limits accountability and auditability.

2. Selective Coding (Defining the Core Phenomenon)

Through iterative coding and constant comparison (comparing incidents across different offices to identify consistent patterns), the analysis distilled the Core Phenomenon: Fragmented Reporting Processes and Informal Communication Undermine Accountability and Operational Efficiency.

Conceptual Model of Current Workflow Failure

The GT analysis resulted in a conceptual model that explains how the existing conditions perpetuate inefficiency:

| Component | Manifestation |
|------------------------|---|
| Casual Conditions | <ul style="list-style-type: none"> 1. <i>Lack of Reporting Standardization</i> (Non-uniform templates and unstructured file types across offices). 2. <i>Manual Data Integration</i> (Consolidation relies on manual searching and merging of documents). 3. <i>Reliance on Informal Digital Channels</i> (Messaging applications for critical communication and follow-up). |
| Core Phenomenon | <p>Fragmented Reporting & Loss of Centralized Accountability.</p> |
| Intervening Conditions | <p>High staff commitment to field work limits time for administrative tasks; Unit Director's high oversight volume; complexity arising from varied program cycles across six distinct offices.</p> |
| Interaction Strategies | <ul style="list-style-type: none"> 1. <i>Reactive Oversight</i>: Monitoring triggered only by missed deadlines or explicit requests. 2. <i>Manual Retrieval</i>: Reports are located and reviewed by manually accessing disparate file repositories. 3. <i>Ad-hoc Communication</i>: Unscheduled and informal reminders sent via unstructured platforms. |
| Consequences | <p>Inconsistent follow-ups, delayed submission compliance, prolonged lead time for report consolidation, and elevated risk of data errors or omissions (Viteri-Sánchez &</p> |

| | |
|--|--------------------------|
| | Novillo-Villegas, 2025). |
|--|--------------------------|

Table 1. Conceptual Model of Current Workflow.

This analysis demonstrates that operational inefficiencies in SDU project monitoring are **systemic** rather than isolated, stemming from structural and procedural shortcomings. The conceptual model developed from this analysis provides a structured framework for understanding current workflow failures and serves as a foundation for evidence-based recommendations aimed at improving standardization, accountability, and overall operational efficiency through a centralized system.

ANALYSIS

This section examines the proposed system by defining its functional processes and data requirements. It presents the proposed system analysis to illustrate how information flows within the system and introduces the conceptual data model, including the Entity Relationship Diagram and business rules, and underlying SQL syntax to ensure data integrity and logical structure.

A. Proposed System Analysis

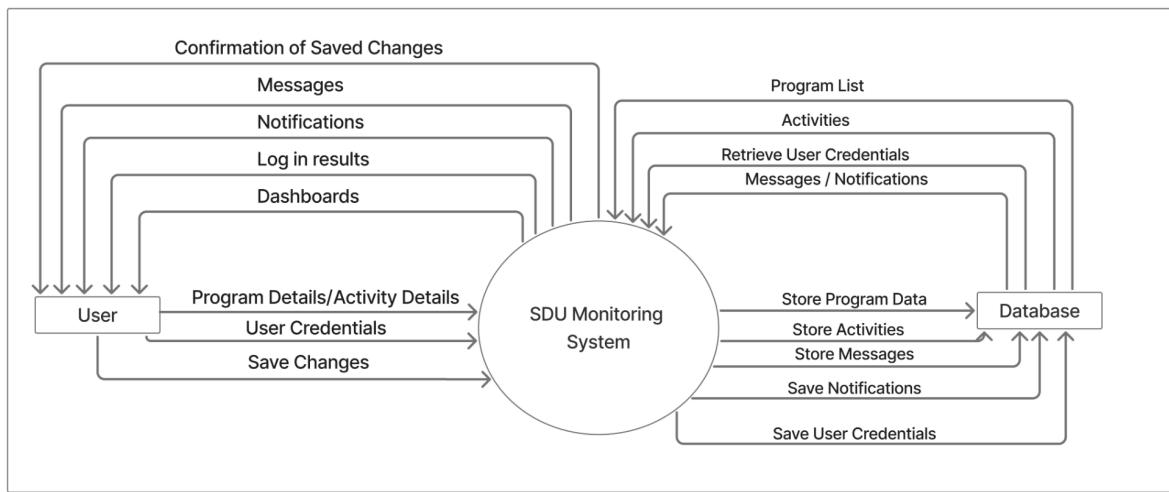


Figure 3. Data Flow Diagram (Level 0) of SDU Project Monitoring System.

The Data Flow Diagram visualizes the movement of information between the User, the SDU Monitoring System, and the backend Database. It demonstrates how the system acts as a central processor, accepting user inputs—such as credentials and status updates—and routing them for storage, while simultaneously retrieving archived data like dashboards and program lists to display back to the interface.

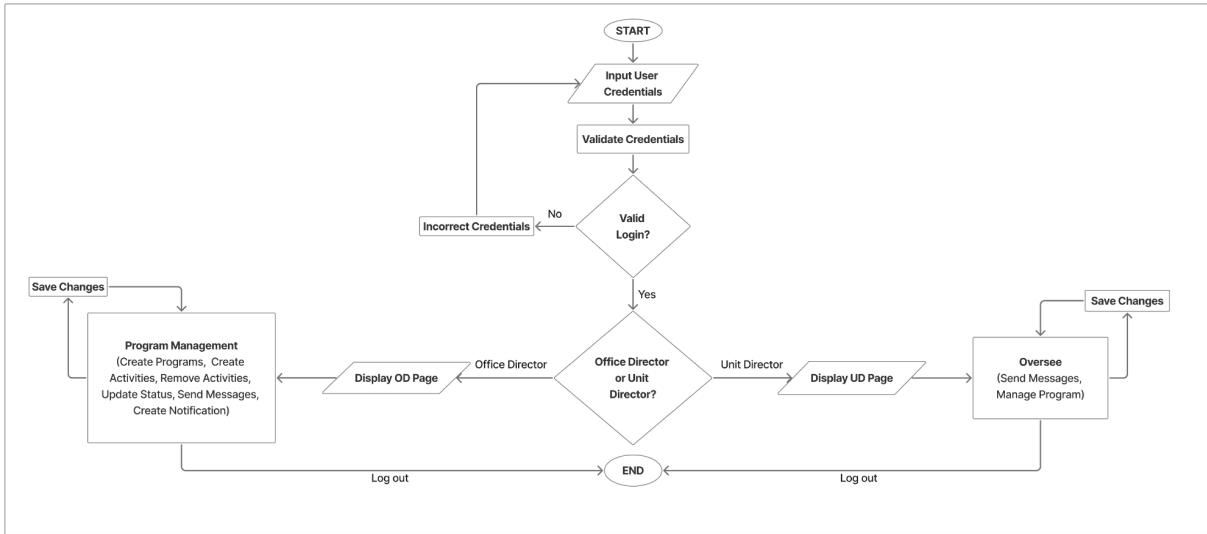


Figure 4. Flowchart of SDU Project Monitoring System.

The System Flowchart maps the operational logic and user journey, beginning with a secure authentication loop that validates credentials. Upon successful login, the system executes a decision process to route users based on their role—directing Office Directors to management interfaces and Unit Directors to oversight dashboards—before finalizing the session by saving changes to the database upon logout.

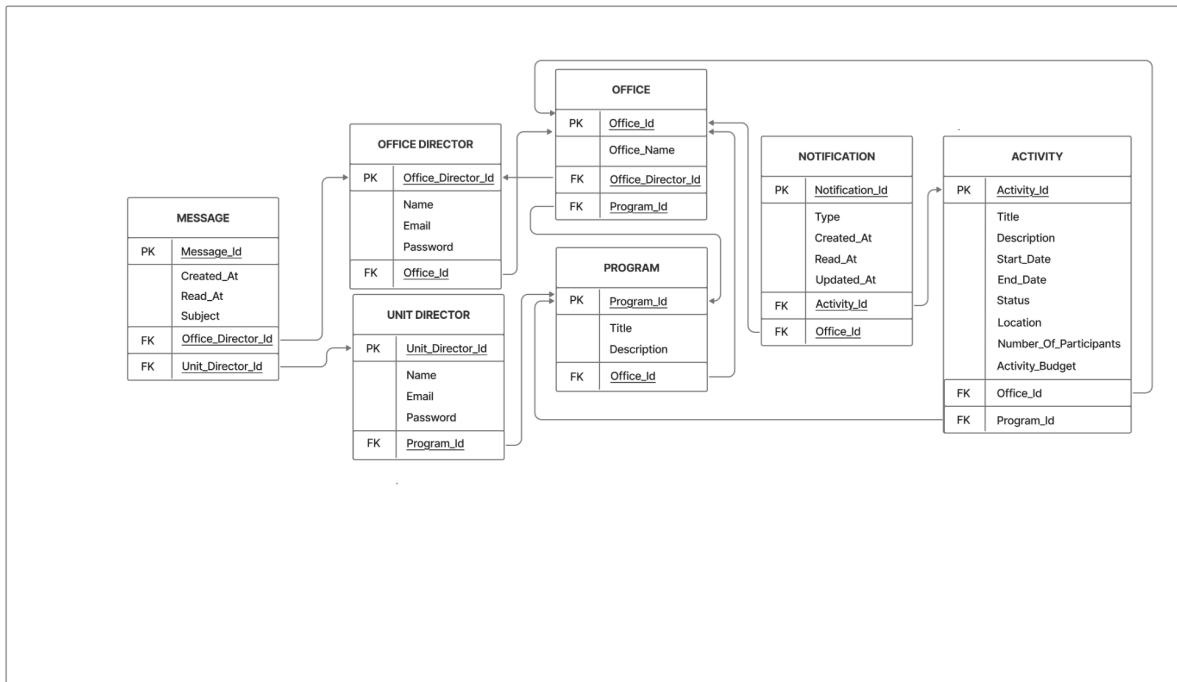


Figure 5. Relational Database Model of SDU Project Monitoring System.

The Database Schema defines the system's logical structure, organizing data into relational tables that link Offices, Programs, and Activities through primary and foreign keys. This design separates user roles into Office and Unit Directors to enforce access control, while also including specific entities for Messages and Notifications to maintain a structured history of internal communication and alerts.

B. Conceptual Data Modeling

I. Entity Relationship Diagram (ERD)

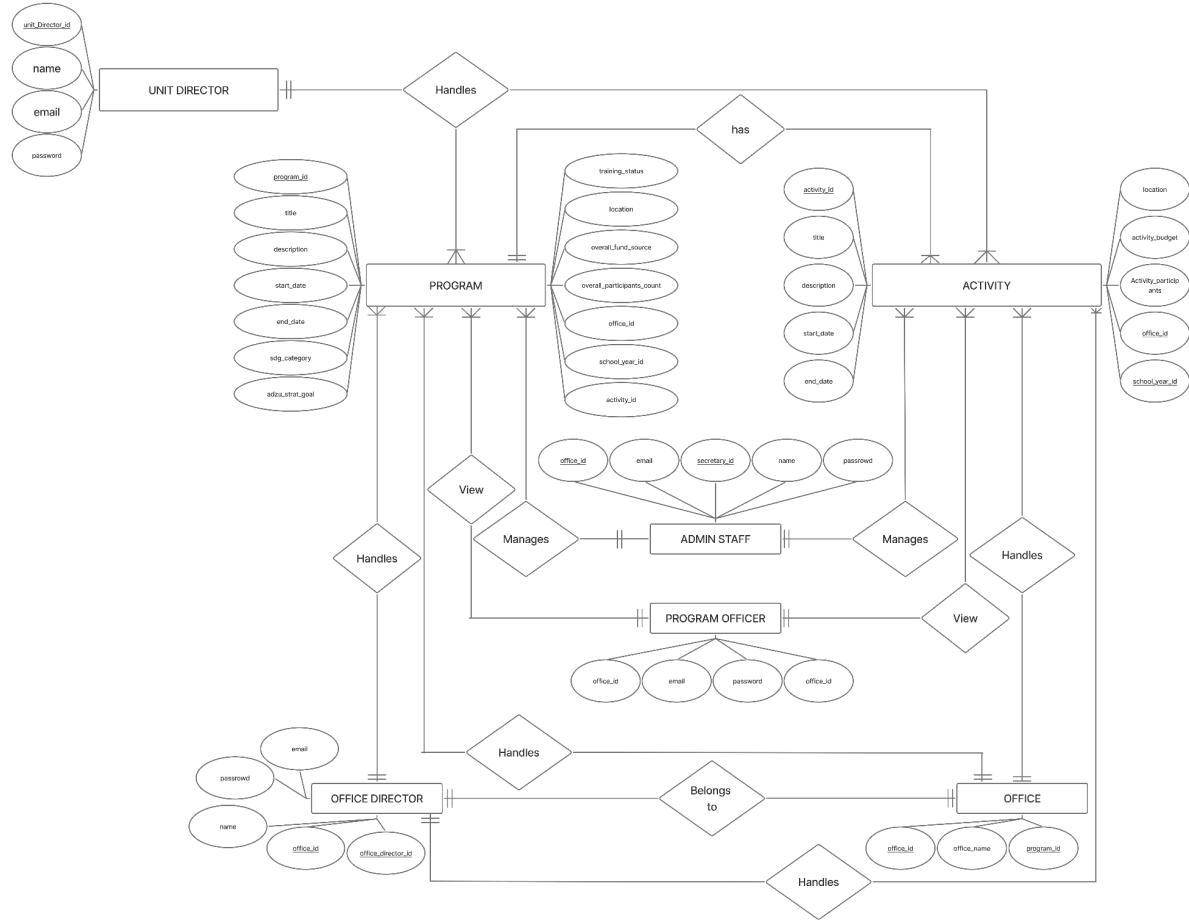


Figure 6. Entity-Relationship Diagram of SDU Project Monitoring System.

The SDU Project Monitoring System's ERD mainly focuses on the **ACTIVITY** entity, thus making it the center for handling the organization's programs and events. The **ACTIVITY** entity keeps the activity name, schedule, objectives, partners, and other needed details. It may look like we are focusing on the **PROGRAM** entity; however, the **PROGRAM** is only a general grouping. The real information and work happen in the **ACTIVITY** itself.

Moreover, the **PROGRAM** just serves as the bigger category, while the **ACTIVITY** shows what is actually going on inside the system. Thus, the **ACTIVITY** becomes the main point in the ERD,

and the other entities connect to it to show how activities are managed, tracked, and monitored in the organization.

1. ACTIVITY OF A PROGRAM: PROGRAM – ACTIVITY

- Each Program includes at least one Activity.
- Every Program that exists contains at least one single Activity.
- On the other hand, every Activity is associated with at least one Program.
- Each Activity that exists is categorized by the Program it belongs to.

2. PROGRAMS OF AN OFFICE: OFFICE – PROGRAM

Every Office manages at least one Program, containing all the Activities an Office implements or has implemented. Relatively, a Program is managed by the Office that it belongs to.

3. OFFICE DIRECTOR OF AN OFFICE: OFFICE DIRECTOR – OFFICE

Each Office assigns exactly one Director – a person or personnel, a representative to use the system.

4. MESSAGES TO OFFICE DIRECTOR: OFFICE DIRECTOR – MESSAGES

The Office Director of each Office only communicates with the Unit Director. They may send and receive many messages, but only from the Unit Director.

5. MESSAGE TO UNIT DIRECTOR: UNIT DIRECTOR – MESSAGES

Unlike Office Directors, the Unit Director can communicate with all Offices. The Unit Director may send and receive messages from every Office under the SDU.

6. OFFICE ROLE: OFFICE DIRECTOR – PROGRAM & UNIT DIRECTOR – PROGRAM

Programs of the other offices are beyond the concern of one office, thus allowing them to be only overseen by the Unit Director and to only oversee their implemented programs or programs implemented under their office. However, the Unit Director as the Admin of all these offices may oversee all the programs under any offices.

7. NOTIFICATION: ACTIVITY – NOTIFICATION

One Activity can generate many Notifications. This means that for each activity implemented, the system can create multiple notifications. Every Notification, however, is related to exactly one Activity, which shows that each notification has a specific origin and purpose. In other words, notifications are tied to the activity they are created for, so it is clear what event or task they are referring to.

8. ACTIVITY TO UNIT DIRECTOR: UNIT DIRECTOR – ACTIVITY

The Unit Director, serving as the admin, will receive many Notifications. These Notifications come from all the Activities under every Office. Each of these Notifications, however, is related to exactly one Activity. This means that even though the Unit Director can get many notifications, each notification clearly shows which specific Activity it is about.

II. Business Rules

UNIT DIRECTOR–PROGRAM

- **B1:** The Unit Director oversees many Programs.
- **B2:** Each Program is overseen by one Unit Director.

UNIT DIRECTOR–MESSAGE

- **B3:** The Unit Director may send or receive many messages
- **B4:** Each Message is sent or received by one Unit Director.

UNIT DIRECTOR–NOTIFICATION

- **B5:** The Unit Director may receive many Notifications.
- **B6:** Each Notification is related to exactly one Activity.

OFFICE DIRECTOR–OFFICE

- **B7:** The Office Director is assigned to exactly one office.
- **B8:** Each Office has one Office Director assigned to.

OFFICE DIRECTOR–MESSAGE

- **B9:** The Office Director may send and receive many Messages.
- **B10:** Each Message is sent or received by one Office Director.

OFFICE–PROGRAM

- **B11:** An Office manages at least one Program.
- **B12:** Each Program is managed by one Office.

PROGRAM–ACTIVITY

- **B13:** A Program contains at least one Activity.
- **B14:** Each Activity must belong to a Program.

ACTIVITY–NOTIFICATION

- **B15:** One Activity can generate many Notifications.

- **B16:** Each Notification is related to exactly one Activity.

C. Query Statements

I. System Functions

User Access & Roles

- The system supports two main user roles: Unit Director and Office Director.
- Only the Office Director from each of the six designated offices can access and encode data specific to their office.
- Login credentials are role-based and strictly tied to the user's assigned office. (Unit Director and Office Director).
- Office Director – has the responsibility to encode and fill out the forms in the website (similar to Google Forms) corresponding to the required table columns.
- Users can only view and manage their own office data; cross-office access is prohibited.
- The SDU Office serves as the Unit Director, with full access to view, manage, and oversee all office data and submissions.

Password and Security Rules

- Users may change their passwords anytime through the system settings.
- Users may change their username anytime through the system settings.
- The system retains the last 7 password histories to prevent password reuse.
- Password policies must follow minimum complexity standards (e.g., length, character types) to enhance security.
- The system enforces strict role-based authentication to protect sensitive office data.

Program and Activity Management

- Each office has a set of program categories aligned with SDU's standards.
- Office Directors can propose new programs, which must be approved by the Unit Director before being added.
- The Unit Director may add programs directly without approval.

- The system includes a fixed set of 17 Sustainable Development Goals (SDGs) selectable per activity.
- Within each program category, offices can add dynamic, user-defined activities, which may vary monthly.
- All activity submissions are encoded through standardized online forms.
- Required data fields for activities include:
 - **Project Name** – official title of the activity.
 - **Program / Component (Dropdown)** – category where the activity belongs.
 - **Description** – short overview of the activity.
 - **Sex Count (Number)** – number of male and female participants.
 - **Venue** – full location of the activity (street, barangay, city, province).
 - **Dates** – inception date and implementation date of the activity.
 - **Sustainable Development Goal(s) (Multi-select)** – SDG(s) addressed by the activity.
 - **AdZU Strategic Goal(s)(Multi-select)** – AdZU Strategic Goal(s) addressed by the activity.
- Activity details must follow the standardized reporting table format to ensure consistency.
- The system must generate a consolidated monthly report every first day of the month, accessible by the SDU Unit Director..
- Offices must be able to encode both present activities and future planned activities for better tracking.

Dashboard Features

- The Unit Director's dashboard provides:
 - **Projects' Percentage** – total percentage for all completed, in progress, and pending projects across all offices.
 - **Upcoming Activities** – scheduled upcoming activities from all offices.

- **SDG Coverage** – displays how many projects have been under the certain SDG from each office.
 - **Activity Status** – number of completed, in progress, and pending projects from all offices.

- Each office dashboard provides:
 - **Activity Status** – number of completed, in progress, and pending projects that the office currently have.
 - **Upcoming Activities** – scheduled upcoming activities from their own office.
 - **Activity Phases** – number of activities that are in each phase (planning, implementation, in progress, completed, and cancelled).
- Office dashboards must include customized login page backgrounds that reflect their identity (logos, photos, or activities).

Reporting & Documentation

- The system supports PDF report generation for official documentation and printing.
- The Unit Director can view consolidated reports from all offices.
- Office Directors can print individual program or activity reports.

II. SQL Syntax

This section presents the SQL syntax used in the SDU Project Monitoring System, including sample queries, screenshots of actual backend code, and explanations of how each query interacts with the PostgreSQL/Supabase database. All SQL operations, SELECT, INSERT, UPDATE, DELETE, filtering, and ordering—are demonstrated in this section.

1. Retrieving All Records from a Table

```
```javascript
const { data: offices, error } = await supabase
 .from("offices")
 .select("id, name, acronym, description")
 .order("name");
```

```

Figure 7.1. Screenshot of Syntax for Retrieving All Records from a Table.

This code generates and executes the following SQL query: ``sql
SELECT id, name, acronym, description FROM offices ORDER BY name; ``
The ` `.from()` method specifies the table, ` `.select()` defines which columns to retrieve, and ` `.order()` sorts the results. The query returns all offices sorted alphabetically by name.

2. Filtering Records with WHERE Clause

```
```javascript
const { data: programs, error } = await supabase
 .from("programs")
 .select("id, title, description, is_fixed, office_id, created_at,
updated_at")
 .eq("office_id", officeId)
 .order("created_at", { ascending: false });
```

```

Figure 7.2. Screenshot of Syntax for Filtering Records with WHERE Clause.

The ` `.eq()` method adds a WHERE clause to filter records. This retrieves all programs belonging to a specific office, sorted by creation date with newest first.

3. Filtering with Multiple Conditions

```
```javascript
const { data: activities, error } = await supabase
 .from("activities")
 .select("id, program_id, status, phase")
 .in("program_id", programIds);
```

```

Figure 7.3. Screenshot of Syntax for Filtering with Multiple Conditions.

The ` `.in()` method creates a WHERE IN clause to filter records matching any value in an array. This retrieves all activities belonging to multiple programs.

4. Nested Joins with Multiple Levels

```
```javascript
const { data: activities, error } = await supabase.from("activities")
 .select(`

 id,
 sdg_number,
 programs(
 id,
 offices(id, name, acronym)
)
 `);
```

```

Figure 7.4. Screenshot of Syntax for Nested Joins with Multiple Levels.

This creates multiple JOIN operations to retrieve data from three related tables in a single query.

5. JOIN Operation

```
```javascript
const { data: upcomingData, error } = await supabase
 .from("activities")
 .select(
 `

 id,
 activity_name,
 start_date,
 end_date,
 status,
 program_id,
 programs(
 id,
 title
)
 `

)
 .in("program_id", programIds)
 .gte("end_date", currentDate)
 .order("end_date", { ascending: true });
```

```

Figure 7.5. Screenshot of Syntax for JOIN Operation.

The nested `programs()` syntax creates an INNER JOIN between activities and programs tables. The ` `.gte()`` method adds a greater-than-or-equal-to comparison for filtering dates.

6. Insert Operations/Creating New Records

```
```javascript
const { data, error } = await supabase
 .from("activities")
 .insert({
 activity_name: activityData.activity_name,
 program_id: activityData.program_id,
 start_date: activityData.start_date,
 end_date: activityData.end_date,
 objectives: activityData.objectives,
 venue: activityData.venue,
 participants_count: activityData.participants_count,
 budget_amount: activityData.budget_amount,
 sdg_number: activityData.sdg_number,
 strategic_goals: activityData.strategic_goals,
 status: activityData.status,
 })
 .select();
```

```

Figure 7.6. Screenshot of Syntax for Insert Operations/Creating New Records.

The `insert()` method creates a new record in the database. The `select()` at the end returns the newly created record, similar to SQL's RETURNING clause.

7. Updating Single Record

```
```javascript
const { data, error } = await supabase
 .from("activities")
 .update({
 activity_name: updatedData.activity_name,
 start_date: updatedData.start_date,
 status: updatedData.status,
 })
 .eq("id", activityId)
 .select();
````
```

Figure 7.7. Screenshot of Syntax for Updating Single Record.

The `update()` method modifies existing records. The `eq()` clause specifies which record to update using the WHERE condition.

8. Updating Multiple Records

```
```javascript
const { data, error } = await supabase
 .from("program_requests")
 .update({ status: "approved", reviewed_at: new Date() })
 .in("id", requestIds);
````
```

Generated SQL :

Figure 7.8. Screenshot of Syntax for Updating Multiple Records.

This updates multiple records at once by matching multiple IDs in the WHERE IN clause.

9. Deleting Single Record

```
```javascript
const { error } = await supabase.from("programs").delete().eq("id",
programId);
```

```

Figure 7.9. Screenshot of Syntax for Deleting Single Record.

The `delete()` method removes records from the database. The `eq()` specifies which record to delete.

10. Combining Multiple Filters

```
```javascript
const { data, error } = await supabase
 .from("activities")
 .select("*")
 .eq("status", "completed")
 .gte("start_date", startDate)
 .lte("end_date", endDate);
```

```

Figure 7.10. Screenshot of Syntax for Combining Multiple Filters.

Multiple filter methods chain together with AND logic, narrowing results to match all conditions.

11. Complete Route Query Example

```
```javascript
router.get("/:officeId/dashboard", authenticateToken, async (req, res) => {
 try {
 const { officeId } = req.params;

 // Get office details
 const { data: office, error: officeError } = await supabase
 .from("offices")
 .select("*")
 .eq("id", officeId)
 .single();

 if (officeError) throw officeError;

 // Get programs for this office
 const { data: programs, error: programsError } = await supabase
 .from("programs")
 .select("id, title, description, created_at")
 .eq("office_id", officeId)
 .order("created_at", { ascending: false });

 if (programsError) throw programsError;

 // Get activities for these programs
 const programIds = programs.map((p) => p.id);
 let activities = [];

 if (programIds.length > 0) {
 const { data: activitiesData, error: activitiesError } = await supabase
 .from("activities")
 .select("id, program_id, status, phase")
 .in("program_id", programIds);

 if (activitiesError) throw activitiesError;
 activities = activitiesData || [];
 }

 res.json({ office, programs, activities });
 } catch (error) {
 console.error("Dashboard error:", error);
 res.status(500).json({ error: "Server error" });
 }
});
```

```

Figure 7.11. Screenshot of Syntax for Complete Route Query Example.

This example demonstrates a complete route that executes multiple sequential queries. It retrieves office information, associated programs, and activities, performing error handling at each step

12. Query Error Management

```
```javascript
try {
 const { data: activities, error } = await supabase
 .from("activities")
 .select("*")
 .eq("program_id", programId);

 if (error) throw error;

 res.json({ activities });
} catch (error) {
 console.error("Get activities error:", error);
 res.status(500).json({ error: "Server error" });
}
```

```

Figure 7.12. Screenshot of Syntax for Query Error Management.

Every database query returns both `data` and `error` properties. The backend checks if an error occurred and handles it appropriately. Errors are logged for debugging while generic error messages are sent to clients for security.

The backend routes use the Supabase JavaScript client to communicate with PostgreSQL. Query builder methods like `select()`, `insert()`, `update()`, and `delete()` generate SQL statements automatically. Methods such as `eq()`, `in()`, `gte()`, and `order()` add WHERE clauses, filters, and sorting. Nested selections create JOIN operations between related tables. This approach provides a secure and maintainable way to execute database queries without writing raw SQL, while still leveraging the full power of PostgreSQL's relational database capabilities.

DESIGN

This section translates the system's functional requirements into concrete and implementable designs. It presents the user interface, key screens, and workflows for each user role, and outlines the report and dashboard designs that provide the Unit Director with clear and actionable oversight.

A. User Interface

The interface design is clean, modern, and minimalist. It uses a dominant blue and white color palette to reflect the Ateneo de Zamboanga University's identity, conveying professionalism and trust. The contemporary look utilizes subtle gradients, rounded elements, and white space.

I. System-Wide

A. Log-in Page

The Login Page functions as the system's secure authentication gateway. Its primary role is to validate the user's identity using credentials and establish a secure browsing session. Crucially, a successful log-in determines the user's role (Unit Director or Office Director) and triggers the appropriate authorization profile, directing them to their specialized dashboard.

Figure 8.1.1. Log-in Page.

B. Search Component

The Search component is a universal tool located in the header of the interface. This function allows the user to quickly find any project, activity, or keyword across the entire system, returning instant results. Conversely, the Office User utilizes the Search component to find activities only within their specific office, with results displayed instantly.

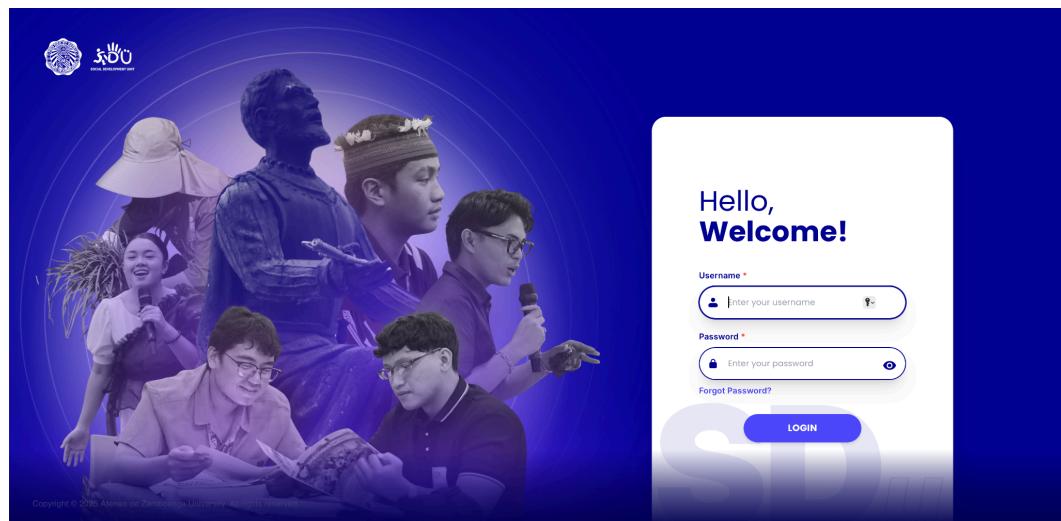
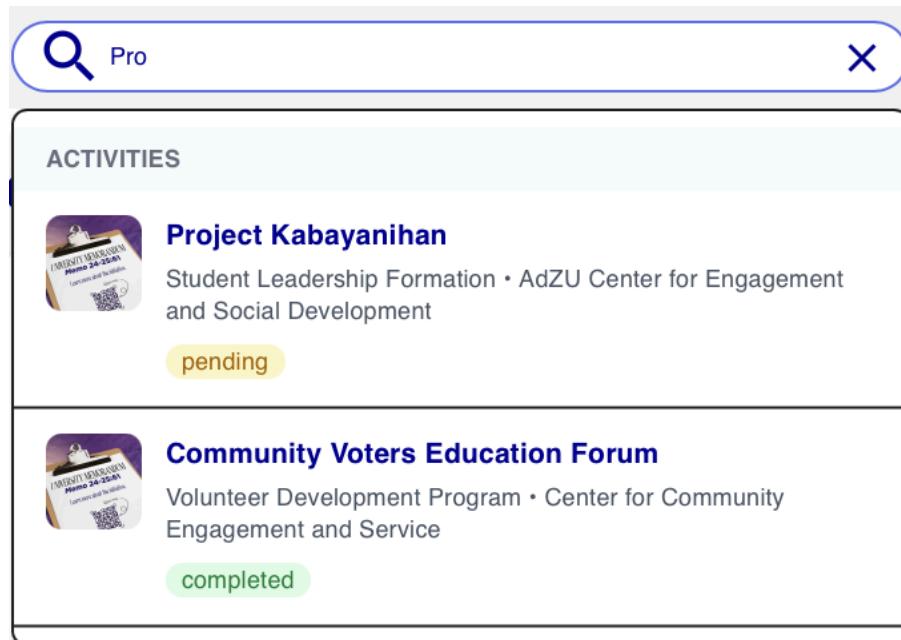


Figure 8.1.2. Search Component.

C. Notification Component

The Notifications component is accessed via a header icon that,



when clicked, reveals a drop-down menu with updates categorized by "All" and "Unread". The notification pop-up clearly displays the Office, Activity Name, the Type of Update (e.g., 'Updated', 'Reminder'), and the corresponding date/time. For the Office User, a specific reminder will display the project's Due date and time

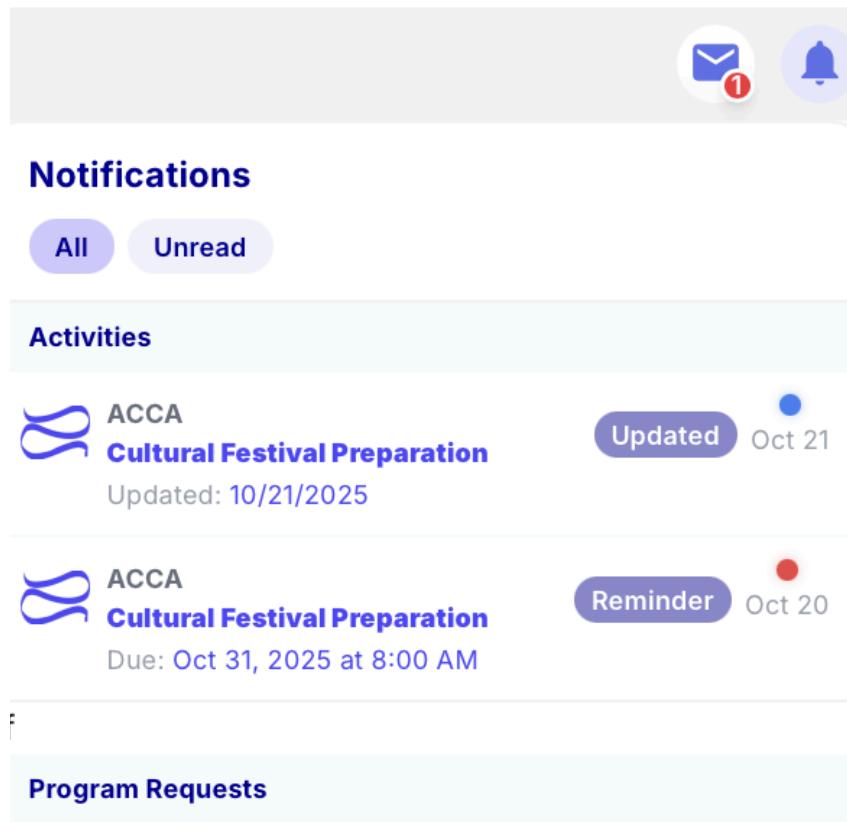


Figure 8.1.3. Notification Component.

- **Unit Director:** Notifications inform the Unit Director of activity updates requiring review, such as a status change (e.g., activity Updated or Submitted by an office) or a program request from a specific office.
- **Office User:** Notifications primarily alert the Office User to updates regarding their own activities, such as an activity being Updated by a colleague, a system Reminder for a due date, or the Approval/Rejection status of a submitted program request.

D. Messaging Component

The Messaging Component is accessed via the Inbox icon in the header and displays messages categorized as "All" or "Unread".

- **Unit Director:** The Unit Director manages the centralized Communication Hub. They can View Messages, Compose messages (to specific offices or broadcast to all), and monitor message metrics and the Automatic Message Deletion Notice.
- **Office User:** The Office User uses the system to send messages directly to the Unit Director for coordination and questions. All inter-office communication is routed through the Unit Director for oversight.

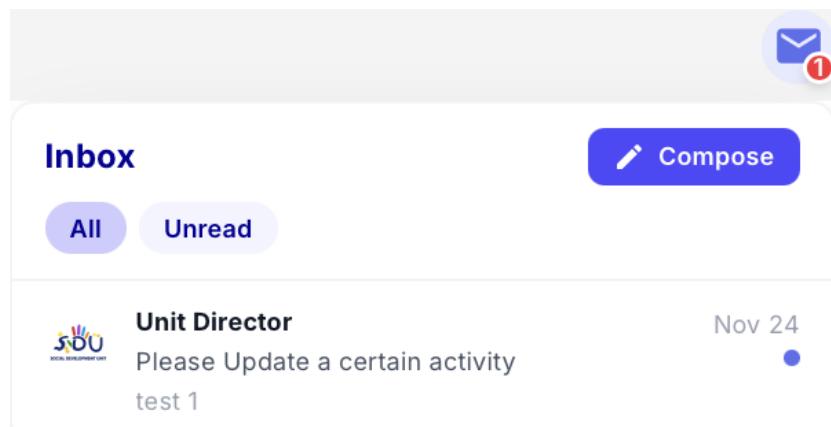


Figure 8.4.1. Messaging Component (Inbox).

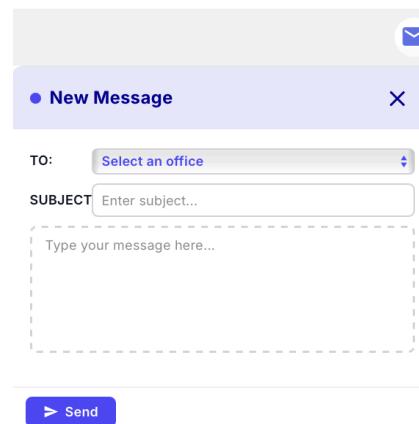


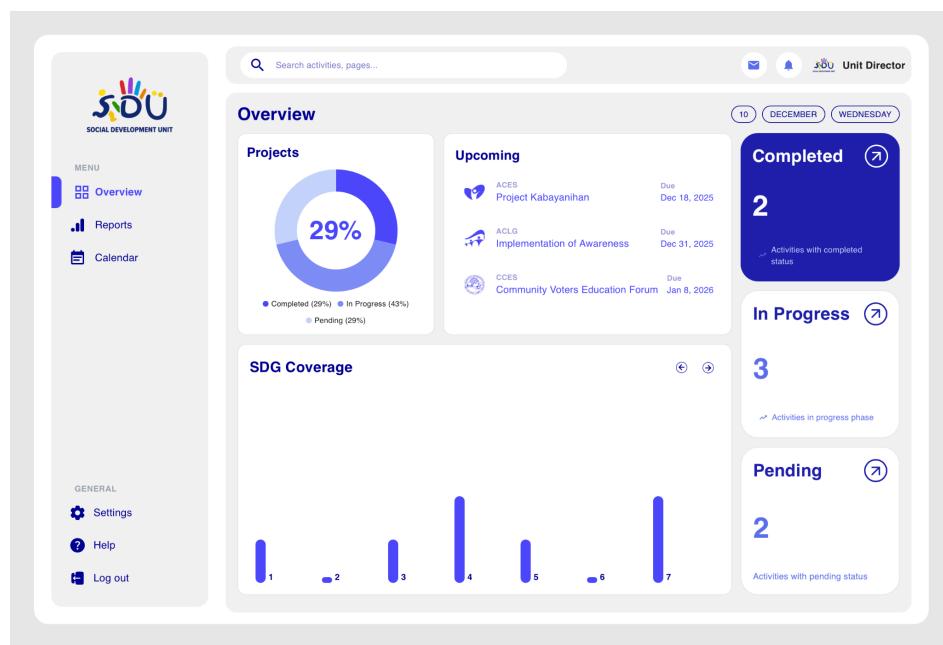
Figure 8.1.4. Messaging Component (New Message).

II. Unit Director

The interface is specifically designed for supervisory monitoring, management of unit-level performance, and administrative oversight of critical system structures.

A. Overview Dashboard

This page provides an executive summary of unit performance,



focusing on key actionable data.

Figure 8.2.1. Unit Director's Overview Dashboard.

| Component | Functionality |
|-------------------------------|--|
| Project Status Donut Chart | Visually represents the percentage distribution of all Offices' projects or activities by their current status: <i>Completed</i> , <i>In Progress</i> , and <i>Pending</i> . |
| Status Metric Tiles | Actionable Navigation: Clicking a status tile filters the Reports tab to display only corresponding activities, aiding in tracking activities per office in that phase. |
| Upcoming Activities List | Highlights the three (3) projects with the nearest deadlines. Clicking any listed activity shows its full report details (figure 1). |
| SDG Alignment Bar Chart | Displays project or activity alignment across the SDG. Hovering over a bar reveals specific project counts or office alignment data per SDG. |

Table 2.1. Functionalities for the Overview Dashboard

B. Reports Dashboard

The Reports Dashboard serves as the central repository for all project and activity documentation submitted by offices under the SDU.

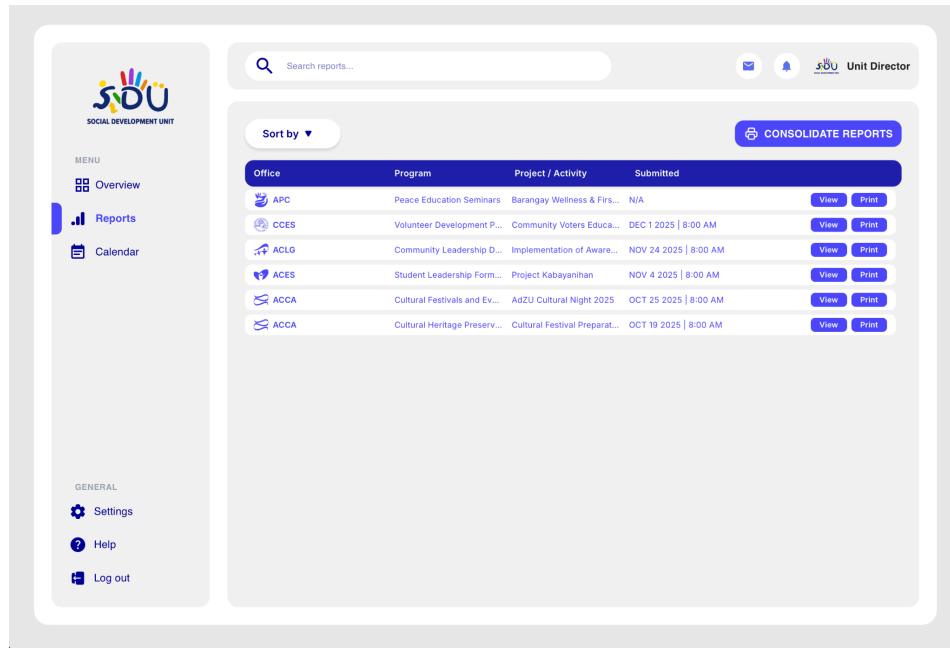


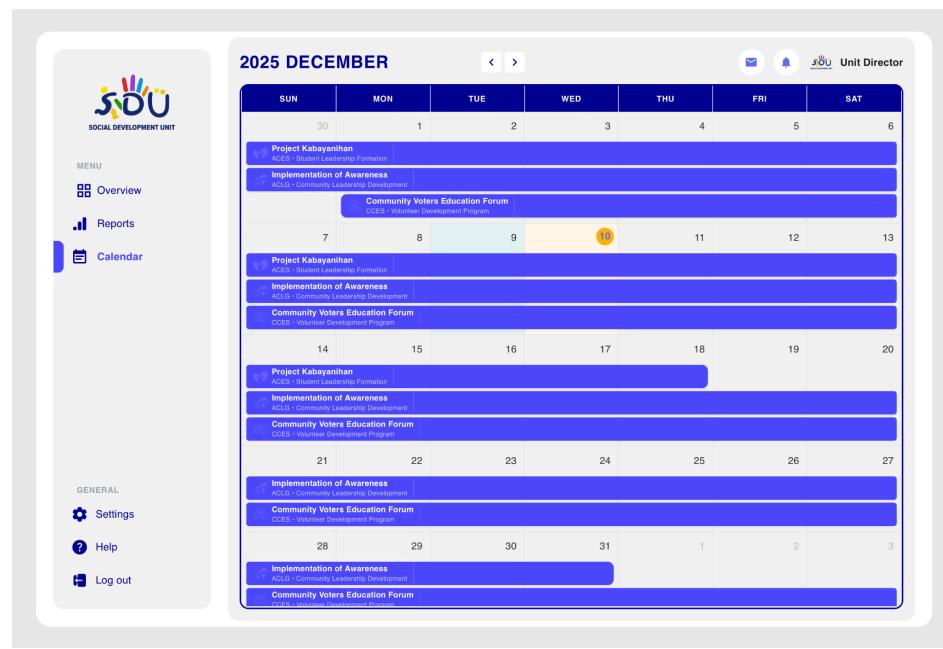
Figure 8.2.2. Unit Director's Reports Dashboard.

| Component | Functionality |
|---------------------------------|--|
| Reports Data Grid | Displays a comprehensive listing of submitted reports, detailing the <i>Office</i> , <i>Program</i> , the Name of the <i>Project/Activity</i> , and <i>Submitted date</i> . |
| Row Action Buttons (View/Print) | Documentation Management: For each listed report, the Unit Director clicks View to open the full detailed report (Figure X) or clicks Print to generate a physical copy (Figure Y). |
| Sort By Dropdown Menu | Data Filtering: Provides flexible options to sort the report list based on criteria such as <i>Recent</i> , <i>Oldest</i> , <i>Office</i> , <i>Program</i> , and by status: <i>Pending</i> , <i>In Progress</i> , <i>Completed</i> , or <i>Cancelled</i> . |
| "Consolidate Reports" Button | Data Aggregation: Initiates the generation of a comprehensive, merged report. Clicking this opens the specialized report generator interface (Figure Z). |

Table 2.2. Functionalities for the Reports Dashboard.

C. Calendar Dashboard

The Calendar Dashboard provides the Unit Director with a dynamic, month-by-month visualization of all scheduled activities and



project deadlines.

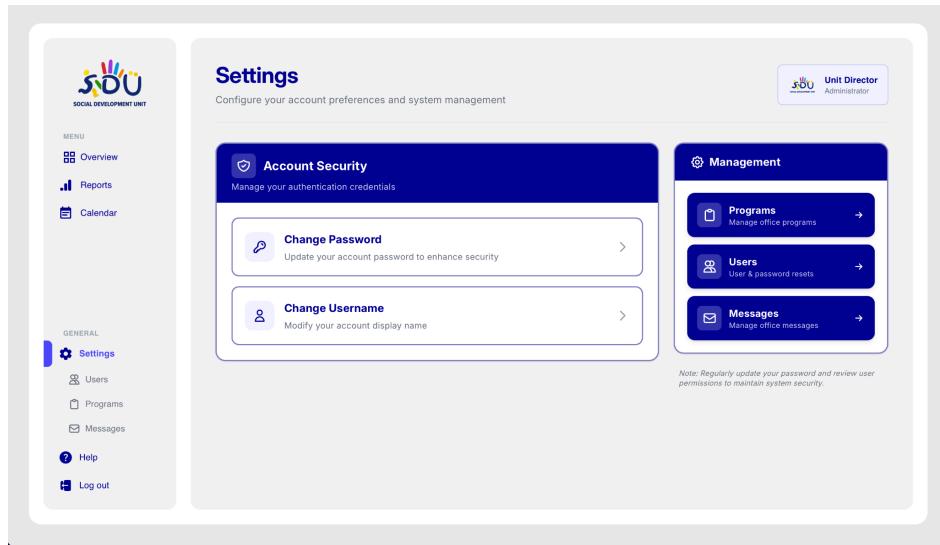
Figure 8.2.3. Unit Director's Calendar Dashboard.

| Component | Functionality |
|----------------------------------|--|
| Project / Activity Visualization | Displays projects and activities as color-coded entries within the monthly calendar grid, showing scheduled dates and duration. |
| Interaction | Clicking an activity entry (e.g., "Project Kabayanihan") immediately redirects the user to the full, detailed report page for that activity. |
| Navigation Controls | Header controls allow for easy navigation to view previous or future months and years. |

Table 2.3. Functionalities for the Calendar Dashboard.

D. Settings Dashboard

The Settings Dashboard serves as the configuration center,

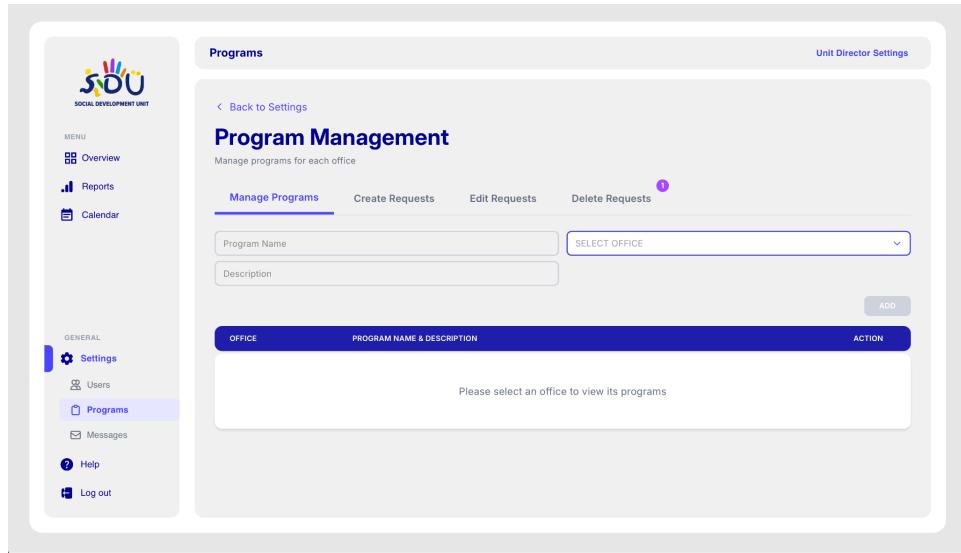


allowing the Unit Director to manage both their personal account security and critical administrative aspects of the Project Monitoring System.

Figure 8.2.4. Unit Director's Settings Dashboard.

| Component | Functionality |
|-------------------------|---|
| Account Security Module | Personal Authentication Management: Contains controls for the Unit Director to update their credentials, including Change Password and Change Username. |
| Management Module | Administrative Oversight Controls: Provides links to Programs, Users, and Messages management sub-dashboards. |

Table 2.4. Functionalities for the Settings Dashboard.



a. Program Management Sub-Dashboard

Figure 8.2.4.1. Programs Management Sub-dashboard.

This sub-dashboard acts as the administrative approval queue for program structural changes. Uses Request Processing Tabs and a Request Data Grid where the Unit Director reviews and performs the Approve or Reject action on pending requests.

The screenshot shows a sub-dashboard titled "User Management". On the left, there's a sidebar with the SDU logo and a menu including Overview, Reports, Calendar, Settings (which is selected), Users, Programs, Messages, Help, and Log out. The main area is titled "User Management" and shows a list of "All Users (6)". It includes columns for OFFICE, ACTIVE STATUS, and LAST PASSWORD UPDATE. The data is as follows:

| OFFICE | ACTIVE STATUS | LAST PASSWORD UPDATE |
|--------|---------------|----------------------|
| APC | Inactive | 10/10/2025, 04:58 AM |
| ACES | Active Now | 10/10/2025, 04:58 AM |
| ACLG | Active Now | 10/10/2025, 04:58 AM |
| CCES | Inactive | 10/10/2025, 04:58 AM |
| ALTEC | Inactive | 10/10/2025, 04:58 AM |
| ACCA | Inactive | 10/10/2025, 04:58 AM |

b. User Management Sub-Dashboard

This sub-dashboard provides oversight of all system users and manages password reset requests initiated by the respective offices.

Figure 8.2.4.2 User Management Sub-Dashboard.

c. *Messages Management Sub-dashboard*

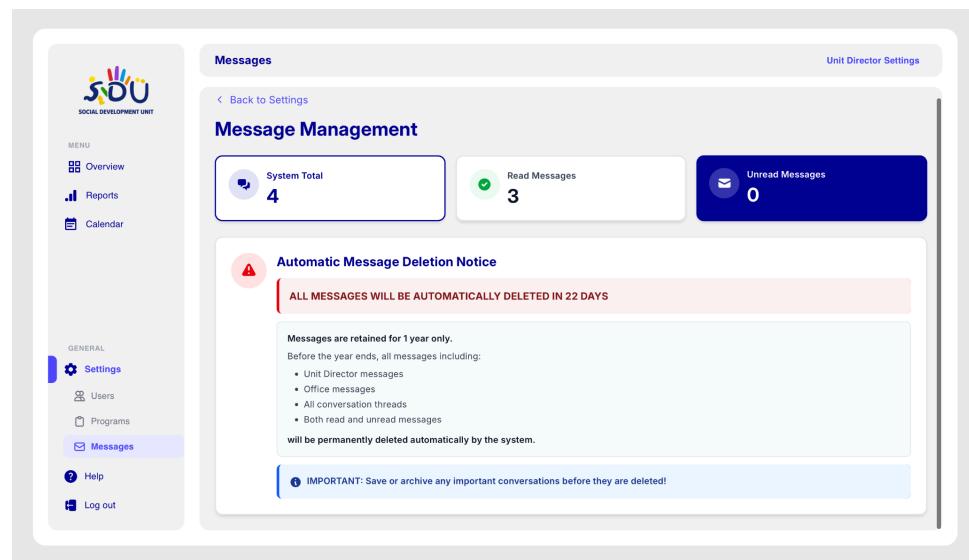


Figure 8.2.4.3. Messages Management Sub-Dashboard.

It serves as the system communication oversight center and data retention alert point. Displays Message Metrics Cards and prominently alerts the Unit Director about the Automatic Message Deletion Notice and the data retention policy.

E. Help & Documentation Dashboard

This serves as the Unit Director's complete guide to the system. administrative confirmation of its capabilities and quick navigation to support resources.

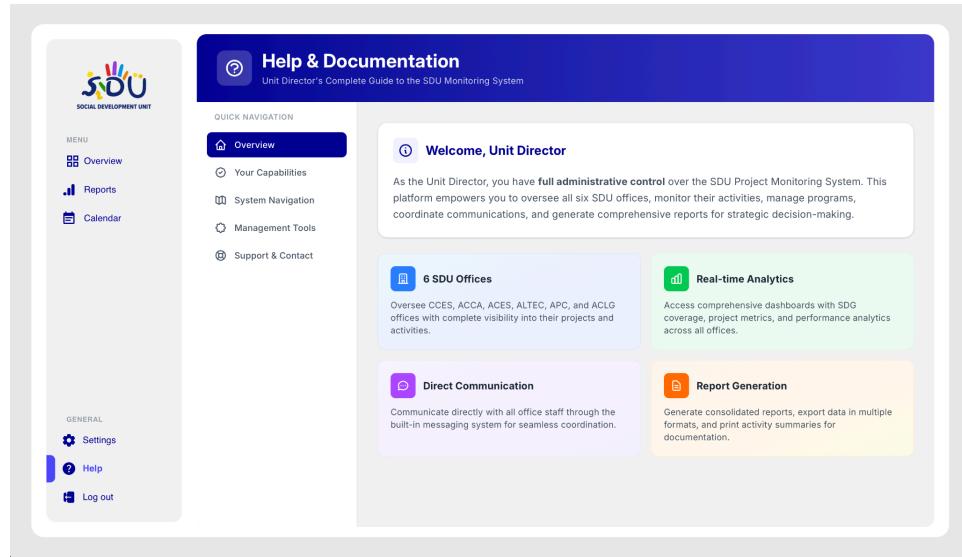


Figure 8.2.5. Help & Documentation Dashboard.

| Component | Functionality |
|-------------------|---|
| Welcome/Overview | Confirms the Unit Director has full administrative control over the six SDU offices and is empowered to manage programs, coordinate communications, and generate comprehensive reports. |
| Your Capabilities | Details administrative privileges: Full Project Oversight, Program Management(Create/Approve/Edit/Remove programs), User Administration (Manage accounts/Reset passwords), Communication Hub (Centralized routing), Analytics & Reporting, and Calendar Overview. |

| | |
|-------------------------|--|
| System Navigation Guide | Explains the function of the main menu items: Overview Dashboard, Reports(Browse/Filter/View/Edit/Print), Calendar, and Settings. |
| Management Tools Guide | Provides detailed instructions on how to use the Program Management, User Management, and Message Center functionalities found under the Settings tab. |
| Support & Contact | Provides resources for technical issues, including the Developer Contact, forms to Report a Bug or submit a Feature Request, and Quick Tips for troubleshooting. |
| Component | Functionality Summary |

Table 2.5. Functionalities for the Help & Documentation Dashboard.

III. Offices

The user's interface is designed for execution, localized project management, and reporting within their specific office or program. Their permissions are focused on creating and updating project data and submitting those changes for Unit Director approval.

A. Overview Dashboard

The Office User's Overview Dashboard serves as a localized monitoring center, focusing strictly on the activities and status of their own office.

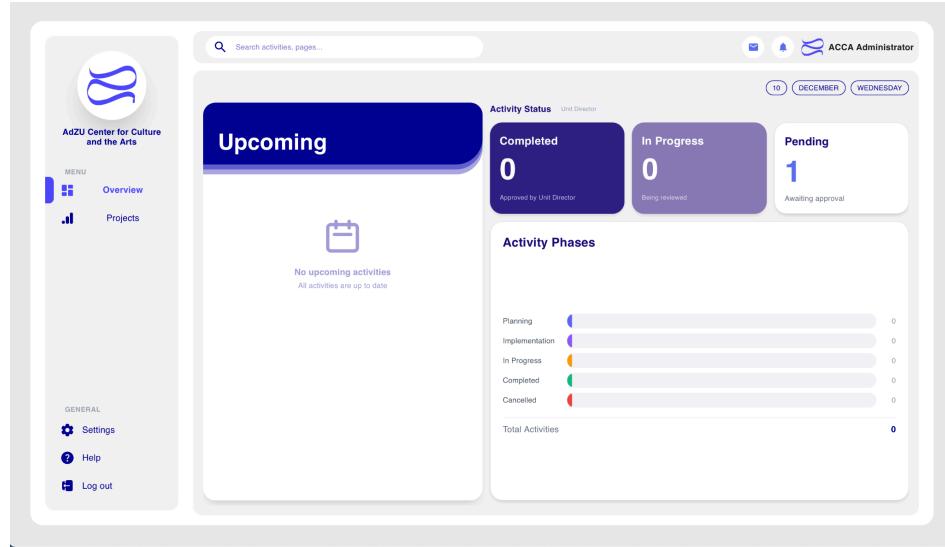


Figure 3.1. Offices Overview Dashboard.

| COMPONENT | FUNCTIONALITY |
|------------------------------|--|
| Activity Status Metric Tiles | Displays key metrics specific to the office's projects, including Completed, In Progress, and Pending. These reflect the current review status (e.g., <i>Approved by Unit Director</i> , <i>Being reviewed</i> , <i>Awaiting approval</i>). |
| Upcoming Activities Section | Alerts the user to projects with near-term deadlines. Clicking on an activity would link to the project editing/reporting interface. |
| Activity Phases Chart | Displays the distribution of all office activities across the project lifecycle stages: Planning, Implementation, In Progress, Completed, and Cancelled. |

Table 2.6. Functionalities for the Offices Overview Dashboard.

B. Projects Dashboard

The Projects Dashboard is the main workspace for creating, managing, and submitting the office's projects or activities.

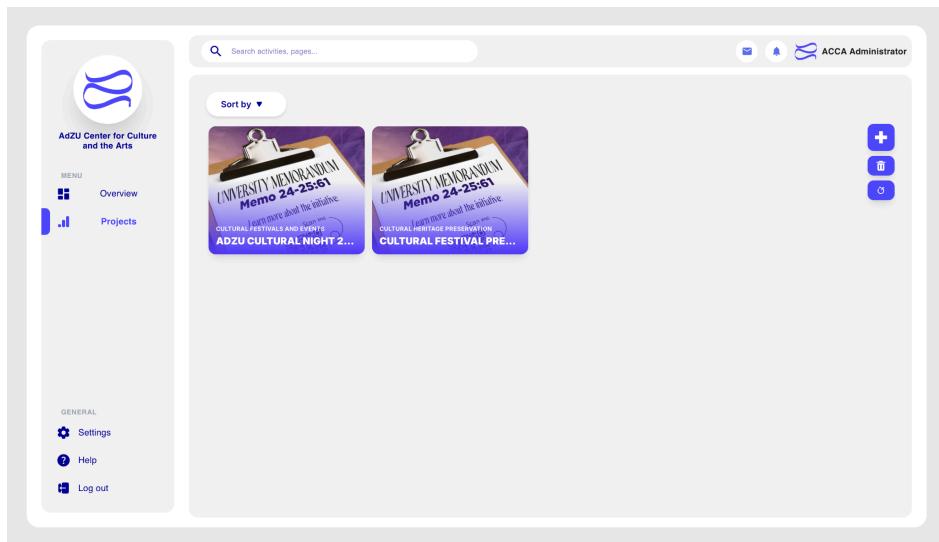


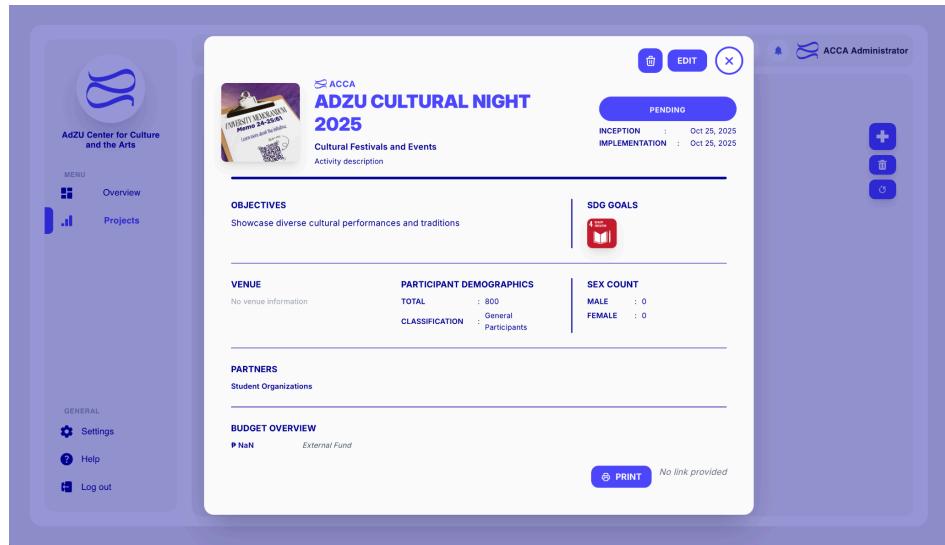
Figure 3.2. Projects Dashboard.

| Component | Functionality |
|-------------------------------------|---|
| Project List Cards | Displays visual cards or a list of all projects/activities owned by the office, showing the Title and Program Name |
| Card Interaction | Clicking on any Project/Activity Card triggers a Modal Window to appear, displaying the full report details and editing interface while keeping the Projects Dashboard visible in the background. |
| Action Buttons (Add/Delete/Refresh) | Provides quick controls on the side panel to Create a New Project (using the plus icon), Delete a selected project, or Refresh the list. |
| "Sort by" Dropdown | Allows the Office User to filter and arrange the list of projects based on specific status criteria: Pending, Completed, and In Progress. |

Table 2.7. Functionalities for the Projects Dashboard.

a. Project Detail Modal View

The Modal Window appears upon clicking any Project



Card. Its purpose is to display all project data, including Objectives, SDG Goals, Dates, and Participant Demographics.

Figure 3.2.1. Project Detail Modal View.

b. Project Form

This is the primary workspace to input all required project data. The form is accessed by clicking either the "Add" button (for new projects) or the "EDIT" Button (for existing projects) on the Projects Dashboard.

Figure 3.2.2. Project Form.

C. Help and Documentation Dashboard

The Help & Documentation Dashboard serves as the complete user guide for the office staff, focusing on project management, performance tracking, and communication within the system.

Figure 3.3. Office Help and Documentation Dashboard.

| Component | Functionality |
|-----------------------|---|
| Welcome/Overview Card | Role Confirmation: Confirms the Office User's primary role to create, manage, and track their office's projects, align work with SDG Goals, and communicate with the Unit Director. |
| Quick | Provides |
| Navigation | instructional sections: Overview, What You Can Do, System |

| | |
|---------------------------|--|
| Menu | Navigation, Managing Activities, and Support & Contact. |
| What You Can Do Guide | Explains the Office User's core system capabilities, including: Activity Management, Program Requests, Dashboard Analytics, Messaging System, Report Generation, and Account Settings. |
| Managing Activities Guide | Provides step-by-step instructions for: Creating a New Activity and Editing an Activity This section also specifies the Required Fields for activity submission. |
| System Navigation Guide | Defines the function of the main menu items: Overview Dashboard, Projects, Settings, and Help. |
| Support & Contact Section | Provides resources for technical issues, including the Developer Contact, forms to Report a Bug or submit a Feature Request, and Quick Tips for troubleshooting (e.g., refreshing the page). |

Table 2.8. Functionalities for the Help and Documentation Dashboard.

B. Report Design

The reporting interface ensures that data viewed within the SDU Project Monitoring System can be formalized into professional, readable documents. The report design adheres to

the clean, modern, and minimalist system aesthetic while utilizing the institutional header for academic integrity.

I. Single Project/Activity Report

|  Ateneo de Zamboanga University <i>The Jesuit University in Western Mindanao, Philippines</i> SOCIAL DEVELOPMENT UNIT | |  | | | | | | | | | | | | | |
|--|----------------------|---|--------|--------------|--|-----------|--|-------|----------------|------|--------|----|----------------------|----|----|
|  COMPLETED | | | | | | | | | | | | | | | |
| Barangay Wellness & First Aid Training <small>A hands-on training program aimed at equipping barangay volunteers with basic first aid skills and emergency response techniques to improve community-level preparedness.</small> | | | | | | | | | | | | | | | |
| OBJECTIVES <small>To train barangay volunteers in first aid, wound care, CPR basics, and emergency response coordination.</small> | | | | | | | | | | | | | | | |
| SDG GOALS <small>SDG 1: No Poverty SDG 4: Quality Education SDG 5: Gender Equality SDG 15: Life on Land SDG 9: Industry, Innovation and Infrastructure</small> | | | | | | | | | | | | | | | |
| VENUE <small>Multipurpose Hall, Barangay Sta. Maria, Sta. Maria, Zamboanga City, Zamboanga del Sur</small> | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">DEMOGRAPHICS</th> <th colspan="2">SEX COUNT</th> </tr> <tr> <th>TOTAL</th> <th>CLASSIFICATION</th> <th>MALE</th> <th>FEMALE</th> </tr> </thead> <tbody> <tr> <td>30</td> <td>General Participants</td> <td>25</td> <td>25</td> </tr> </tbody> </table> | | | | DEMOGRAPHICS | | SEX COUNT | | TOTAL | CLASSIFICATION | MALE | FEMALE | 30 | General Participants | 25 | 25 |
| DEMOGRAPHICS | | SEX COUNT | | | | | | | | | | | | | |
| TOTAL | CLASSIFICATION | MALE | FEMALE | | | | | | | | | | | | |
| 30 | General Participants | 25 | 25 | | | | | | | | | | | | |
| PARTNERS <small>Philippine Red Cross – Zamboanga Chapter Technical Training Provider</small> | | | | | | | | | | | | | | | |
| BUDGET OVERVIEW <small>₱ 6,700.00 Internal Fund</small> | | | | | | | | | | | | | | | |
| Noted By: <hr/> | | | | | | | | | | | | | | | |

*Figure
Project/Activity*

*4.1. Single
Report.*

This report design functions as the official, printable record of a single activity. The header prominently features the Ateneo de Zamboanga University logo and full name, along with the Social Development Unit title. All displayed information—including project details, strategic alignment, logistical data, and participant metrics—is sourced directly from the fields inputted and submitted via the Project Creation/Editing Form. The final line for "Noted By" is included for official sign-off.

II. Consolidated Report

|  Ateneo de Zamboanga University <i>The Jesuit University in Western Mindanao, Philippines</i> SOCIAL DEVELOPMENT UNIT | | | | | |
|---|----------------------|-----------------------------|---------------------|-------------------------------|-------------------------------|
| CONSOLIDATED REPORT | | | | | |
| Month of DECEMBER 2025 | | | | | |
| <p>This consolidated report captures all projects implemented under the Social Development Unit (SDU) for the month of DECEMBER 2025, covering office-specific accomplishments, partner engagements, beneficiaries served, SDG alignment, and financial utilization.</p> | | | | | |
| SUMMARY | | | | | |
| OFFICE | PROJECTS IMPLEMENTED | PARTICIPANTS/ BENEFICIARIES | INT. FUNDS UTILIZED | EXT. FUNDS GENERATED | SDG FOCUS |
| CCES | 1 | 40 | ₱ 0 | ₱ 0 | 03 Good Health and Well-Being |
| TOTAL | 1 | 40 | ₱ 0 | ₱ 0 | - |
| PROJECT STATUSES | | PARTNERSHIPS | | SDG Alignment | |
| Planned/Pending | 0 | Government | 1 | 3 Good Health and Well-Being | 1 |
| Completed | 1 | Non-Government | 0 | 7 Affordable and Clean Energy | 1 |
| In Progress | 0 | Institutional | 0 | | |
| Cancelled | 0 | Internal | 0 | | |
| Total Projects | 1 | Total Partners | 1 | | |
| Noted By: <hr/> | | | | | |

Figure 4.2. Consolidated Report.

This interface manifests as a modal window, triggered when the Unit Director initiates the consolidated reporting function from the Reports Dashboard. To establish the temporal scope of the aggregated data, the Unit Director is required to select the specific Month and Year via mandatory dropdown fields. Subsequent to parameter selection, the Generate Report Button processes the data and outputs the comprehensive, merged report. The resultant document (Consolidated Report for the Month) is formally branded with the Ateneo de Zamboanga University and Social Development Unit (SDU) header, providing a composite overview of all projects and data submitted by the respective offices within the designated monthly period.

TESTING

This section presents the overall evaluation process used to ensure the system's correctness, reliability, and usability. It includes the testing methods applied to validate system functions, the testing results that summarize issues found and resolved, and the training of users, which explains how end-users were prepared to properly operate the system after testing.

A. Testing Method

The system was tested through user testing within the group. Each member of the team acted as a user, logged in to the different kinds of accounts then performed a series of tasks based on the requirements that were provided by our client. The goal during the testing process was to verify that every feature of the system functions was correctly functioning according to the client's expectations, to ensure that the user interface validates all functions such as login, viewing activities, CRUD operations for programs and projects, search functionality, report viewing and printing, messaging, and calendar operations. Each feature was used repeatedly in different scenarios to check if it consistently produced correct results like how the printing format would react when a longer number or text was inputted and such. The group also inspected the system's ability to prevent issues such as incorrect data entry, duplicated records, and missing information by confirming whether form validations, error prompts, and input restrictions were working as it should be.

Aside from the accuracy of the functions, the group also tested the usability of the interface. The group members tested the system on how easy it was to navigate through the pages, whether labels and buttons were clear and readable, and if users could accomplish tasks without any confusion. The readability and clarity of the layout were also observed by the group members to ensure that the system would be intuitive for both the Unit Director and all other Office Directors.

Although full device testing was limited, the team conducted basic responsiveness checks as well by resizing the browser window and testing the system on different screen sizes (e.g. phone, tablet, laptop, and desktop). This helped verify that key interface

elements were able to adjust properly and remain accessible even on smaller or bigger displays.

To measure the expected performance, the team used the following criteria:

- **Functional Accuracy:** All system features should operate as requested by the client.
- **Usability:** Users should be able to understand and navigate the system without difficulty and confusion.
- **Readability:** Text, labels, and interface elements must be clear and easy to understand.
- **Data Integrity:** No incorrect, duplicated, or missing data should occur during use.
- **Responsiveness:** The interface should remain usable on multiple screen sizes (e.g. phone, tablet, laptop, and desktop).
- **Stability:** The system must run without any crashes or errors during normal operations.

Through this user testing approach, the team were able to verify that the system meets the functional and usability requirements set by the client. Minor issues discovered during testing were corrected immediately, ensuring that the final system is stable, reliable, and ready for deployment.

B. Testing Results

Since the testing was conducted internally within the development group, communication during the evaluation phase was clear and efficient. Team members were able to provide immediate feedback as they interacted with the system, allowing issues to be identified and addressed quickly. This made the testing process dynamic, as problems could be demonstrated in real-time and fixes were implemented shortly after they were reported.

The *positive feedback* primarily focused on the system's user interface (UI). Testers consistently noted that the design looked modern, clean, and visually appealing. The layout was praised for being organized and readable, which contributed to a smooth user experience. The consistent color palette, spacing, and overall structure gave the system a professional and updated look, aligning well with what the client expected.

On the other hand, the *negative feedback* centered mostly on functional inconsistencies and certain UX concerns. Some features were not initially working as intended, such as incorrect filtering behavior, missing validations, or unintended button actions. There were also comments regarding the flow of some pages, where users felt that a few actions were not intuitive enough or required additional refinements to make the process smoother. Another comment during our project updates with our client includes how there are some misspelled terms and as well the incorrect use of the office logos that are shown on the sidebar. These issues were systematically addressed by the team, and improvements were implemented based on the feedback.

Overall, the testing results showed that while the system had minor functional and usability issues, most of them were resolved promptly due to the direct and efficient communication within the group. The final refinements helped improve both the functionality and user experience, resulting in a more polished and reliable system.

C. Training of User

To ensure that the SDU Project Monitoring System could be used effectively by the client, several training mechanisms were implemented throughout the development and turnover process:

1. Initial Orientation With the Unit Director and Office Directors

A brief but comprehensive orientation session was conducted with the Unit Director and all Office Directors. During this session, the development team demonstrated the overall system flow, showed how to navigate each page, and explained the main processes such as project encoding, report submission, message viewing, and

calendar tracking.

2. User-Friendly Interface for Easy Learning

The interface was intentionally designed to be clean, simple, and intuitive, allowing users to immediately understand how each feature works without requiring technical knowledge. Buttons, icons, and layouts were organized in a way that mirrors common web applications, making the system naturally easy to learn.

3. Built-in Notes and Descriptions Across the System

To support the users even outside the training session, the system includes short explanatory notes across several pages—especially in the Settings module.

4. Dedicated Help Page With User Guide

A complete Help Page was provided within the system, containing a structured User Guide. This guide serves as a reference for users who need clarification while using the system. The guide includes:

- Purpose: Explains the goal of the SDU Project Monitoring System and its role in improving project coordination.
- Roles and Access: Describes the permissions of Unit Directors and Office Directors.
- Pages and Functions: Covers major modules such as Overview, Reports, Calendar, and Edit Project pages.
- Navigation Controls: Explains how to use the Search tab, Inbox, Notifications, and Account settings.

IMPLEMENTATION

This section details how the system was delivered and made operational for the client. It covers the user support services provided to ensure continued system usability and sustainability, as well as the deployment documentation, which describes the system deployment process,

installation on client devices, and all relevant training and onboarding activities conducted with the client.

A. User Support Services

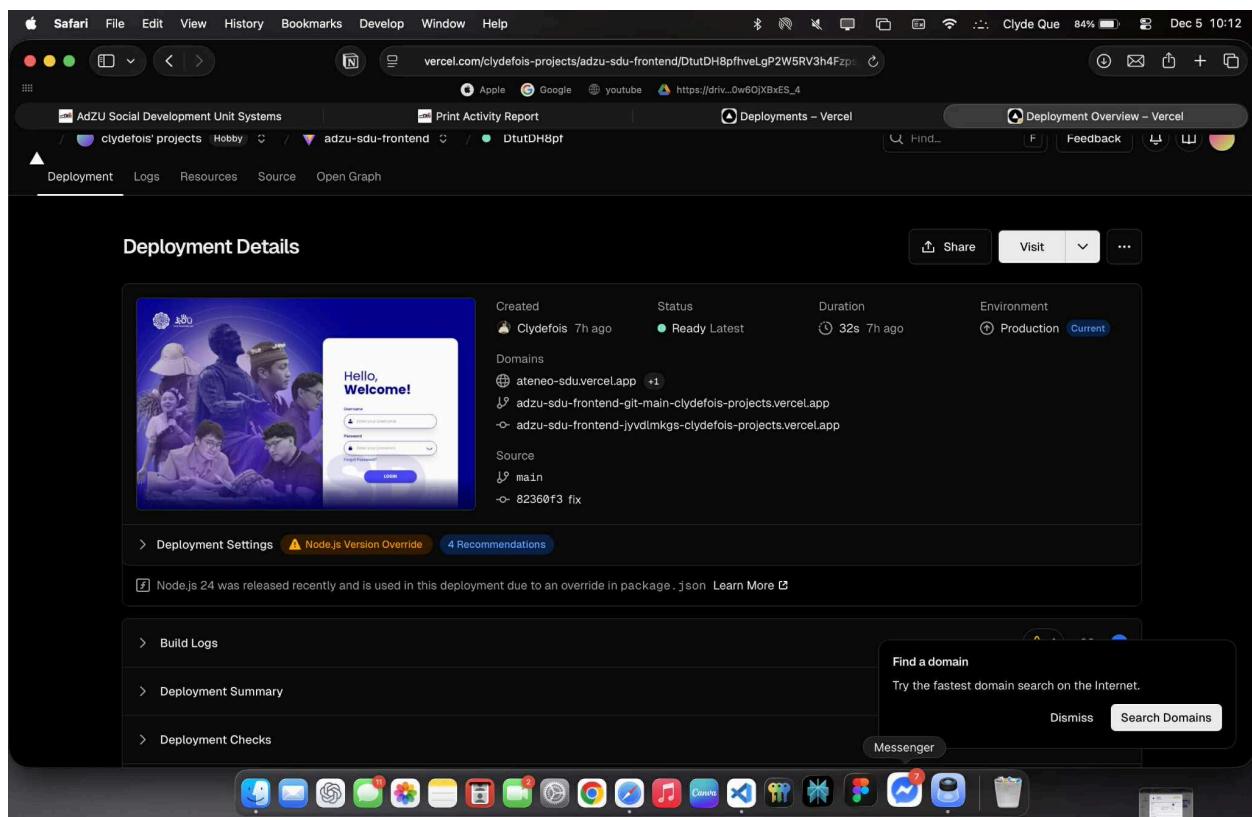
To ensure that the SDU Project Monitoring System remains sustainable and consistently usable by the client, the implementation strategy prioritized platform independence and long-term maintainability. The system is deployed entirely as a cloud-based web application, utilizing Supabase as its Backend-as-a-Service (BaaS) and Vercel for hosting both its frontend and backend components. This architectural choice ensures that the system can operate reliably without requiring the institution to manage its own servers or undergo complex technical provisioning.

To further support long-term usability, modern cloud-hosting practices were applied during deployment.

- **Code Repository:** The source code is maintained in GitHub, enabling version control, developer collaboration, and transparent change tracking.
- **Hosting Platform:** The system, developed using React, is deployed through Vercel. This hosting platform integrates directly with GitHub, enabling continuous deployment (CD), where updates made to the repository are automatically reflected in the live system without requiring manual redeployment.
- **Database Integration:** The Vercel-hosted application is securely connected to Supabase through environment variables containing API keys and database URLs, ensuring encrypted and reliable data exchange between the application and the database.

By utilizing cloud-managed services, automated deployment pipelines, and externally maintained backend infrastructure, the SDU Project Monitoring System ensures operational stability and long-term sustainability. These support devices collectively guarantee that the system will continue functioning smoothly and remain easy for the client to use, maintain, and scale in the future.

For user support, we conducted an initial orientation with the unit director and office heads to ensure they understand how to navigate the system. The help page within the website also contains our contact information, allowing users to directly reach us for bug reporting, feature requests, or other technical concerns. We have also recommended to the client that they appoint our team as the official maintenance group for the system to ensure consistent updates and long-term support.



B. Deployment Documentation

a. Documentation for the System Deployment

The deployment of the SDU Project Monitoring System's database is currently hosted on a free cloud-based service using Supabase, which temporarily serves as the system's provider for the backend database. As for the system's frontend and backend, it is currently deployed through Vercel, which hosts the application and handles the functions used by the system.

Figure 5.1.1. System Deployment.

The screenshot shows the 'Deployments' section of a cloud-based application management interface. At the top, there are navigation tabs: Overview, Deployments (selected), Analytics, Speed Insights, Logs, Observability, Firewall, AI Gateway, Storage, Flags, and Settings. A search bar and a 'Feedback' button are also present. Below the tabs, a message says 'Automatically created for pushes to Clydefois/ADZU-Social-Development-Unit'. The main area displays a table of deployment logs:

| Deployment ID | Environment | Status | Author | Commit Hash | Duration | Timestamp |
|---------------|----------------------|--------|--------|--|----------|---------------------|
| DtutDH8pf | Production (Current) | Ready | main | 82360f3 fix | 32s | 7h ago by Clydefois |
| 6mSHB8sJRF | Preview | Ready | Khan | 64bc5bd Merge remote-tracking branch 'origin/main' | 1m 4s | Nov 29 by Clydefois |
| 8d7VFsQoT | Production (1) | Ready | main | 224f588 fix | 29s | Nov 27 by Clydefois |
| G5dSB7zRa | Production (1) | Ready | main | 4707e93 fix form s | 31s | Nov 27 by Clydefois |
| HKWxpwhkp | Production (1) | Ready | main | 64bc5bd Merge remote-tracking branch 'origin/main' | 32s | Nov 26 by Clydefois |
| 3VNT962rx | Production (1) | Ready | main | 650d010 add programs from offices | 4m 1s | Nov 26 by Clydefois |

Figure 5.1.2. System Deployment.

The screenshot shows a database schema visualization tool. On the left, a sidebar lists categories: Database Management (Tables, Functions, Triggers, Enumerated Types, Extensions, Indexes, Publications), Configuration (Roles, Policies, Settings), and Platform (Replication, Backups, Migrations, Wrappers, Webhooks). The main area displays a schema diagram with tables like activities, notifications, messages, programs, office_credentials, and offices. Relationships between tables are shown as dashed lines. A legend at the bottom right defines symbols: Primary key (key icon), Identity (key icon with arrow), Unique (key icon with circle), Nullable (circle icon), and Non-Nullable (diamond icon).

Figure 5.1.3. System Deployment.

b. Documentation for the Installation Process in Client Devices

The installation of the SDU Project Monitoring System onto client devices is quite different from the usual way software installations are carried out. This system does not require users to download and install software packages on their own personal computers; instead, it works directly through a web browser, so no local installation is necessary. In such a way, this is a zero-installation paradigm: end users can immediately access this system without downloading executable files, running any Installation Wizard, or even configuring anything in the software on their devices. The only thing a user has to do to access the system is open their preferred web browser, be it Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge. They need to enter an address of this system into the address bar. Upon that, the browser will download the relevant pieces of the interface from the cloud server automatically and will render them as a fully functional application. At the same time, these components will be cached within a browser to further speed up subsequent visits. However, they do not get installed permanently on the device of the user. This means that users do not need to concern themselves with storage space or software conflicts and compatibility issues that are common with many traditional software installations. From a technical point of view, the requirements toward this system are minimal and boil down to the necessity for the client device to have a modern web browser with JavaScript turned on, which, by default, is already the case in practically every contemporary browser.

The system is responsive, meaning it automatically adjusts its display and functionality to suit different device types, from desktop computers to laptops, tablets, and smartphones. Even though the system can be accessed from any device, desktop or laptop computers are preferred when extended data entry tasks need to be performed, as their screen size is larger and input methods are more comfortable. Additionally, this system requires access to an active internet connection since all the data processing and storage are performed on the cloud server and not on the local device of the user.

From a practical standpoint, this "installation" process consists of no more than the user accessing the system URL at ateneo-sdu.vercel.app for the first time. During this initial access, it may take a few seconds for the browser to download and cache the interface components, which is an automated process. After the login page appears, users can authenticate with credentials provided by the Unit Director, and upon successful login, immediate access to their personalized dashboard will be granted. There are no software licenses to activate, no configuration files to modify, and no system requirements beyond using a modern web browser and having an internet connection.

During the installation process, we were able to present the system to the Unit Director on how to fully access it. Later on, they also asked us to email the complete domain and complete credentials for all the accounts that the system has.



Figure 5.2.1. Installation Process in Client's Device.



Good day po!

Here is the domain for the SDU Project Monitoring System:

ateneo-sdu.vercel.app

Below are the login credentials po:

Username | Password

unitdirector : password123

acca : password123

aces : password123

aclg : password123

altec : password123

apc : password123

cces : password123

Figure 5.2.2. Email for Domain & Credentials.

c. Documentation for Training and Relevant Activities

The training for the SDU Project Monitoring System took place by client collaboration meetings, project updates and final demonstration presentation. These several meetings achieved a double function of keeping the client informed about how to use and operate the system, and training in an informal/formal way for them to be able to understand the system's concept and functionality. At these update meetings, the development team met the client who would become stakeholders and also had to pitch them the features of functions of the system while showing the updates time-by-time. We were also able to show and answer

questions by the users, giving them a feel for the system before it was placed out there.

We had a total of three (3) meetings with the client that were related to training and other relevant activities. First, we had our initial meeting where the client explained what they wanted for the system. This served as the starting point, as we identified the functionalities that the client needed for the Project Monitoring System. Secondly, we had our first project update with the client, where we asked questions about parts that were still unclear to us. We also presented the initial design through Figma and asked for suggestions regarding the design—such as the colors, typography, and images they wanted to be implemented in the system. We also requested background information about the Social Development Unit, including its programs, the offices under SDU, logos, and photos provided by them. Lastly, we had our final project update and training session with all the Office Directors and the Unit Director, where we presented the currently deployed system and demonstrated how to properly use it—from creating, editing, and deleting projects, and other related actions. We also addressed questions they were confused about, and likewise asked our own questions to clarify certain parts of the system.

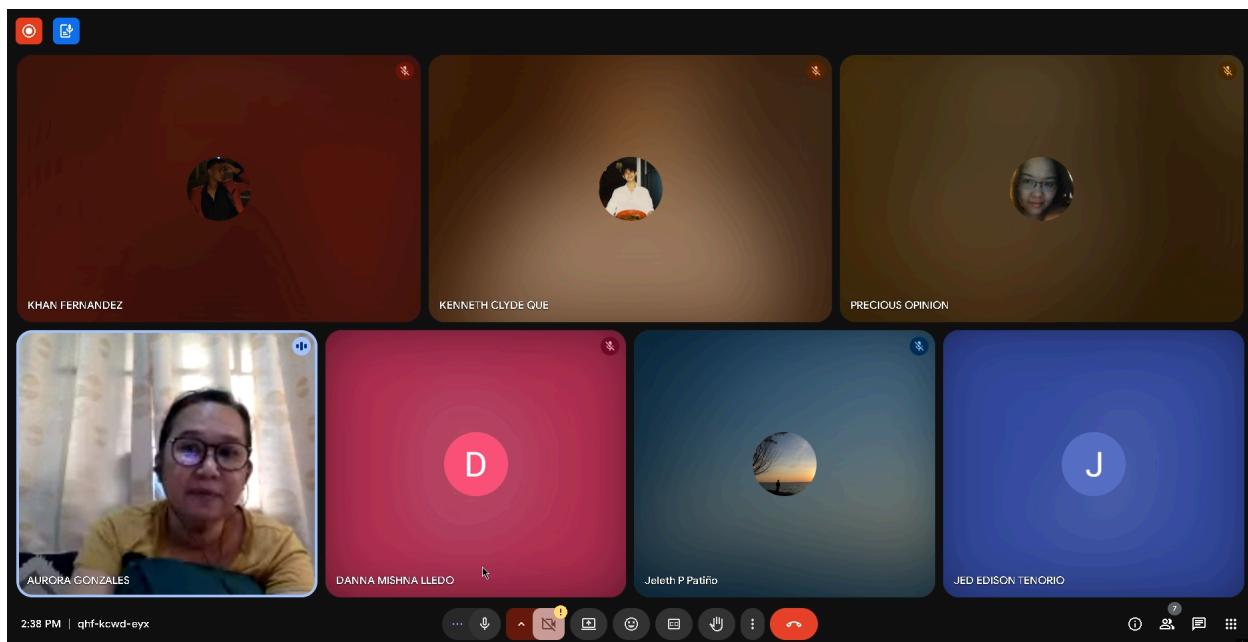


Figure 5.3.1. First Meeting.



Figure 5.3.2. Second Meeting | First Project Update.



Figure 5.3.3. Third Meeting | Second Project Update | Training Session

APPENDICES

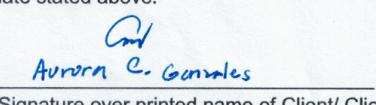
| Ateneo de Zamboanga University | |
|--|--|
| Computer Science Department | |
| Web Database System – Service Learning Program Acceptance Document | |
|  | |
| Field | Detail |
| Client Organization Name | Social Development Unit (SDU) |
| System Name | SDU Project Monitoring System |
| System Version | V1.0.0 |
| Development Team/Vendor | Jerome P. Arawangsa <i>UI/UX Developer</i> Khan O. Fernandez <i>Backend Developer</i> Danna Mishna I. Lledo <i>Frontend Developer</i> Kenneth Clyde A. Que <i>Project Manager & Fullstack Developer</i> Jed Edison I. Tenorio <i>Frontend Developer & QA Analyst</i> |
| Date of Final User Acceptance, Deployment and Turnover | [04/12/2025] |
| Acceptance Declaration | |
| I, the undersigned Client Representative, hereby declare the following: | |
| Full and Unconditional Acceptance | |
| The System/Product named above has been thoroughly tested, meets all agreed-upon requirements and specifications, and is formally accepted by the Client Organization as of the date stated above. | |
|  <u>Aurora C. Gonzales</u> (Signature over printed name of Client/ Client Representative) | |

Figure 6. SDU Project Monitoring System's Acceptance Document

REFERENCES

- Abamonga, E. E. (2023). *Grounded Theory Approach: Its Steps and Procedures in a Scientific Study*. IOER International Multidisciplinary Research Journal.
<https://www.ioer-imrj.com/grounded-theory-approach-its-steps-and-procedures-in-a-scientific-study>
- Aguayo, R. (2023). *Iterative prototyping and stakeholder engagement in system design*. TechInsights.
<https://www.techinsights.com/articles/iterative-prototyping-stakeholder-engagement>
- Akkaya, B. (2023). *Grounded Theory Approaches: A Comprehensive Examination of Systematic Design Data Coding*. International Journal of Contemporary Educational Research, 10(1), 89–103.
<https://files.eric.ed.gov/fulltext/EJ1385933.pdf>
- Alvarez-Sánchez, D., Velázquez-Victorica, K., Mungaray-Moctezuma, A., & López-Guerrero, A. (2023). Administrative Processes Efficiency Measurement in Higher Education Institutions: A Scoping Review. *Education Sciences*, 13(9), 855. <https://www.mdpi.com/2227-7102/13/9/855>
- Cavicchi, C., & Vagnoni, E. (2022). Digital information systems in support of accountability: The case of a welfare provision non-governmental organisation. *The British Accounting Review*, 55(5), 101112. <https://doi.org/10.1016/j.bar.2022.101112>
- fundsforgos. (2024, June 30). End of the project: Evaluation and reporting for youth development projects - fundsforNGOs - Grants and resources for sustainability. *FundsforNGOs - Grants and Resources for Sustainability*.
<https://www.fundsforgos.org/free-resources-for-ngos/project-evaluation-reporting-youth-development-projects/>
- Leduc, A. (2021, May 25). Data warehouse modeling: Is Kimball still relevant? *Medium*.
<https://audreyleduc.medium.com/data-warehouse-modeling-is-kimball-still-relevant-a6b6554b2448>
- Luther, D. (2024, February 2). 17 benefits of ERP software for nonprofits. *Oracle NetSuite*.
<https://www.netsuite.com/portal/resource/articles/erp/nonprofit-erp-benefits.shtml>

- Martins, J. (2025, February 12). Understanding the iterative process, with examples. *Asana*. <https://asana.com/resources/iterative-process>
- Onyango, G. M., & Wairimu, M. A. M. (2024). Monitoring practices and performance of projects by non-governmental organizations in Nairobi City County, Kenya. *SAGE Global Publishers*. <https://sagepublishers.com/index.php/ijssme/article/viewFile/435/466>
- Paapa, C. & Kambona, O. O. (2025). A critical review of Grounded Theory and Thematic Analysis in qualitative research: A way forward for qualitative researchers. *International Journal of Science and Research Archive*, 16(3), 302–313. <https://www.researchgate.net/publication/395414447>
- Rao, P. (2025, March 3). Key challenges faced by NGOs and how to overcome them. *AlmaShines*. <https://www.al mashines.io/challenges-faced-by-ngos>
- Shrestha, P. R. (2025). *Grounded Theory as a Qualitative Research Method*. Journal of Vishwa Adarsha College, 2(1), 114–134. <https://www.nepjol.info/index.php/jovac/article/view/83882>
- United Nations. (2025). *Sustainable Development Goals*. United Nations: UNODC Regional Office for Southeast Asia and the Pacific. <https://www.unodc.org/roseap/en/sustainable-development-goals.html>
- Viteri-Sánchez, C. & Novillo-Villegas, S. (2025). A framework for sustainable adoption of business process management. *Sustainability*, 17(21), 9827. <https://www.mdpi.com/2071-1050/17/21/9827>