

---

# Software Requirements Specification

for

## AngelCare: A Web-Based Inventory Management System with Integrated Medicine Dispenser for Angel's Elderly Home Care Services

Version 1.0

Prepared by

Group Name: KARTON TECH SOLUTIONS

CASTELLTORT, Sevaztiene Izach L.	2022-132568	castelltortizach17@gmail.com
ETOM, Eian L.	2022-130415	etomeian5@gmail.com
LEYSON, Vladimir D.	2022-130127	vleyson12345@gmail.com
OLIVEROS, John Mark	2021-130079	justmarcooo29@gmail.com

**Instructor:** Ms. Mikee Gonzaga-Obing

**Course:** Bachelor of Science in Information  
Technology with specialization in Mobile  
and Internet Technology (BSIT-MI)

**Date:** 10/01/2025

# Contents

## Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>6</b>
1.1	DOCUMENT PURPOSE.....	6
1.2	PRODUCT SCOPE .....	7
1.3	INTENDED AUDIENCE AND DOCUMENT OVERVIEW .....	8
1.4	DEFINITIONS, ACRONYMS AND ABBREVIATIONS .....	10
1.5	DOCUMENT CONVENTIONS.....	10
1.6	REFERENCES AND ACKNOWLEDGMENTS.....	11
<b>2</b>	<b>OVERALL DESCRIPTION .....</b>	<b>13</b>
2.1	PRODUCT OVERVIEW .....	13
2.1.1	<b>Context &amp; Origin .....</b>	<b>13</b>
2.1.2	<b>Product Perspective .....</b>	<b>13</b>
2.1.3	<b>Major Components .....</b>	<b>14</b>
2.1.7	<b>System Boundaries &amp; External Interfaces .....</b>	<b>17</b>
2.2	PRODUCT FUNCTIONALITY .....	17
2.3	DESIGN AND IMPLEMENTATION CONSTRAINTS .....	20
2.4	ASSUMPTIONS AND DEPENDENCIES.....	23
<b>3</b>	<b>SPECIFIC REQUIREMENTS .....</b>	<b>24</b>
3.1	EXTERNAL INTERFACE REQUIREMENTS.....	24
3.1.1	<i>User Interfaces.....</i>	<i>24</i>
3.1.2	<i>Hardware Interfaces.....</i>	<i>32</i>
3.1.3	<i>Software Interfaces .....</i>	<i>36</i>
3.2	FUNCTIONAL REQUIREMENTS.....	37
3.3	USE CASE MODEL.....	39
3.3.1	<i>Use Case #1 .....</i>	<i>40</i>
3.3.2	<i>Use Case #2 .....</i>	<i>41</i>
3.3.3	<i>Use Case #3 .....</i>	<i>1</i>
3.3.4	<i>Use Case #4 .....</i>	<i>1</i>
3.3.5	<i>Use Case #5 .....</i>	<i>2</i>
3.3.6	<i>Use Case #6 .....</i>	<i>3</i>
3.3.7	<i>Use Case #7 .....</i>	<i>3</i>
3.3.8	<i>Use Case #8 .....</i>	<i>4</i>
3.3.9	<i>Use Case #9 .....</i>	<i>5</i>
<b>4</b>	<b>OTHER NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>5</b>
4.1	PERFORMANCE REQUIREMENTS.....	5
4.2	SAFETY AND SECURITY REQUIREMENTS .....	6
4.3	SOFTWARE QUALITY ATTRIBUTES .....	7
<b>5</b>	<b>OTHER REQUIREMENTS .....</b>	<b>9</b>
5.1	DATABASE REQUIREMENTS.....	9
5.2	LEGAL AND COMPLIANCE REQUIREMENTS.....	9
5.3	REUSE OBJECTIVES .....	10
5.1	5.4 INTERNATIONALIZATION REQUIREMENTS .....	10
1.	<i>BACKGROUND .....</i>	<i>12</i>
2.	<i>INTRODUCTION .....</i>	<i>12</i>

3. ASSUMPTIONS .....	12
4. TEST ITEMS .....	12
5. FEATURES TO BE TESTED .....	13
6. FEATURES NOT TO BE TESTED .....	13
7. APPROACH .....	13
8. ITEM PASS/FAIL CRITERIA .....	13
9. SUSPENSION/RESUMPTION CRITERIA .....	14
10. TEST DELIVERABLES .....	14
5.1.1 11. TESTING TASKS .....	14
12. ENVIRONMENTAL NEEDS .....	15
13. RESPONSIBILITIES .....	15
14. STAFFING & TRAINING .....	15
15. SCHEDULE .....	15
5.1.2 16. RESOURCES .....	15
17. RISKS & CONTINGENCIES .....	16
18. APPROVALS .....	16

## Figures

### Table of Figures

Figure 1. Context Diagram .....	17
Figure 2. COMET Diagram .....	21
Figure 3. UML Diagram (Swim Lane Diagram) .....	22
Figure 4. Login Page .....	24
Figure 5. Dashboard Page .....	25
Figure 6. Patients Page .....	26
Figure 7. Patients Profile Page .....	26
Figure 8. Inventory Management Page .....	27
Figure 9. Dispenser Page .....	28
Figure 10. Dispensing Page .....	28
Figure 11. Notification Page .....	29
Figure 12. Reports Page .....	30
Figure 13. User Management Page .....	30
Figure 14. User Account Page .....	31
Figure 15. User Management Page .....	31
Figure 16. Full Hardware Interface .....	33
Figure 17. Arduino Uno .....	33
Figure 18. Arduino Mega .....	34
Figure 19. Stepper Motor .....	34
Figure 20. Servo Motor .....	35
Figure 21. ESP32 WiFi-Module .....	35

**Tables**

Table 1. Software Interfaces .....	36
Table 2. Software Modules .....	37
Table 3. Hardware Modules .....	38
Table 4. Test Case 1 .....	40
Table 5. Test Case 2 .....	41
Table 6. Test Case 3 .....	1
Table 7. Test Case 4 .....	1
Table 8. Test Case 5 .....	2
Table 9. Test Case 6 .....	3
Table 10. Test Case 7 .....	3
Table 11. Test Case 8 .....	4
Table 12. Test Case 9 .....	5
Table 13. Performance Requirements .....	5
Table 14. Safety and Security .....	6
Table 15. Software Quality Attributes .....	7
Table 16. Test Case Table .....	11

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

# 1 Introduction

The increasing demand for efficient healthcare solutions in elderly care facilities has highlighted the importance of accurate medicine management and reliable inventory tracking. Angel's Elderly Home Care Services currently relies on manual processes for monitoring stock levels and dispensing medicines, which often lead to challenges such as human error, delayed administration, and difficulties in record-keeping.

To address these issues, our team has developed **AngelCare: A Web-Based Inventory Management System with Integrated Medicine Dispenser**. This project combines a centralized digital platform with an automated dispensing device, ensuring that medications are administered on schedule while keeping real-time inventory data accurate and accessible. By integrating software and hardware, the system minimizes human error, improves efficiency, and supports caregivers in providing safe and timely healthcare services to residents.

With this context established, the following sections of this document define the purpose, scope, and requirements of the AngelCare system.

## 1.1 Document Purpose

The purpose of this Software Requirements Specification (SRS) is to formally define and document the requirements for the project entitled **AngelCare: A Web-Based Inventory Management System with Integrated Medicine Dispenser for Angel's Elderly Home Care Services**. This document provides a structured foundation that describes what the system must accomplish, how it is expected to behave, and the boundaries within which it will operate. The SRS is intended to serve as a comprehensive reference for all stakeholders involved in the project. For the **client**, it ensures that the agreed objectives and expectations are clearly captured and will be addressed during development. For the **development team**, it provides the detailed specifications needed to design, implement, and test the system effectively. For the **academic panel and evaluators**, it acts as evidence of systematic planning and proper documentation of requirements in line with software engineering standards. Finally, for **future maintainers**, it provides a roadmap that can guide modifications, enhancements, or troubleshooting once the system is deployed.

By outlining the project scope, functional and non-functional requirements, assumptions, and constraints, this SRS reduces ambiguity and establishes a common understanding of the system

to be developed. It also acts as a baseline against which the progress and success of the project can be measured, ensuring that the final output meets the intended goals of providing accurate, efficient, and reliable inventory management within the healthcare setting.

## 1.2 Product Scope

The **AngelCare system** is intended to transform the way medicine inventory and administration are handled in elderly care facilities. Traditionally, caregivers at Angel's Elderly Home Care Services rely on manual methods such as logbooks and verbal reminders to monitor stocks and dispense medications. While these methods are common, they are prone to inconsistencies, human error, and delays, which can compromise resident safety. The proposed system seeks to address these challenges by introducing a **web-based solution** that centralizes, automates, and streamlines medicine management.

The system provides an integrated set of software modules that work together to ensure that medicines are administered accurately and on time. Its ultimate purpose is to **assist caregivers in delivering the right medication, at the right dose, and at the right schedule**, while also supporting administrators in maintaining compliance, accountability, and operational efficiency.

The scope of the system covers the following major functions:

- **Real-time inventory tracking of available medicines** – The system records and updates stock levels immediately whenever medicines are added, dispensed, or marked as expired. This allows administrators and caregivers to monitor current availability without relying on manual counts, reducing the likelihood of shortages or overstocking.
- **Management of residents' medication schedules** – Caregivers can create and update personalized medication plans for each resident, including dosage details, frequency, and timing. This feature ensures that individual treatment regimens are strictly followed and can be modified quickly if medical instructions change.
- **Notifications for low stock, upcoming schedules, and missed doses** – Built-in alerts proactively notify caregivers of critical events, such as when a medicine supply falls below a set threshold, when a dose is due, or when a scheduled intake has been missed. These notifications help prevent delays and enhance patient safety by ensuring timely action.
- **Secure role-based access for administrators and staffs** – To maintain data integrity and security, the system enforces role-based permissions. Administrators have control over inventory, reports, and user accounts, while caregivers are primarily responsible for

scheduling and monitoring residents. This ensures accountability and prevents unauthorized changes to sensitive records

- **Reporting features for audits and monthly consumption reviews** – Administrators can generate comprehensive reports summarizing medicine usage, dispensing history, and stock status. These reports support decision-making for future supply orders, facilitate compliance with internal audits, and enhance transparency for management reviews

By focusing on these modules, the AngelCare system aims to **improve the reliability, efficiency, and accountability of medication management** within the care facility.

The following functions are **explicitly out of scope** in this version of the project:

- Advanced healthcare modules such as patient diagnostics and vital sign monitoring.
- Billing, payment processing, and financial reporting.
- Full-scale electronic medical records (EMR) integration.
- Third-party pharmacy synchronization or external supplier APIs.

These areas may be considered for future enhancements, but are excluded from the current scope to maintain feasibility within the project's timeline and resource constraints.

### 1.3 Intended Audience and Document Overview

This Software Requirements Specification (SRS) is prepared for multiple stakeholders who play essential roles in the development, evaluation, and long-term sustainability of the project **AngelCare: A Web-Based Inventory Management System with Integrated Medicine Dispenser for Angel's Elderly Home Care Services**.

- **Client / Sponsor** – The primary client for this project is **Angel's Elderly Home Care Services**, owned and managed by **Ms. Margelyn Gabbac**. As the system's end-user representative, Ms. Gabbac will oversee the project's alignment with the operational needs of the care facility. The client's input ensures that the system addresses real-world challenges in medicine inventory tracking and elderly care management.
- **Developers (KARTON TECH SOLUTIONS)** – The project is being developed by **KARTON TECH SOLUTIONS**, a student-led team composed of:
  - **John Mark Oliveros** – Project Manager
  - **Sevaztiene Izach L. Castelltort** – Lead Programmer & Documenter
  - **Eian L. Etom** – QA Analyst



- **Vladimir D. Leyson** – UI/UX Designer & Hardware Engineer  
As the development team, KARTON TECH SOLUTIONS is responsible for system design, coding, testing, documentation, and deployment, ensuring the solution fulfills the requirements defined in this SRS.
- **Subject Instructor and Evaluator – Mrs. Mikee Gonzaga-Obing** is the subject instructor for Software Development. She evaluates the project as part of the academic requirements for the course, reviewing its methodology, design, and deliverables to ensure compliance with course outcomes.
- **Faculty Adviser** – The project is guided by **Ms. Kris P. Garcia**, who serves as the faculty adviser. She provides mentorship and guidance in aligning the system with proper software engineering methodologies, ensuring that the project meets both academic and professional standards.
- **Panelists** – The project will also be reviewed by a panel of faculty members during the Capstone Defense. Their role is to critically assess the system's compliance with software engineering practices, its functionality, and its overall contribution to the discipline. Feedback from the panelists will be instrumental in refining and finalizing the system for deployment.
- **Future Maintainers** – This document is also intended for future developers, IT staff, or project successors who may enhance or maintain AngelCare after its initial deployment. The SRS provides them with a reliable reference to understand the system's goals, structure, and constraints.

## Document Organization

This SRS is structured into four main chapters for clarity and completeness:

- **Chapter 1: Introduction** – Provides the context, purpose, scope, and intended audience of the system.
- **Chapter 2: Overall Description** – Describes the product perspective, features, design constraints, assumptions, and dependencies.
- **Chapter 3: Specific Requirements** – Details the functional requirements, external interfaces, and use cases that define system behavior.
- **Chapter 4: Non-Functional Requirements** – Outlines system qualities such as performance, security, usability, and maintainability.

This structure ensures that every stakeholder, from the client to the evaluators, has a clear and detailed understanding of the system's requirements and intended purpose.

## 1.4 Definitions, Acronyms and Abbreviations

To maintain clarity and avoid confusion, this section provides the key terms, abbreviations, and acronyms used throughout the Software Requirements Specification (SRS). These definitions ensure that all stakeholders — including the client, developers, faculty, and evaluators — share a common understanding of the technical and project-specific language used in this document.

- **AngelCare** – The proposed system for Angel’s Elderly Home Care Services.
- **CRUD** – Create, Read, Update, Delete; standard database operations.
- **Admin (Administrator)** – User with full control over system settings, users, and inventory.
- **Caregiver/Nurse** – User responsible for managing resident schedules and medicine dispensing.
- **FR (Functional Requirement)** – Specific task or feature the system must perform.
- **NFR (Non-Functional Requirement)** – System qualities such as usability, reliability, and security.
- **SRS (Software Requirements Specification)** – Document defining the project’s requirements in IEEE format.

## 1.5 Document Conventions

This SRS follows the guidelines of **IEEE Std 830-1998** for Software Requirements Specifications [5]. To maintain consistency and readability, the following conventions are used throughout the document:

- **Requirement Labels** – Functional requirements are labeled as **FR-x** and non-functional requirements as **NFR-x**, where x represents the unique identifier.
- **Numbering** – Sections and subsections are numbered hierarchically (e.g., 1.0, 1.1, 1.1.1) to provide structured navigation.
- **Terminology** – Key terms, acronyms, and abbreviations are defined in Section 1.4 for clarity.
- **Formatting** – Important terms are highlighted in **bold**, while technical terms and acronyms appear in italics when first introduced.
- **References** – In-text citations follow IEEE referencing style, with full details provided in the References section.

## **1.6 References and Acknowledgments**

### **References**

This SRS references the SOFTDEV Paper (2025), academic works in healthcare IT, and official software documentation. Additional scholarly sources and technical manuals have also been consulted to support the requirements and design specifications of this project.

### **Acknowledgments**

The success of this project would not have been possible without the unwavering support, guidance, and cooperation of many individuals and groups. Their contributions, whether big or small, served as an essential part of our journey in developing this capstone project. We, the members of KARTON TECH SOLUTIONS, wish to dedicate this section to express our deepest gratitude to all who have been part of this endeavor.

We sincerely thank our client, Angel's Elderly Home Care Services, for giving us the opportunity to design and develop a system that directly addresses the challenges faced in elderly care. Their openness to collaborate and willingness to provide us with the necessary resources and information enabled us to better understand the needs of the institution and tailor the system accordingly.

Our deepest gratitude is extended to Ms. Margelyn Gabbac, the owner of Angel's Elderly Home Care Services, for her trust and continued support throughout the development process. Her valuable feedback, insights, and encouragement motivated us to stay focused on creating a system that would not only serve the staff but, most importantly, safeguard the welfare of the patients under their care.

We are also thankful to the nurses and caregivers who willingly participated in interviews and discussions, despite their busy schedules. Their first-hand knowledge of the medication process and the struggles faced in manual monitoring gave us practical insights that greatly influenced the design and functionality of AngelCare. Without their contributions, the system would not have been as closely aligned to real-life operations as it is now.

We acknowledge the patients of Angel's Elderly Home Care Services, who served as the heart and inspiration of this project. Their well-being and safety were our top priorities during the

entire development process. Every feature and improvement in AngelCare was designed with the intention of creating a system that would ensure they receive the right medication at the right time, thereby improving their quality of life.

Our heartfelt appreciation goes to Mrs. Mikee Gonzaga-Obing, our subject instructor for Software Development, for her guidance, patience, and encouragement. Her constructive feedback and keen eye for detail helped us meet the academic requirements of this project while also challenging us to uphold high standards in software development. She consistently reminded us of the value of discipline, perseverance, and responsibility throughout this process.

We are equally grateful to Ms. Kris P. Garcia, our project adviser, whose mentorship and professional advice helped us align the project with sound software engineering practices. Her guidance greatly influenced the quality and success of this work.

We also extend our gratitude to our panelists, who carefully evaluated our system during the Capstone Defense. Their constructive criticisms, recommendations, and insights provided us with valuable perspectives that allowed us to improve both the functionality and presentation of AngelCare. Their role in reviewing our work was instrumental in ensuring that this project adhered to academic and professional standards.

Our deepest thanks also go to our families for their unwavering love, patience, and encouragement. Their support gave us the strength to continue even during the most challenging times of this project. We give special recognition to Mrs. Jocelyn D. Leyson and Mr. John Leslie Leyson, the parents of our teammate Vladimir D. Leyson, for their extraordinary generosity in providing us with shelter, food, and every necessity we required during the course of system development. Their kindness and hospitality made it possible for us to work comfortably and productively.

We also wish to thank our friends, who stood by us and offered their support throughout this journey. Their encouragement and positivity gave us comfort during stressful times and reminded us that we were not alone in our struggles. Their simple yet meaningful acts of support pushed us to stay motivated and determined.

We acknowledge our special someones, who served as our inspiration throughout the project. Their belief in us and their encouragement to continue striving for success reminded us of the greater goals we are working toward, not only for this project but also for our future endeavors. Their presence gave us an added sense of purpose and drive to accomplish this milestone.

Finally, above all else, we give our sincerest thanks to God Almighty, whose guidance and blessings have been with us every step of the way. We are grateful for the wisdom, patience, and strength He has provided, enabling us to complete this project successfully. Without His divine intervention, this work would not have been possible.

## 2 Overall Description

### 2.1 Product Overview

#### 2.1.1 Context & Origin

AngelCare is a **new, self-contained** software product developed to replace manual medicine inventory and scheduling processes currently used at Angel's Elderly Home Care Services. It was created in response to operational problems observed during stakeholder interviews (delays, manual counting errors, and difficulty auditing medication events) and therefore is intended as a direct replacement for those manual workflows rather than as a plug-in to an existing enterprise system. The product is designed to be deployable within a single facility (initial deployment).

#### 2.1.2 Product Perspective

From a product-perspective viewpoint, AngelCare acts as the central **digital hub** for inventory, scheduling, and audit logging. Its main responsibilities are:

- Provide a reliable authoritative record of medicine stock and dispensing events.
- Host resident medication schedules and trigger / log dispensing actions.
- Expose administrative functions (reports, user management, thresholds).
- Provide near real-time status for caregivers and administrators via dashboards and alerts.

AngelCare is not an electronic medical record (EMR) and does not replace clinical charting; instead, it focuses on medication logistics.

### 2.1.3 Major Components

The system is composed of the following major components (these correspond to items in your context diagram):

1. **Web Client (Frontend)** — Browser-based UI used by Administrators and Caregivers for inventory management, schedule creation, reporting, and manual overrides. Responsive for tablet/desktop use.
2. **Database (PhyMyAdmin)** — Persistent store for users, residents, medicines, inventory items, schedules, dispense events, alerts, and audit logs.
3. **Medicine Dispenser (Device / Edge Controller)** — The physical dispensing unit that receives scheduled commands and returns confirmations and diagnostic data.
4. **Notification & Activity Log (Internal Web Module)** — Instead of relying on external channels like email or SMS, AngelCare provides a dedicated web page within the system where all dispenser-related activities and notifications are listed in real time. This module records staff actions, automated events, and system alerts, ensuring transparency, accountability, and efficient monitoring for both administrators and caregivers. Admin / Caregiver Devices, Tablets or PCs used on-site to access the Web Client.
5. **Optional External Services** — Supplier APIs for ordering (future), EMR/Health Information Exchange (future), and cloud backup services.

### 2.1.4 Context Diagram Explanation

Figure 2.1 illustrates the operational context of AngelCare, showing how the system supports the entire medication management process, from physician prescription to final administration and incident handling. The diagram highlights the interconnected subsystems and the flow of information between them.

#### Actors and Stakeholders

- **Physician:** Provides prescriptions that serve as the basis for medication schedules.

- **Admission Staff / Nurse:** Responsible for entering patient medication details into AngelCare during admission and reconciling them with physician prescriptions.
- **Caregivers / Nurses:** Use AngelCare to schedule, monitor, and verify dispensing events. They also log any incidents or irregularities in the system.
- **Inventory Officer:** Oversees medicine stock, acts on low-stock or expiry alerts, and manages replenishment workflows.
- **Registered Nurse:** Reviews incidents flagged in the system and validates actions taken by staff.
- **Family Members:** Receive notifications when an incident (missed dose, refusal, or error) is logged.
- **Patients / Residents:** The ultimate recipients of care, whose medication needs drive the entire workflow.

## Primary Workflows and Data Flows

### 1. Prescription and Admission

- The process begins with a physician's prescription, which is reconciled during patient admission. Medication details are then encoded into AngelCare to initialize schedules and inventory tracking.

### 2. Inventory Management

- When medications are entered, the system performs an **inventory check**.
- If stock is insufficient, a **reorder trigger** is generated, leading to supplier purchase orders, receipt of medications, quality control, and storage in inventory.
- If stock is sufficient, the medication schedule proceeds without interruption.

### 3. Scheduling and Dispensing

- Medication schedules are loaded into both AngelCare and the physical dispenser.
- At the scheduled time, a dispense event is triggered.
- The dispenser verifies the action through a sensor or camera confirmation.

- Caregivers then validate and log the dispensing event in AngelCare.

#### 4. Alerts and Follow-up

- AngelCare continuously monitors for **low-stock or expiry conditions**.
- The system generates alerts in the Notifications and Activity Log page, where the **Inventory Officer** acts to resolve the issue, ensuring continuous availability of medicines.

#### 5. Administration and Incident Handling

- AngelCare verifies whether the medication was successfully administered.
- If administered correctly, inventory is decremented automatically.
- If the medication was missed, refused, or an error occurred, the system logs an incident. This triggers a workflow involving registered nurse review and, when appropriate, family notification to maintain transparency and accountability.

### System Boundaries and Integration

- AngelCare is responsible for coordinating all internal workflows shown in the diagram.
- External systems such as physician prescription platforms or supplier systems are outside the current implementation scope, but the design allows for possible integration in future versions.
- The Notifications/Activity Log page serves as the central hub for transparency, replacing external communication methods such as SMS or email.



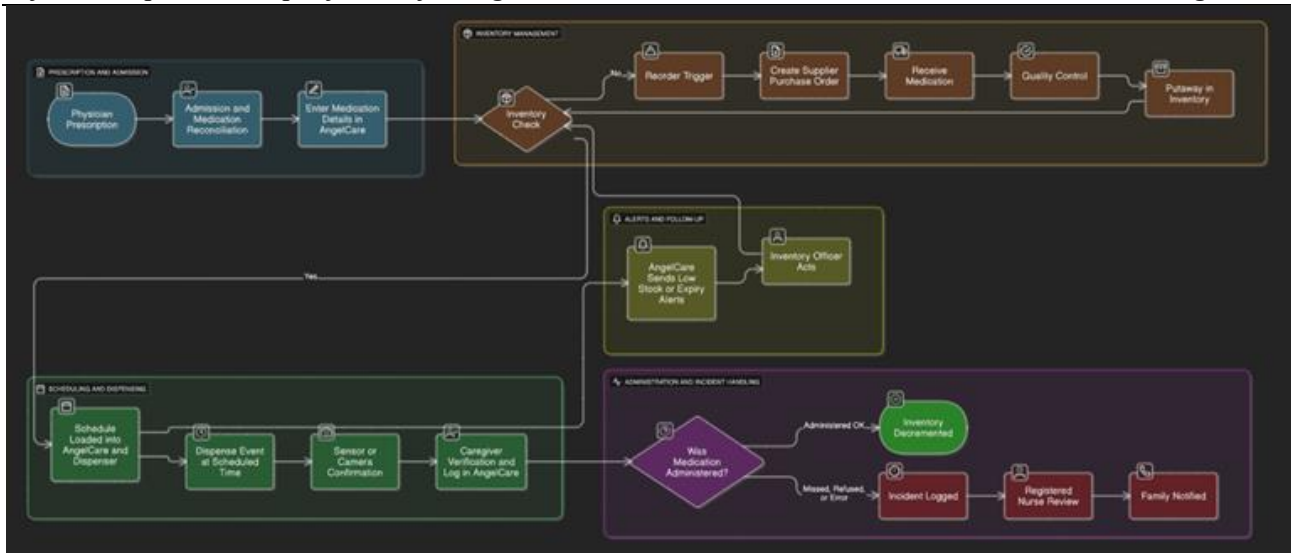


Figure 1. Context Diagram

### 2.1.7 System Boundaries & External Interfaces

**Inside scope:** Web app UI, database, dispenser command/confirmation interface, notifications, reports, audit logs.

**Outside scope (for this SRS):** clinical EMR, billing, automated ordering to external pharmacies (unless future scope), full patient medical records. These external systems are recognized as potential integration points; interface requirements for these systems are documented as future extension considerations.

## 2.2 Product Functionality

The AngelCare system, together with its integrated medicine dispenser, provides a comprehensive set of functions that streamline patient management, medicine inventory, dispensing operations, and reporting within Angel's Elderly Home Care Services. Below is a detailed summary of its major functionalities:

- **User Authentication and Access Control**

- Staff and administrators access the system through the login page ([angelcare.space/login.php](http://angelcare.space/login.php)).
- Upon successful login, users are redirected to the dashboard ([index.php](http://angelcare.space/index.php)).
- Role-based access control ensures that only authorized users (Admin, Caregiver, Inventory Officer) can perform specific actions within the system.

---

- **Dashboard Overview**

- The dashboard serves as the central hub for monitoring system activities and status.
- It displays comprehensive metrics such as total patients, total stock in inventory, upcoming medicine intakes, missed medicine intakes, quick action shortcuts, daily schedules, recent activities, staff performance (last seven days), stock levels, dispenser status, and notifications.
- This provides staff and administrators with a real-time snapshot of both patient care and operational efficiency.

- **Sidebar Navigation**

- A sidebar menu gives users quick access to all core modules, including Patients, Inventory, Dispenser, Dispensing, Notifications, and Reports.
- This ensures efficient navigation across functional areas of the system.

- **Patient Management**

- The Patients page allows staff to add, edit, or delete patient profiles.
- Users can view each patient's profile, which contains personal information and medication records.
- Medication intake schedules can be assigned directly to individual patients.

- **Inventory Management**

- The Inventory page allows users to add, edit, or delete medicine records.
- All medicines listed here correspond to the medicines stored in the dispenser's 20 containers.
- Features include real-time visibility of medicine types inside the dispenser, monitoring of low-stock medicines, and identification of expired medicines.

- **Dispenser Control**

- The Dispenser page enables staff to manually dispense medicines from specific containers.
- Each container can be refilled with up to 32 capsules or tablets.
- Users can replace medicines in containers as needed and view the real-time status of each container.
- **Scheduled Dispensing and Monitoring**
  - The Dispensing page provides an overview of scheduled medication intake for patients.
  - If a medicine is not taken within 30 minutes of its scheduled time, the system marks it as “missed.”
  - A dispense history table records all dispensing events within the day for review and accountability.
- **Notifications and Alerts**
  - The Notifications page serves as a centralized log of real-time system notifications.
  - It records events such as low stock, upcoming medicine schedules, missed doses, and dispenser errors.
  - Notifications are structured in a table format for easy tracking and review.
- **Reports and Analytics**
  - The Reports page provides administrators with access to a variety of performance and usage reports.
  - These include activity distributions, daily revenue trends, cost category distributions, and medicine dispensing volume summaries.
  - Reports support decision-making and long-term resource planning.
- **Administrative Functions**
  - **User Management:** Administrators can create and manage user accounts, assigning roles and access permissions to staff.

- **System Logs:** The logs page allows administrators to monitor and audit all activities performed within the system, ensuring accountability and traceability.

## 2.3 Design and Implementation Constraints

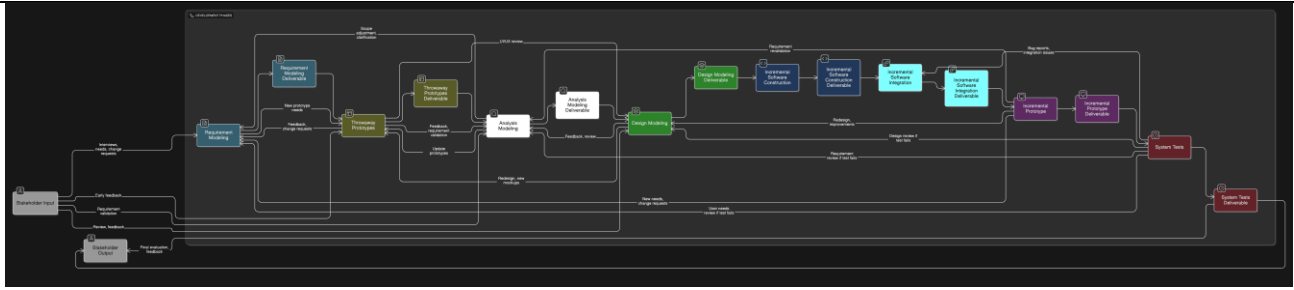
The development of AngelCare was guided by several constraints that shaped the design, implementation, and validation of the system. These include methodological, technical, business, and academic constraints. Each constraint ensured that the project stayed aligned with both client requirements and academic standards while remaining practical for deployment in a real-world healthcare setting.

### 2.3.1 COMET Methodology Constraint

The project followed the **COMET (Collaborative Object Modeling and Enterprise Techniques)** methodology as the prescribed approach for software development. COMET emphasizes iterative refinement, prototyping, and stakeholder involvement at every phase. For AngelCare, this meant:

- **Requirement Modeling:** Gathering user needs from Ms. Margelyn Gabbac (owner), nurses, and caregivers through interviews and observations.
- **Throwaway Prototypes:** Developing early drafts of the user interface and workflows, reviewed directly by stakeholders.
- **Analysis & Design Modeling:** Using structured models to define system behavior and architecture, validated against requirements.
- **Incremental Construction & Integration:** Building the system in modules (patients page, inventory, dispenser control, etc.) and integrating with the dispenser hardware step by step.
- **System Tests:** Validating the system through both stakeholder evaluation and academic panel reviews.

The constraint here was that every phase had to be backed by stakeholder review and documentation, which extended development time but ensured alignment with real-world caregiver needs



*Figure 2. COMET Diagram*

### 2.3.2 UML Modeling Constraint

The project was also constrained by the academic requirement to use **Unified Modeling Language (UML)** as the standard modeling language. UML was applied across multiple perspectives to clearly represent AngelCare’s functionality and structure:

- **Use Case Diagrams:** To illustrate how actors (Administrators, Caregivers, Patients, and Owner) interact with the system across major functions like inventory, dispensing, and reporting.
- **Class Diagrams:** To define the core entities (Patients, Medicines, Schedules, Users, Dispenser Containers) and their relationships, forming the backbone of the database schema.
- **Activity & Sequence Diagrams:** To depict workflows such as medicine scheduling, dispensing, and missed-dose handling, ensuring logical consistency in process flows.

This constraint provided a uniform way of documenting the system but required additional effort to translate evolving prototypes into UML diagrams

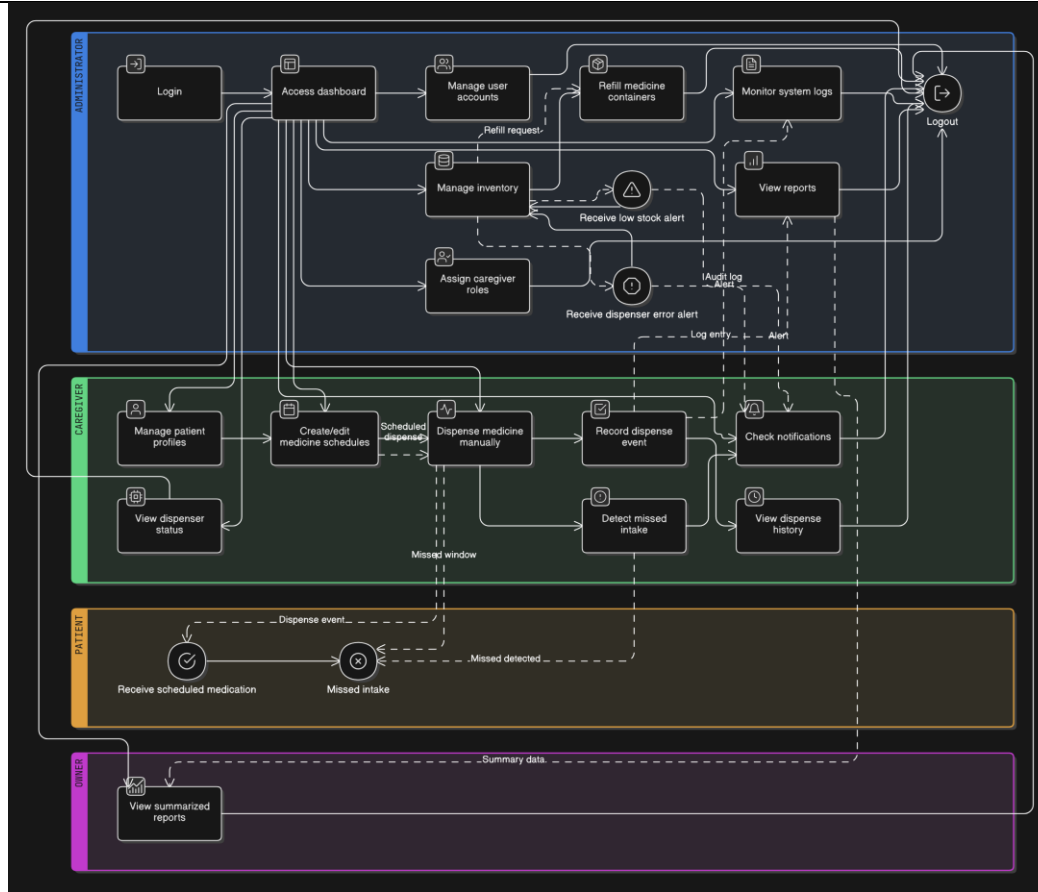


Figure 3. UML Diagram (Swim Lane Diagram)

### 2.3.3 Technical Constraints:

- **Technology Stack:** PHP and JavaScript were mandated for web development, with MySQL as the database.
- **Hardware Limitation:** Each dispenser container can store a maximum of **32 capsules or tablets**, requiring careful validation in scheduling and refilling logic.
- **Network Dependency:** A stable Wi-Fi connection is required for synchronization between the web system and the dispenser; an RTC fallback mitigates outages.

### 2.3.4 Business Rules and Operational Constraints

- **Dispensing Window:** Doses must be taken within 30 minutes or they are logged as missed.
- **Inventory Thresholds:** Automatic alerts are triggered when medicines reach a low stock or expiry.
- **Role Restrictions:** Only Administrators can manage users and logs, while Caregivers and Admins both manage patients and schedules.
- **Audit Trail:** Every action (adding patients, scheduling, dispensing, refilling) is permanently logged for accountability.

### 2.3.5 Academic and Evaluation Constraints

As a Capstone project, AngelCare was subject to **faculty and panelist evaluation** milestones. UML and COMET were required methodologies, and prototypes had to be demonstrated at specific checkpoints. This created academic constraints around deliverable deadlines and evaluation formats, in addition to technical considerations.

## 2.4 Assumptions and Dependencies

The development and operation of AngelCare rely on several assumptions and external dependencies. These factors, while not guaranteed, are expected to remain true for the system to function effectively. Any change to these assumptions or failure in dependencies may significantly affect system performance and reliability.

### Assumptions

- Caregivers and administrators possess basic computer literacy and are capable of navigating the AngelCare web interface without extensive training.
- A stable supply of electricity and Wi-Fi connectivity is available at Angel's Elderly Home Care Services to ensure continuous synchronization between the web system and the dispenser hardware.
- Medicines are consistently available for stocking the inventory and refilling the dispenser's 20 containers.
- The dispenser hardware will be properly maintained through regular refills, cleaning, and calibration by the staff.
- Users (staff and administrators) will provide accurate and updated information when entering patient records, medication schedules, and inventory data.

### Dependencies

- The system depends on reliable **web hosting services** and the availability of the project domain (angelcare.space) to ensure accessibility.
- The database relies on **MySQL** as its management system; any migration to a different DBMS would require adjustments in schema and application logic.
- The dispenser hardware depends on its **microcontroller firmware** to process and execute dispensing commands sent from the backend.
- The notification and activity log features depend on the internal **AngelCare web module**; failure of this module would prevent staff from receiving critical alerts.

- Future scalability and system improvements may depend on integration with **third-party services**, such as Electronic Health Records (EHRs), supplier APIs for automated ordering, and cloud-based backup solutions.

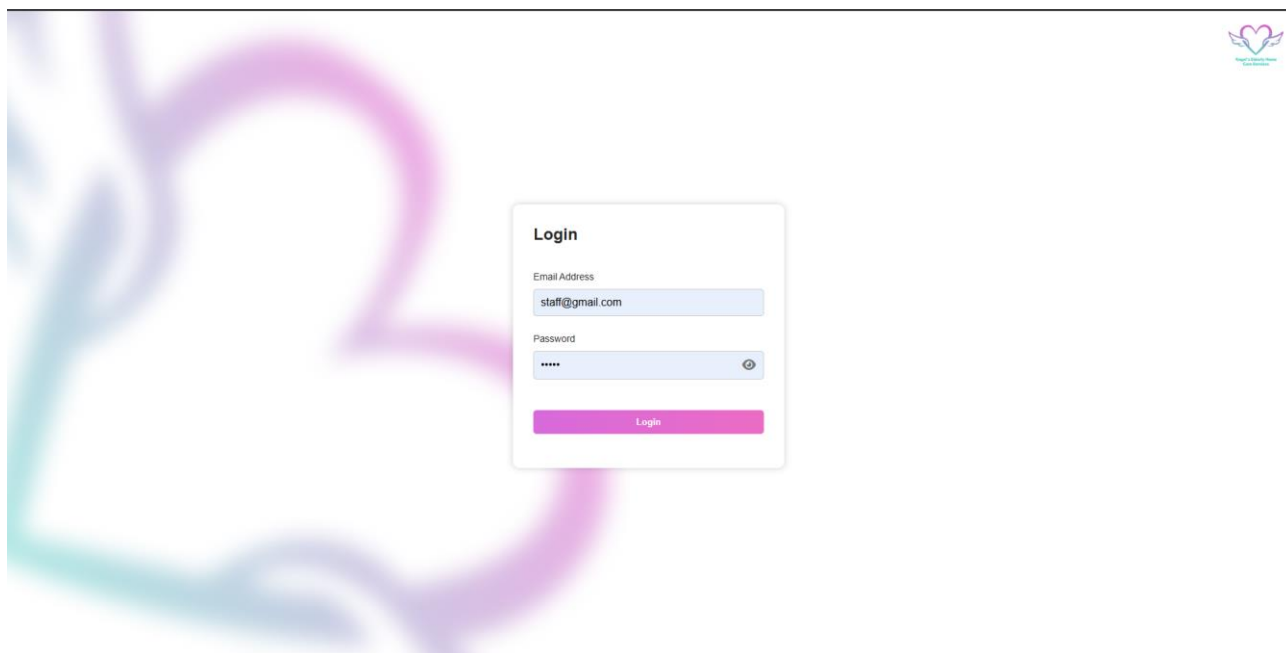
## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

The primary interface for AngelCare is a **web-based dashboard** accessible through the domain [angelcare.space](https://angelcare.space). Users interact with the system using standard input devices (keyboard and mouse for desktops, or touchscreen for tablets). The interface is designed to be simple, responsive, and intuitive to accommodate caregivers and administrators who may have varying levels of digital literacy.

#### STAFF SIDE:



*Figure 4. Login Page*

The login page at [angelcare.space/login.php](https://angelcare.space/login.php) provides secure access to the AngelCare system using unique credentials for each user. **Administrator accounts** are created by the system admin and grant full access to all modules, including user management, logs, and reporting. **Staff or caregiver accounts**, on the other hand, are limited to daily operational



functions such as managing patients, schedules, dispensing, and viewing notifications. This separation of credentials enforces role-based access control, ensuring security, accountability, and protection of sensitive administrative functions.

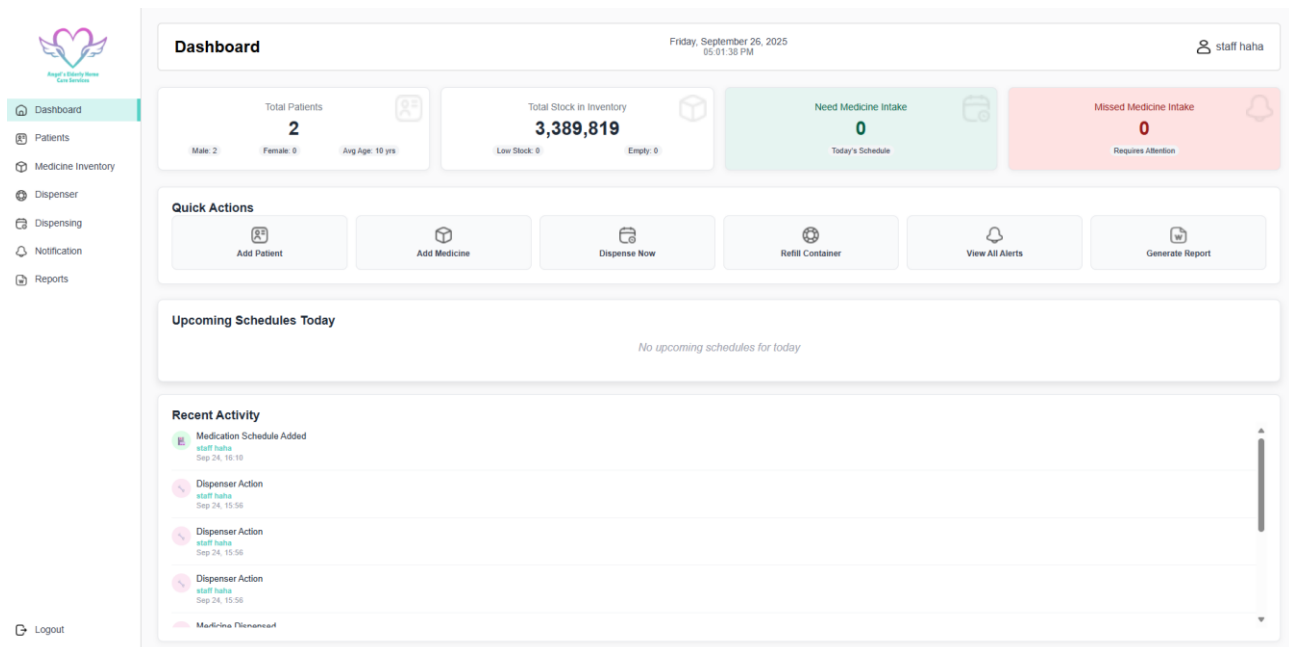


Figure 5. Dashboard Page

The dashboard presents a real-time overview of system operations, showing metrics like total patients, current inventory levels, and upcoming medicine schedules. It highlights problem indicators such as missed intakes, low stock alerts, and dispenser status to immediately draw user attention. It also features quick actions (shortcuts) and recent activity logs for rapid navigation and monitoring. A staff performance widget shows trends over the past seven days, giving insight into caregiver productivity and system usage.

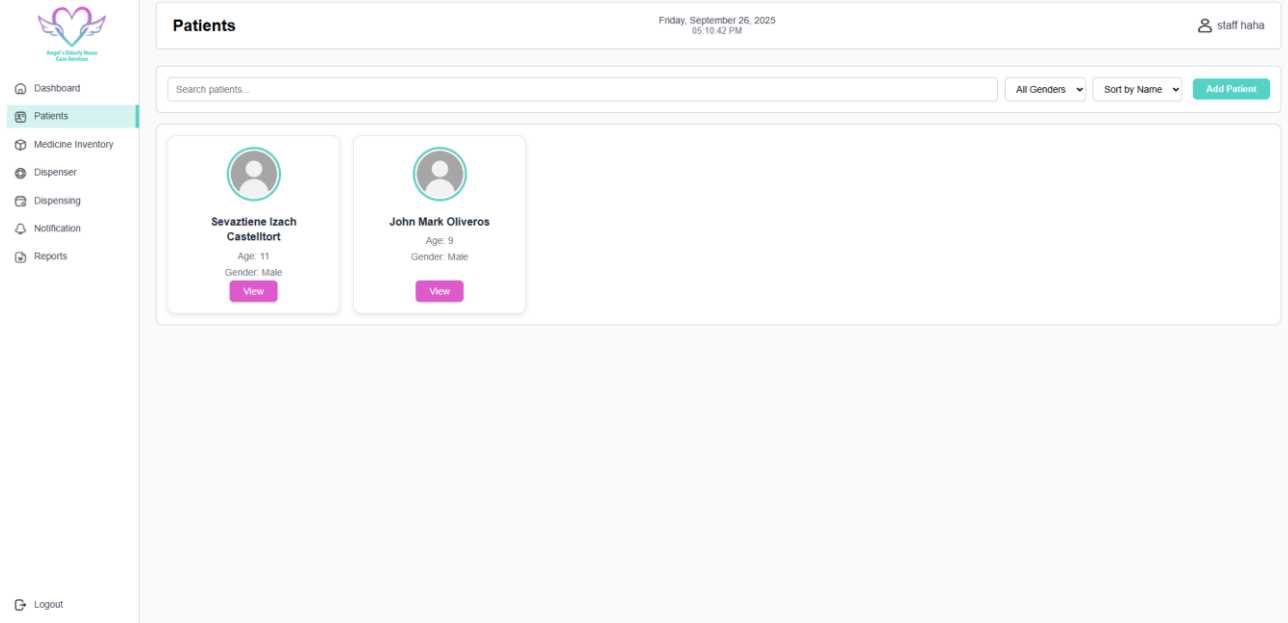


Figure 6. Patients Page

The Patients page lets users add new resident profiles or edit existing ones, capturing essential personal details and care information. From each profile, users can schedule or modify medication regimens tailor-made for individual residents. It also allows deletion of records when needed, under proper authorization. Finally, users can view a patient's full medication history and associated schedule data in one integrated interface.

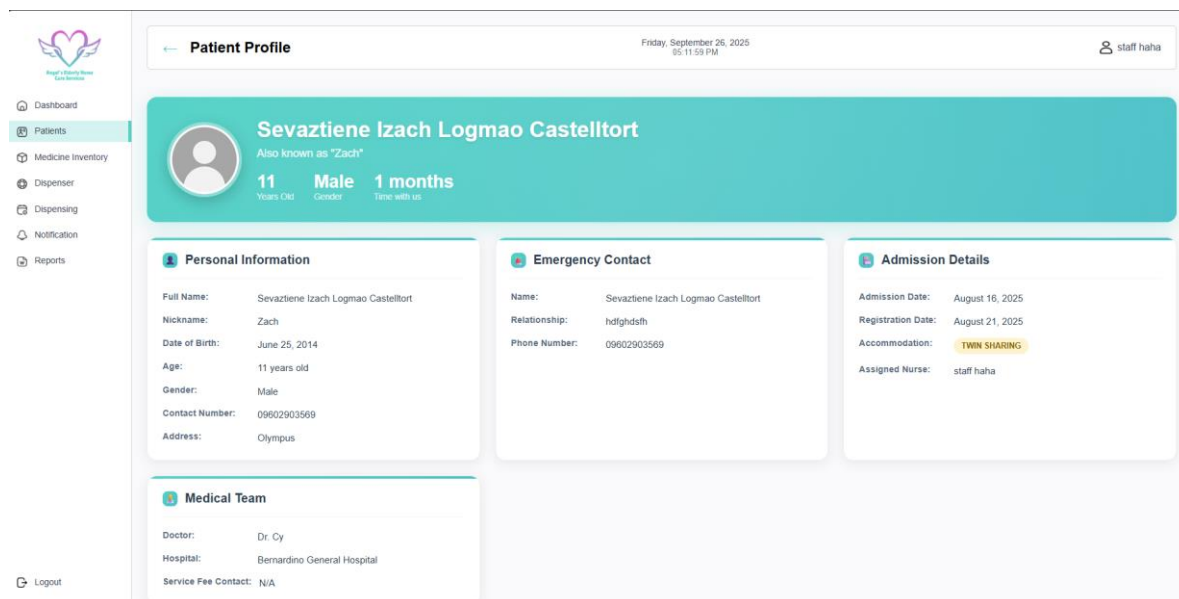


Figure 7. Patients Profile Page

The Patient Profile page displays detailed information about a resident, including their personal details, room or contact info, and medical notes or allergies. From this page, users can view, add, or modify the medication schedule specific to that resident. It shows a history of past dispenses and any missed or overridden doses for that patient. It also provides a convenient interface to monitor, update, or deactivate the resident's account when needed.

**Medicine Inventory** Friday, September 26, 2025 05:14:25 PM staff haha

24 Total Medicines Unique medicine types

0 Low Stock Need restocking

0 Expired Batches Require attention

Search Medicines Type medicine name or dosage...

Sort Order A to Z Filter All Medicines

+ Add Medicine + Add Stock

MEDICINE DETAILS	STOCK & PRICING	EXPIRATION & STATUS	ACTIONS
<b>Amiodipine</b> 300mg	95 total Threshold: 10		
Batch #3448592385	95 units \$3,000.00	2025-10-30 GOOD	Edit Delete
<b>Amiodipine</b> 500mg	4968 total Threshold: 10		
Batch #532423544335	4968 units \$2,000.00	2029-10-31 GOOD	Edit Delete
<b>Amoxicillin</b> 500mg	4698 total Threshold: 10		

Logout

**Figure 8. Inventory Management Page**

The Inventory Management page displays all medicines currently registered in the system with their batches, expiry dates, and current stock levels. Users can add new medicine items or edit the details of existing ones, including quantities and reorder thresholds. The page highlights medicines that are low in stock or have expired to prompt timely action. It also links the inventory items to their respective dispenser container positions, making it clear which medicines are installed in the physical dispenser.

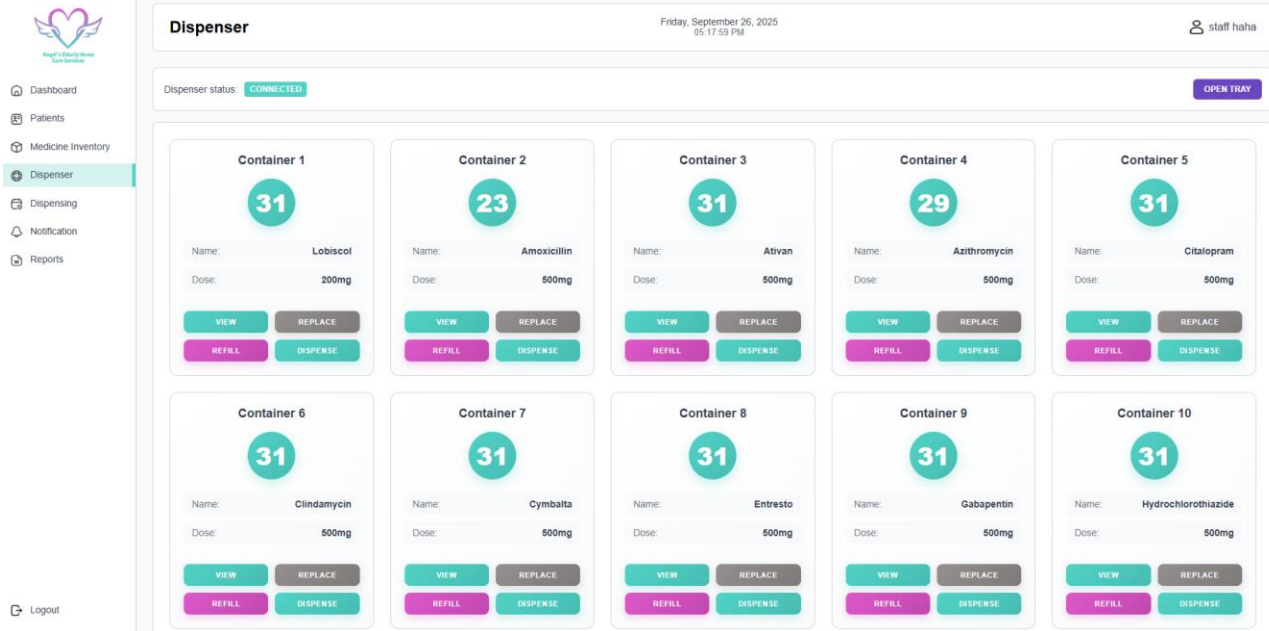


Figure 9. Dispenser Page

The Dispenser page provides a control panel for interacting with the physical medicine dispenser, showing each container's current status (e.g. empty, filled, or error). Users can manually trigger a dispense from a selected container and watch the operation in real time. It also allows refilling individual containers (up to the maximum capacity) or replacing medicines within them. Additionally, the page reflects diagnostic data such as recent dispense results or error codes to assist in maintenance and troubleshooting.

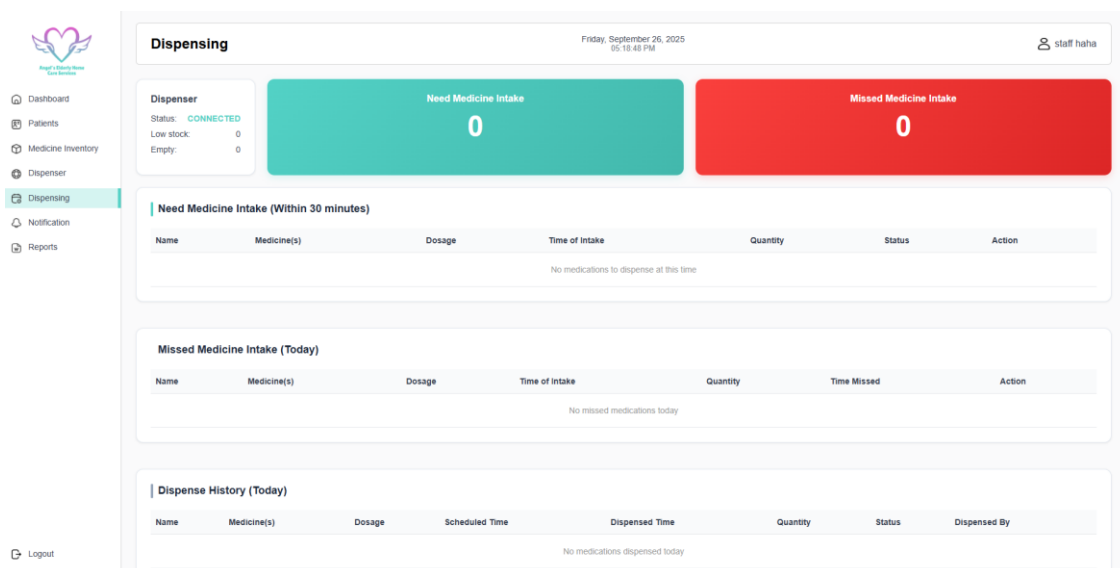


Figure 10. Dispensing Page

The Dispensing page displays all scheduled medication intakes for the day, allowing users to see which doses are due. It also marks any missed intakes if a scheduled medicine was not dispensed within the allowed time window. A history log of dispensed medicines for

the day is shown, with details such as patient name, dose amount, and time. Users can monitor upcoming schedules, review past dispenses, and track dispensing performance at a glance.

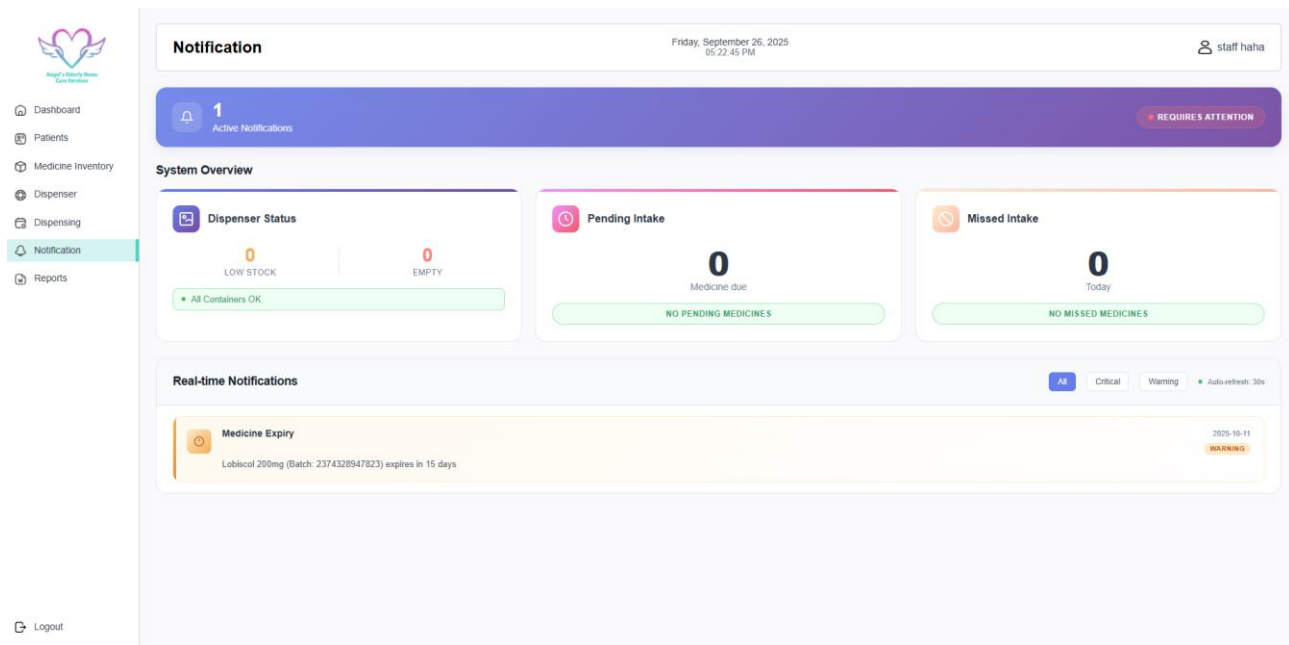


Figure 11. Notification Page

The Notification page shows a real-time list of alerts and system events relevant to dispensing and inventory. It includes notices such as low stock, missed doses, and dispenser errors. Each entry provides context (e.g. which medicine or container triggered the alert) so users can act appropriately. Users can filter and review past notifications to follow up on issues or audit operations.

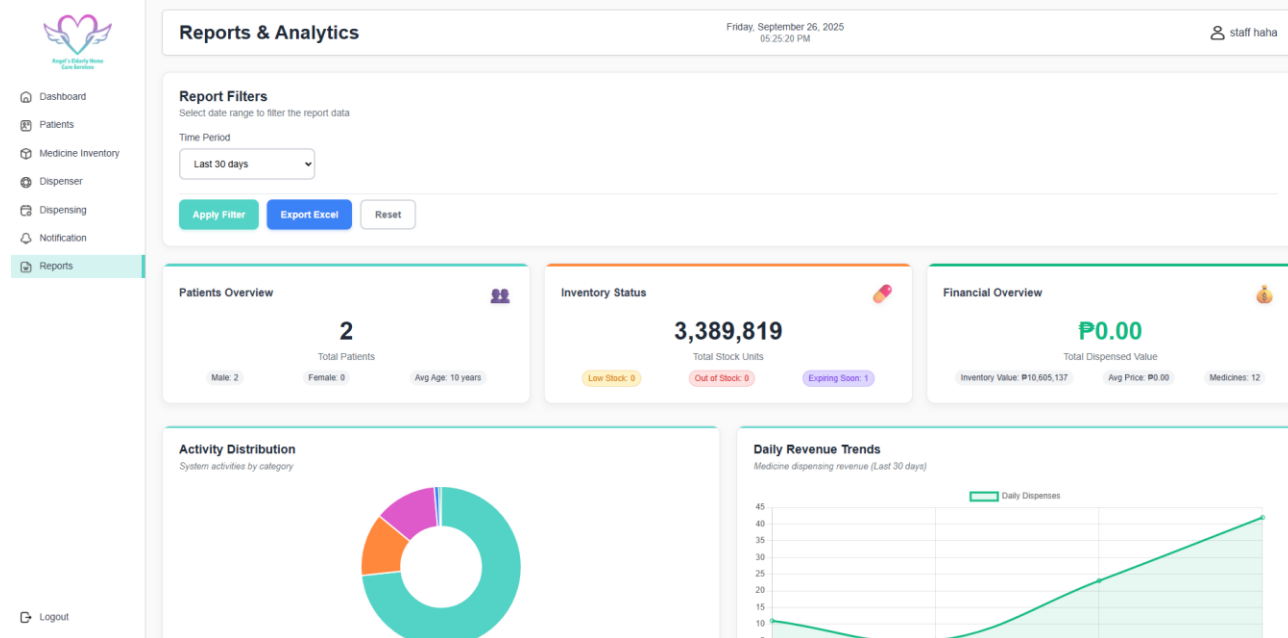


Figure 12. Reports Page

The Reports page presents various summaries and visualizations of system data over selectable time periods. It includes charts and tables detailing medicine usage, dispensing volumes, and inventory changes. Users can filter and export reports to review trends like low stock incidents, missed doses, or container performance. The page helps administrators make informed decisions about procurement, staffing, and system adjustments.

## ADMIN SIDE:

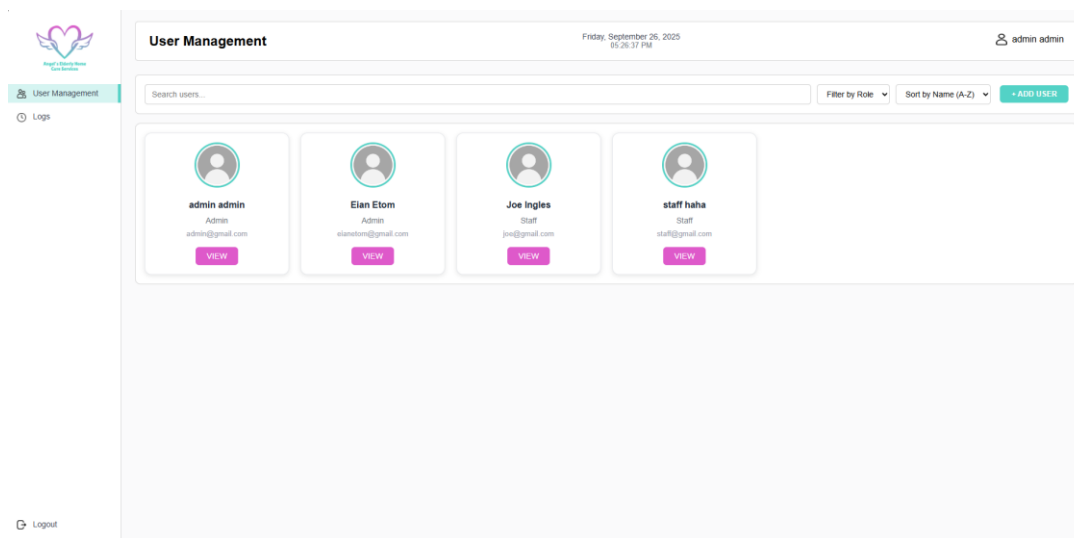
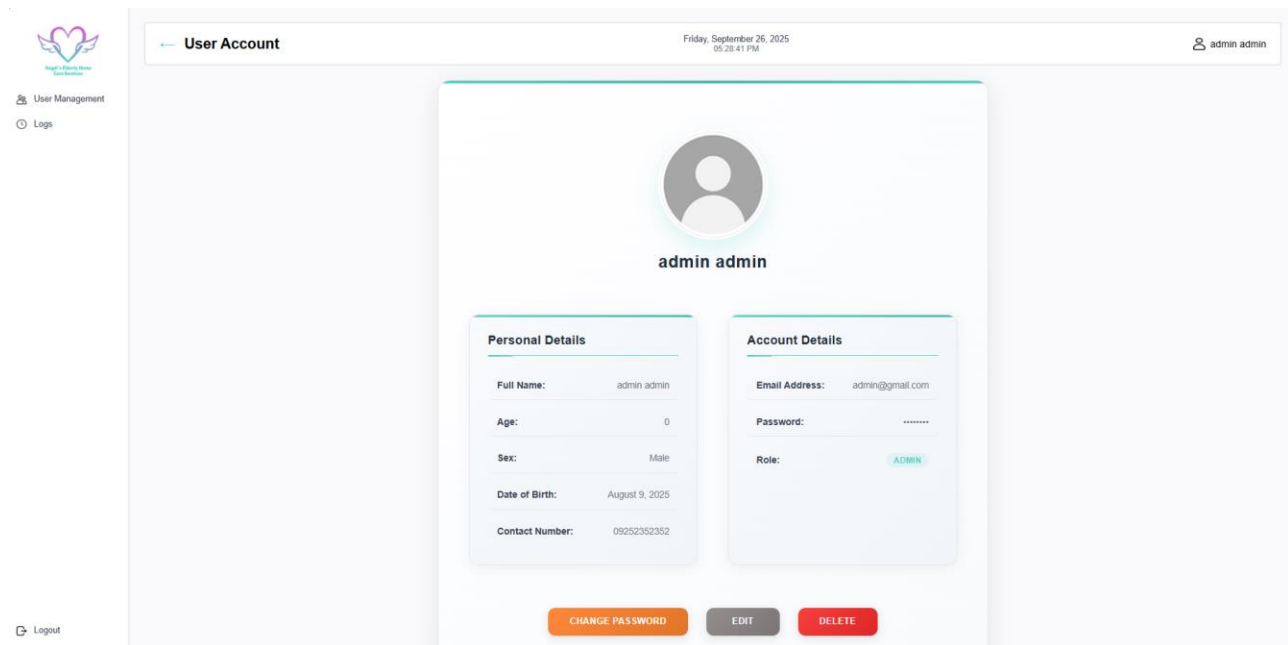


Figure 13. User Management Page

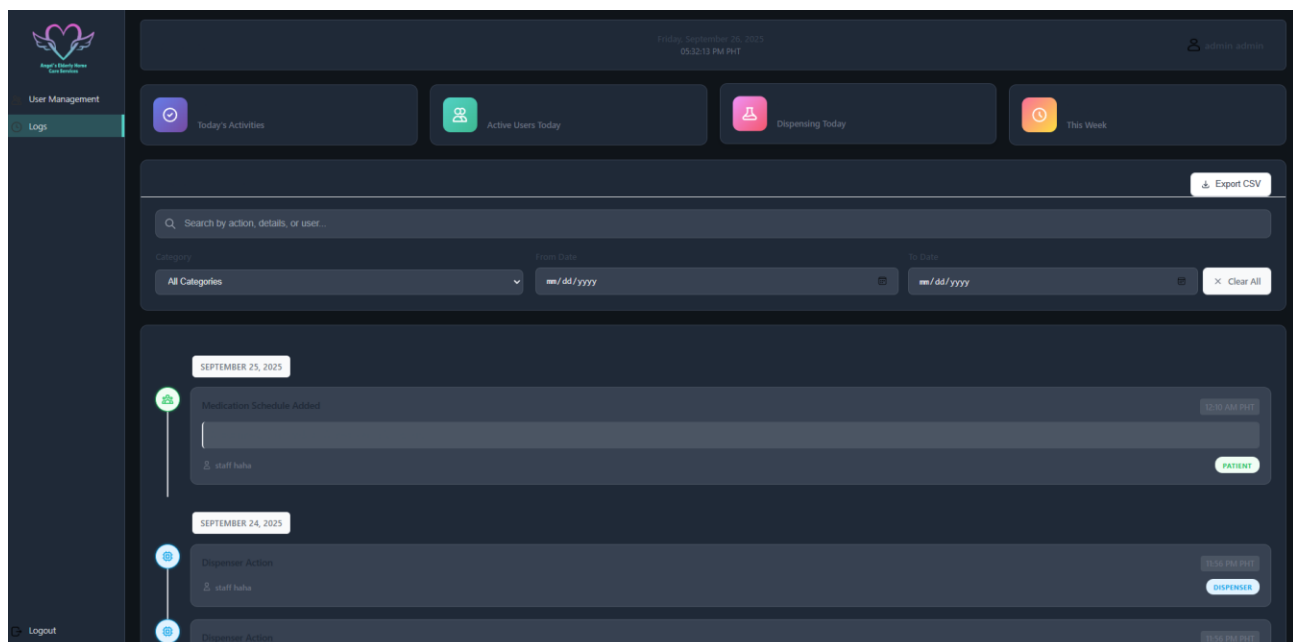
The User Management page allows administrators to create, edit, or delete staff accounts and assign roles to each user. It displays a list of active users with details such as username, role, and status (active/inactive). Administrators can reset passwords, deactivate accounts, or

modify user permissions from this interface. This page ensures that only authorized personnel have access to appropriate modules, maintaining the security and integrity of the system.



**Figure 14. User Account Page**

The User Account page displays personal details for a selected user, such as full name, username, email, and role. It allows administrators to modify these details or update the user's password and access permissions.



**Figure 15. User Management Page**

The Logs page displays a chronological record of system activities performed by all users, including actions like creating or editing patient profiles, dispensing medicines, and modifying

inventory. Each log entry shows relevant details such as timestamp, user identity, action performed, and affected entities. Administrators can filter logs by date range, user, or activity type to locate specific events. This interface supports accountability and auditability by preserving a transparent record of all changes and interactions.

### 3.1.2 Hardware Interfaces

The AngelCare system interfaces with several hardware components that enable automated medicine dispensing. The software communicates with these devices through commands, confirmations, and control signals to ensure scheduled doses are executed correctly.

- **Arduino Uno and Arduino Mega:** These microcontrollers act as the core control units, interpreting commands from the backend and coordinating the operation of motors within the dispenser.
- **ESP32 Module:** Provides Wi-Fi connectivity, enabling real-time communication between the web application and the dispenser hardware for scheduling and confirmation of dispense events.
- **Stepper Motors:** Operate the container mechanisms by rotating in precise increments to dispense the required number of tablets or capsules from each container.
- **Servo Motors:** Used for specific container operations requiring angular movement, such as controlling flaps or gates during the dispensing process.

Together, these hardware components create a seamless interface between the AngelCare web system and the physical dispenser, ensuring that digital schedules translate into accurate, real-world dispensing actions.





*Figure 16. Full Hardware Interface*



*Figure 17. Arduino Uno*

The **Arduino Uno** serves as one of the primary microcontrollers in the AngelCare dispenser, acting as an intermediary between the web application and the mechanical components. It receives commands from the backend via the ESP32 module and translates them into electrical signals that drive stepper and servo motors. The Uno also manages timing and sequencing to ensure the correct container releases the right amount of medicine. Its role

is critical in ensuring that digital instructions are accurately executed in the physical dispensing process.



Figure 18. Arduino Mega

The **Arduino Mega** is used in the AngelCare dispenser to handle multiple I/O operations, making it suitable for managing the system's 20 medicine containers. It processes control signals from the backend and coordinates the operation of numerous stepper and servo motors simultaneously. Its larger memory and greater number of pins compared to the Arduino Uno allow it to manage more complex dispensing tasks reliably. The Mega ensures scalability and stability in handling high-demand dispensing operations within the system.



Figure 19. Stepper Motor

The **stepper motors** are responsible for rotating the 20 medicine containers within the dispenser to release tablets or capsules. Each motor receives precise step signals from the Arduino controllers, ensuring controlled movement for accurate dispensing. The motors allow the system to align and dispense the correct dosage without requiring manual measurement.

By providing reliable and repeatable motion, the stepper motors form the backbone of the dispenser's mechanical operation.



Figure 20. Servo Motor

The **12kg servo motor** is used to control the automatic and manual closing of the medicine tray in the dispenser. It receives angular position commands from the Arduino controllers, allowing it to open or shut the tray smoothly after each dispensing action. This ensures that medicines remain secure inside the dispenser until retrieved by the caregiver or patient. Its strength and precision make it reliable for repeated operations, maintaining both safety and efficiency in the dispensing process.



Figure 21. ESP32 WiFi-Module

The **ESP32 Wi-Fi module** enables real-time communication between the dispenser hardware and the AngelCare web application. It transmits dispensing commands from the backend to the Arduino controllers and sends back confirmation messages after each operation.

This wireless interface allows the system to synchronize schedules, update inventory records, and log activities without the need for wired connections. By serving as the communication bridge, the ESP32 ensures seamless integration of hardware actions with the software platform.

### 3.1.3 Software Interfaces

Table 1. Software Interfaces

Software Component	Description
<b>Web Client (Frontend)</b>	Accessed through a standard web browser at <a href="http://angelcare.space">angelcare.space</a> . Provides dashboards, forms, and tables for Administrators and Caregivers to manage patients, schedules, inventory, dispensing, notifications, and reports.
<b>Backend (PHP Application Server)</b>	Implements the business logic of the system. Processes user requests, validates input, enforces role-based access, and transmits commands to the dispenser hardware through the ESP32 module.
<b>Database (MySQL with phpMyAdmin)</b>	MySQL stores users, patient records, schedules, inventory, logs, and dispense history. phpMyAdmin serves as the management tool, providing a graphical interface for executing SQL queries, backups, and schema maintenance.
<b>ESP32 Communication Layer</b>	Connects the backend to the dispenser hardware. Receives dispensing commands from the backend, relays them to Arduino controllers, and returns confirmation or error messages that are logged into the database.
<b>Arduino IDE</b>	The development environment used to write, compile, and upload embedded C/C++ code to Arduino Uno, Arduino Mega, and ESP32. Ensures hardware is properly programmed for dispensing, motor control, and communication tasks.

## 3.2 Functional Requirements

### A. Software Modules

Table 2. Software Modules

Module / Page	Functional Requirement	Hardware–Software Connection
<b>Login Page</b>	Authenticate users with unique credentials. Redirect Admins and Staff to role-based dashboards. Log failed login attempts for security.	Software-only; credentials stored in MySQL (via phpMyAdmin).
<b>Dashboard</b>	Display real-time overview: total patients, inventory, upcoming schedules, dispenser status, notifications, and staff performance.	Dispenser status updated via ESP32 feedback, other data from MySQL.
<b>Patients Page</b>	Add, edit, and delete patient profiles. Assign medication schedules. View personal and medical details.	Scheduling data stored in MySQL and later linked to dispenser commands.
<b>Patient Profile Page</b>	Show complete patient info including medical notes, allergies, schedules, and dispense history.	Dispense history synced from ESP32 confirmations logged in MySQL.
<b>Inventory Page</b>	Add, edit, and delete medicines. Track stock, expirations, and container assignments. Alert for low or expired stock.	Inventory data stored in MySQL; container mapping linked to dispenser hardware.
<b>Dispenser Page</b>	Manually trigger dispensing, refill containers, replace	Backend sends commands via ESP32 → Arduino → Motors.

	medicines, and check container status.	
<b>Dispensing Page</b>	Display today's schedules, record missed intakes, and show daily dispense history.	Schedules from MySQL; dispenser confirmations sent back to database via ESP32.
<b>Notification Page</b>	List real-time alerts (low stock, missed doses, dispenser errors). Allow review of past alerts.	Alerts generated by backend using MySQL triggers and ESP32 error codes.
<b>Reports Page</b>	Generate/export reports on medicine usage, activity logs, and trends.	Aggregated data from MySQL; raw data includes dispenser confirmations.
<b>User Management Page</b>	Admins add, edit, or delete staff accounts. Assign roles and reset passwords.	Software-only; user data managed in MySQL/phpMyAdmin.
<b>User Account Page</b>	Display or update user details including role, status, and last login.	Software-only; linked to MySQL records.
<b>Logs Page</b>	Record every action with timestamp, user, and activity type. Provide filtering by user/date.	Logs written to MySQL by backend; includes dispenser confirmations via ESP32.

## B. Hardware Components

Table 3. Hardware Modules

Hardware Component	Functional Requirement	Software–Hardware Connection
<b>Arduino Uno</b>	Executes low-level commands for dispensing. Manages stepper/servo motor control.	Receives instructions from backend (via ESP32) and drives motors accordingly.

<b>Arduino Mega</b>	Coordinates multiple container operations using expanded I/O capacity.	Handles parallel motor control; receives backend commands via ESP32.
<b>Stepper Motors</b>	Rotate medicine containers to release tablets/capsules accurately.	Controlled by Arduino Uno/Mega per backend instructions; feedback logged in MySQL.
<b>12kg Servo Motor</b>	Opens and closes the tray automatically or manually.	Commanded by Arduino Uno/Mega; tray status updates reflected in backend logs.
<b>ESP32 Wi-Fi Module</b>	Provides wireless communication between web system and dispenser hardware.	Relays backend commands to Arduino and returns confirmation/error codes to MySQL.

### 3.3 Use Case Model

The Use Case Diagram for AngelCare shows how the system interacts with its two actors: **Administrator** and **Staff**. The Administrator is responsible for user management and logs, while the Staff manages patients, inventory, dispensing, reports, notifications, and the dashboard. This diagram highlights the role-based access and the main functions available to each actor.

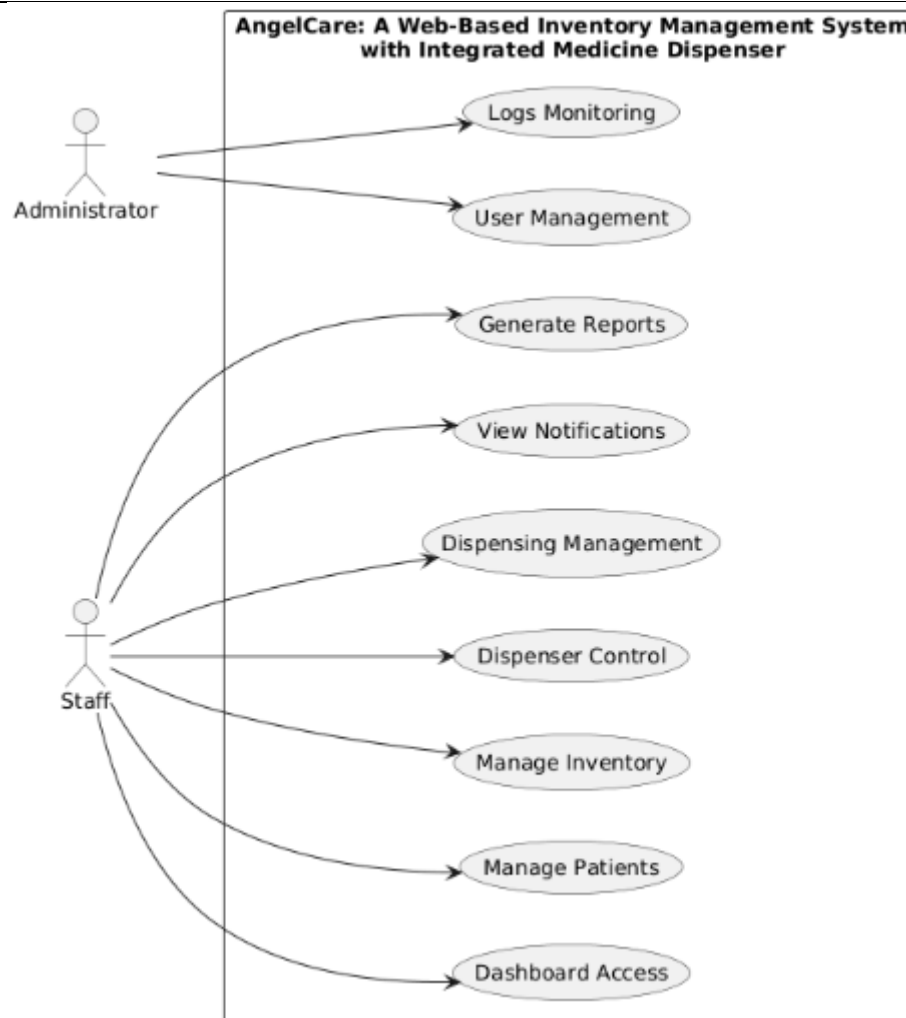


Figure 21. Use Case Diagram

**STAFF SIDE USE CASE DESCRIPTIONS:****3.3.1 Use Case #1**

Table 4. Test Case 1

<b>Use Case ID</b>	1.0 Dashboard (U1)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	Allow staff to quickly view summaries such as patient totals, inventory levels, dispenser status, upcoming medicine schedules, and recent activities.
<b>Requirements Traceability</b>	FR-STAFF-001, FR-STAFF-002
<b>Priority:</b>	Dashboard serves as the primary monitoring hub.
<b>Extends</b>	None.
<b>Includes</b>	U2 (Patients), U3 (Inventory), U5 (Dispensing).
<b>Actors</b>	Staff, Database.



<b>Preconditions:</b>	Staff must be logged in with valid credentials.								
<b>Postconditions:</b>	Dashboard data is successfully retrieved and displayed.								
<b>Flow of Activities:</b>	<table border="1"> <tr> <th colspan="2">Flow</th></tr> <tr> <td>1. <b>Basic Flow:</b></td><td>Staff logs in → navigates to Dashboard → system queries database → dashboard metrics are displayed.</td></tr> <tr> <td>2. <b>Alternative Flow:</b></td><td>Dashboard partially loads if some metrics are unavailable.</td></tr> <tr> <td>3. <b>Exceptions:</b></td><td>Database/server downtime prevents data retrieval.</td></tr> </table>	Flow		1. <b>Basic Flow:</b>	Staff logs in → navigates to Dashboard → system queries database → dashboard metrics are displayed.	2. <b>Alternative Flow:</b>	Dashboard partially loads if some metrics are unavailable.	3. <b>Exceptions:</b>	Database/server downtime prevents data retrieval.
Flow									
1. <b>Basic Flow:</b>	Staff logs in → navigates to Dashboard → system queries database → dashboard metrics are displayed.								
2. <b>Alternative Flow:</b>	Dashboard partially loads if some metrics are unavailable.								
3. <b>Exceptions:</b>	Database/server downtime prevents data retrieval.								

### 3.3.2 Use Case #2

Table 5. Test Case 2

<b>Use Case ID</b>	2.0 Patients (U2)								
<b>Author:</b>	Castelltort, Sevaztiene Izach								
<b>Purpose</b>	allow staff to add, edit, delete, and manage patient profiles and medication schedules.								
<b>Requirements Traceability</b>	FR-STAFF-003, FR-STAFF-004								
<b>Priority:</b>	Patient management is essential for medicine dispensing.								
<b>Extends</b>	None.								
<b>Includes</b>	U5 (Dispensing).								
<b>Actors</b>	Staff, Database.								
<b>Preconditions:</b>	Staff must be logged in.								
<b>Postconditions:</b>	Patient data is updated in the database and linked to schedules.								
<b>Flow of Activities:</b>	<table border="1"> <tr> <th colspan="2">Flow</th></tr> <tr> <td>1. <b>Basic Flow:</b></td><td>Staff selects "Patients" → chooses add/edit/delete → system validates input → database updates.</td></tr> <tr> <td>2. <b>Alternative Flow:</b></td><td>Staff updates only partial patient information (e.g., contact info).</td></tr> <tr> <td>3. <b>Exceptions:</b></td><td>Duplicate patient, invalid/missing info, or database error.</td></tr> </table>	Flow		1. <b>Basic Flow:</b>	Staff selects "Patients" → chooses add/edit/delete → system validates input → database updates.	2. <b>Alternative Flow:</b>	Staff updates only partial patient information (e.g., contact info).	3. <b>Exceptions:</b>	Duplicate patient, invalid/missing info, or database error.
Flow									
1. <b>Basic Flow:</b>	Staff selects "Patients" → chooses add/edit/delete → system validates input → database updates.								
2. <b>Alternative Flow:</b>	Staff updates only partial patient information (e.g., contact info).								
3. <b>Exceptions:</b>	Duplicate patient, invalid/missing info, or database error.								

### 3.3.3 Use Case #3

Table 6. Test Case 3

<b>Use Case ID</b>	3.0 Inventory (U3)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	Manage medicines by adding, editing, refilling, or deleting items in the inventory.
<b>Requirements Traceability</b>	FR-STAFF-005, FR-STAFF-006
<b>Priority:</b>	High – Inventory accuracy ensures correct dispensing.
<b>Extends</b>	None.
<b>Includes</b>	U4 (Dispenser Control).
<b>Actors</b>	Staff, Database, Dispenser.
<b>Preconditions:</b>	Staff logged in with valid role.
<b>Postconditions:</b>	Inventory records are stored and linked to dispenser containers.
<b>Flow of Activities:</b>	<b>Flow</b>
	<ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> Staff selects Inventory → adds or updates medicines → system validates inputs → records saved.</li> <li>2. <b>Alternative Flow:</b> Inventory flagged for low stock/expired medicines.</li> <li>3. <b>Exceptions:</b> Invalid entries, wrong container assignment.</li> </ol>

### 3.3.4 Use Case #4

Table 7. Test Case 4

<b>Use Case ID</b>	4.0 Dispenser (U4)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	manually control containers in the dispenser, refill or replace medicines, and view container status.
<b>Requirements Traceability</b>	FR-STAFF-007, FR-STAFF-008
<b>Priority:</b>	Medium – Used primarily for manual overrides.
<b>Extends</b>	None.
<b>Includes</b>	U3 (Inventory).
<b>Actors</b>	Staff, Dispenser, Database.

<b>Preconditions:</b>	Staff logged in; dispenser connected.
<b>Postconditions:</b>	Dispenser executes command; logs updated.
<b>Flow of Activities:</b>	<p style="text-align: center;"><b>Flow</b></p> <ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> Staff selects container → issues “Dispense/Refill/Replace” → system sends command → dispenser responds.</li> <li>2. <b>Alternative Flow:</b> Container already full; only replacement allowed.</li> <li>3. <b>Exceptions:</b> Motor jam, connectivity issue, invalid command.</li> </ol>

### 3.3.5 Use Case #5

Table 8. Test Case 5

<b>Use Case ID</b>	5.0 Dispensing (U5)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	View, confirm, or monitor scheduled and missed medicine intakes.
<b>Requirements Traceability</b>	FR-STAFF-009, FR-STAFF-010
<b>Priority:</b>	High – Directly tied to patient care.
<b>Extends</b>	None.
<b>Includes</b>	U2 (Patients), U3 (Inventory).
<b>Actors</b>	Staff, Database, Dispenser.
<b>Preconditions:</b>	Staff logged in; schedules already set.
<b>Postconditions:</b>	Dispense event is recorded and inventory updated.
<b>Flow of Activities:</b>	<p style="text-align: center;"><b>Flow</b></p> <ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> System checks schedules → dispenser executes command → staff monitors results.</li> <li>2. <b>Alternative Flow:</b> Staff performs manual dispense if automatic fails.</li> <li>3. <b>Exceptions:</b> Missed intake not logged; dispenser fails.</li> </ol>

### 3.3.6 Use Case #6

Table 9. Test Case 6

<b>Use Case ID</b>	6.0 Notification (U6)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	To inform staff in real-time about system events such as low stock, missed doses, or dispenser errors.
<b>Requirements Traceability</b>	FR-STAFF-011
<b>Priority:</b>	High – Ensures responsive action by staff..
<b>Extends</b>	None.
<b>Includes</b>	U3 (Inventory), U5 (Dispensing).
<b>Actors</b>	Staff, System.
<b>Preconditions:</b>	Staff logged in.
<b>Postconditions:</b>	Notifications displayed in system log.
<b>Flow of Activities:</b>	<b>Flow</b>
	<ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> System detects event → generates notification → staff views alert.</li> <li>2. <b>Alternative Flow:</b> Staff filters notifications by category.</li> <li>3. <b>Exceptions:</b> Notifications fail to load in real-time.</li> </ol>

### 3.3.7 Use Case #7

Table 10. Test Case 7

<b>Use Case ID</b>	7.0 Reports (U7)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	Allow staff to generate and export reports on medicine usage, dispensing activity, and staff performance.
<b>Requirements Traceability</b>	FR-STAFF-012, FR-STAFF-013
<b>Priority:</b>	Medium – Important for auditing and decision-making.

<b>Extends</b>	None.
<b>Includes</b>	U5 (Dispensing), U6 (Notifications). U9 (Logs)
<b>Actors</b>	Staff, Database.
<b>Preconditions:</b>	Staff logged in.
<b>Postconditions:</b>	Report generated and available for export (PDF/CSV).
<b>Flow of Activities:</b>	<b>Flow</b>
	<ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> Staff requests report → system compiles data → report displayed/exported.</li> <li>2. <b>Alternative Flow:</b> Staff applies filters for specific dates or patients.</li> <li>3. <b>Exceptions:</b> No data available for selected timeframe.</li> </ol>

## Administrator Use Case Descriptions:

### 3.3.8 Use Case #8

Table 11. Test Case 8

<b>Use Case ID</b>	8.0 User Management (U8)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	To allow the administrator to add, edit, or remove system users and assign roles.
<b>Requirements Traceability</b>	FR-ADMIN-001
<b>Priority:</b>	High – Ensures secure access.
<b>Extends</b>	None.
<b>Includes</b>	None
<b>Actors</b>	Admin, Database.
<b>Preconditions:</b>	Admin logged in.
<b>Postconditions:</b>	User accounts updated in database.
<b>Flow of Activities:</b>	<b>Flow</b>
	<ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> Admin selects User Management → adds/edits/deletes user → system validates → database updated.</li> <li>2. <b>Alternative Flow:</b> Resetting a password instead of full edit.</li> <li>3. <b>Exceptions:</b> Duplicate username, invalid inputs.</li> </ol>

### 3.3.9 Use Case #9

Table 12. Test Case 9

<b>Use Case ID</b>	9.0 Reports (U9)
<b>Author:</b>	Castelltort, Sevaztiene Izach
<b>Purpose</b>	To monitor all activities and track user actions within the system.
<b>Requirements Traceability</b>	FR-ADMIN-002
<b>Priority:</b>	Medium – Important for auditing and accountability.
<b>Extends</b>	None.
<b>Includes</b>	None.
<b>Actors</b>	Admin, Database.
<b>Preconditions:</b>	Admin logged in.
<b>Postconditions:</b>	User accounts updated in database.
<b>Flow of Activities:</b>	<b>Flow</b>
	<ol style="list-style-type: none"> <li>1. <b>Basic Flow:</b> Admin selects User Management → adds/edits/deletes user → system validates → database updated.</li> <li>2. <b>Alternative Flow:</b> Resetting a password instead of full edit.</li> <li>3. <b>Exceptions:</b> Duplicate username, invalid inputs.</li> </ol>

## 4 Other Non-functional Requirements

### 4.1 Performance Requirements

Table 13. Performance Requirements

ID	Requirement	Target / Timing
P1	After login, the dashboard should render and display essential metrics (patients, medicine stock, schedules, and dispenser status).	Maximum of <b>3 seconds</b>
P2	The dispenser must carry out scheduled dispensing commands sent from the backend.	<b>10–20 seconds</b> , depending on internet connection speed

P3	Manual dispensing actions triggered by staff must be executed by the dispenser and acknowledged by the system.	Within <b>1.5 seconds</b>
P4	System alerts such as low stock, missed doses, or hardware faults should appear on the Notifications page immediately after being logged.	Within <b>1 second</b>
P5	The inventory count must reflect the updated stock after each dispensing transaction.	Within <b>1 second</b>
P6	Reports (e.g., usage logs, dispensing history, staff performance) should be compiled and exported in PDF/CSV format.	Within <b>5 seconds</b> for datasets up to 1,000 records
P7	The platform must remain responsive under concurrent usage.	At least <b>10 staff users</b> with $\leq 2$ <b>seconds</b> delay per request
P8	In case of network downtime, the dispenser's RTC fallback should continue scheduled dispensing and synchronize pending records once the connection is restored.	Sync completed within <b>10–20 seconds</b> , subject to internet speed

## 4.2 Safety and Security Requirements

Table 14. Safety and Security

<b>ID</b>	<b>Requirement</b>	<b>Description / Safeguard</b>
<b>S1</b>	<b>User Authentication</b>	All users must log in with unique credentials. Admin and Staff roles must be clearly separated to prevent unauthorized access to critical functions.
<b>S2</b>	<b>Data Privacy</b>	Patient information (profiles, medication history, schedules) must be securely stored and only accessible to authorized staff, in compliance with the <b>Philippine Data Privacy Act (RA 10173)</b> .
<b>S3</b>	<b>Secure Connections</b>	All communication between the web application, database, and dispenser hardware must use secure channels (e.g., HTTPS/TLS) to prevent interception or tampering.

<b>S4</b>	<b>Audit Trail</b>	Every user action (login, adding/editing patient data, dispensing commands) must be logged in an immutable format to maintain accountability.
<b>S5</b>	<b>Medication Safeguards</b>	The dispenser must prevent dispensing when a container is empty, expired, or mismatched with the patient's schedule. Warnings should be issued immediately.
<b>S6</b>	<b>Hardware Fail-Safes</b>	Stepper and servo motors must automatically stop during errors (e.g., jams) to avoid incorrect doses or mechanical failures.
<b>S7</b>	<b>Access Restriction</b>	Only Admin can manage users and logs. Staff access is limited to patient care, inventory, dispenser operations, notifications, and reports.
<b>S8</b>	<b>Data Backup &amp; Recovery</b>	Daily backups of the database must be maintained. Data recovery must be possible within <b>24 hours</b> in case of accidental loss.
<b>S9</b>	<b>Missed Dose Alerts</b>	The system must notify staff within <b>1 minute</b> if a patient misses a scheduled dose, to ensure immediate corrective action.
<b>S10</b>	<b>Mobile/Remote Security</b>	If mobile access is later introduced, connections must also use authentication tokens and encrypted sessions to prevent unauthorized access outside the facility.

### 4.3 Software Quality Attributes

Table 15. Software Quality Attributes

<b>ID</b>	<b>Requirement</b>	<b>Description / Safeguard</b>
<b>S1</b>	<b>User Authentication</b>	All users must log in with unique credentials. Admin and Staff roles must be clearly separated to prevent unauthorized access to critical functions.
<b>S2</b>	<b>Data Privacy</b>	Patient information (profiles, medication history, schedules)



		must be securely stored and only accessible to authorized staff, in compliance with the <b>Philippine Data Privacy Act (RA 10173)</b> .
<b>S3</b>	<b>Secure Connections</b>	All communication between the web application, database, and dispenser hardware must use secure channels (e.g., HTTPS/TLS) to prevent interception or tampering.
<b>S4</b>	<b>Audit Trail</b>	Every user action (login, adding/editing patient data, dispensing commands) must be logged in an immutable format to maintain accountability.
<b>S5</b>	<b>Medication Safeguards</b>	The dispenser must prevent dispensing when a container is empty, expired, or mismatched with the patient's schedule. Warnings should be issued immediately.
<b>S6</b>	<b>Hardware Fail-Safes</b>	Stepper and servo motors must automatically stop during errors (e.g., jams) to avoid incorrect doses or mechanical failures.
<b>S7</b>	<b>Access Restriction</b>	Only Admin can manage users and logs. Staff access is limited to patient care, inventory, dispenser operations, notifications, and reports.
<b>S8</b>	<b>Data Backup &amp; Recovery</b>	Daily backups of the database must be maintained. Data recovery must be possible within <b>24 hours</b> in case of accidental loss.
<b>S9</b>	<b>Missed Dose Alerts</b>	The system must notify staff within <b>1 minute</b> if a patient misses a scheduled dose, to ensure immediate corrective action.
<b>S10</b>	<b>Mobile/Remote Security</b>	If mobile access is later introduced, connections must also use authentication tokens and encrypted sessions to prevent unauthorized access outside the facility.

## 5 Other Requirements

### 5.1 Database Requirements

The system shall utilize MySQL as its primary database engine, with phpMyAdmin serving as the interface for schema management, monitoring, and maintenance tasks. All records—such as patient profiles, caregiver assignments, medicine inventory, intake schedules, dispensing history, and system notifications—shall be automatically timestamped in Philippine Standard Time (PST) to maintain uniformity and consistency across the system. To safeguard against accidental data loss or corruption, daily automated backups must be performed and stored securely for a minimum of 30 days. Furthermore, database queries and indexing strategies must be optimized to handle growing data efficiently, ensuring that results for standard queries (e.g., patient searches, inventory checks, or report generation) are returned in two seconds or less, even when processing 10,000+ records. This ensures both scalability and reliability as the care facility expands.

### 5.2 Legal and Compliance Requirements

The system must comply with the Philippine Data Privacy Act of 2012 (RA 10173), which requires strict measures in the collection, storage, and processing of sensitive data such as patient medical information and staff credentials. To achieve compliance, the system must implement role-based access control (RBAC), ensuring that only administrators can manage user accounts and view or edit critical system logs. Regular staff members will be restricted to operational functions such as managing patients, inventory, dispensing, and generating reports. Additionally, the system must employ industry-standard IT security practices, including:

Password hashing and salting to protect user credentials.

HTTPS/SSL encryption to secure data transmission over networks.

Secure session handling to prevent unauthorized access via session hijacking.

By adhering to both government regulations and IT best practices, the system guarantees the confidentiality, integrity, and availability of all sensitive data.

### **5.3 Reuse Objectives**

The AngelCare system shall be built with modularity and scalability as guiding principles. Each of its major modules—authentication, reporting, notification, and dispensing management—shall be designed as reusable components that can be integrated into future systems or adapted to suit other organizations. For example, the authentication module can be repurposed for other healthcare applications, while the reporting framework could be applied in inventory or financial systems. On the hardware side, components such as Arduino boards, ESP32 Wi-Fi modules, and stepper motors are to be configured in a way that allows them to be easily adapted or repurposed for similar use cases, such as automated inventory systems in pharmacies, hospitals, or even non-healthcare industries requiring precise dispensing. This approach ensures the longevity, adaptability, and cost-effectiveness of the system as both software and hardware can be extended to larger deployments or entirely new domains.

### **5.1 5.4 Internationalization Requirements**

To maintain consistency and clarity across records, the system shall enforce a DD/MM/YYYY date format and a 24-hour time format for all displayed dates and times within the user interface, notifications, and reports. This avoids ambiguity in recording medication schedules or event logs.

## Appendix A – Test Plan/Test Cases

Table 16. Test Case Table

Test Case ID	Description	Steps to Execute	Expected Outcome	Actual Outcome	Status (Pass/Fail)
TC001	Staff Login	1. Staff enters valid credentials at <code>angelcare.space/login.php</code> . 2. System verifies credentials. 3. Redirects staff to dashboard.	Dashboard loads with staff-specific access.	Works as expected.	Pass
TC002	Admin Login	1. Admin enters valid admin credentials. 2. System verifies role. 3. Redirects admin to dashboard with admin-only access.	Admin dashboard loads with User Management and Logs access.	Works as expected.	Pass
TC003	Dashboard Display	1. Staff logs in. 2. Dashboard loads showing metrics: patients, stock, dispenser status, upcoming schedules, activities, notifications.	Dashboard shows accurate, real-time data.	Works as expected.	Pass
TC004	Add New Patient	1. Staff navigates to Patients page. 2. Clicks <b>Add Patient</b> . 3. Inputs valid data and saves.	Patient record successfully stored in database.	Works as expected.	Pass
TC005	Add Medicine to Inventory	1. Staff goes to Inventory page. 2. Adds new medicine with details (batch, expiry, stock). 3. Saves.	Medicine appears in inventory list and linked to container slots.	Works as expected.	Pass
TC006	Manual Dispense from Dispenser Page	1. Staff navigates to Dispenser page. 2. Selects container and clicks <b>Dispense</b> . 3. Hardware executes command.	Container rotates, medicine is released, confirmation logged.	Works as expected.	Pass
TC007	Scheduled Dispensing	1. Staff creates schedule for patient. 2. At set time, backend triggers dispense. 3. Hardware executes and logs event.	Medicine dispensed automatically, log updated.	Works as expected.	Pass
TC008	Missed Medicine Notification	1. Patient misses scheduled intake. 2. After 30 minutes, system flags missed dose.	Notification logged in Notifications page.	Works as expected.	Pass
TC009	Generate Reports	1. Staff navigates to Reports page. 2. Selects report type (usage, dispensing, staff	Report generated and downloaded	Works as expected.	Pass

		activity). 3. Exports as PDF/CSV.	within 5 seconds.		
<b>TC010</b>	User Management (Admin)	1. Admin navigates to User Management page. 2. Creates new staff account.	New user credentials saved and usable for login.	Works as expected.	Pass
<b>TC011</b>	Logs Monitoring (Admin)	1. Admin opens Logs page. 2. Views system activities (dispense events, login attempts, changes).	Logs displayed with accurate timestamps.	Works as expected.	Pass

## SUMMARIZED TEST PLAN

### 1. BACKGROUND

AngelCare is a web-based inventory management system integrated with an automated medicine dispenser for Angel's Elderly Home Care Services. The system addresses manual inefficiencies in medicine tracking and dispensing, which often result in human error, delays, and compliance issues.

### 2. INTRODUCTION

This Test Plan defines the objectives, scope, approach, resources, and schedule of testing activities for the AngelCare system. It ensures that all system requirements—functional and non-functional—are validated before deployment to ensure safe, accurate, and efficient medicine dispensing.

### 3. ASSUMPTIONS

- Caregivers and staff possess basic computer literacy and will undergo training before system use.
- Stable internet connectivity will be available during tests, but basic offline logging is possible in case of disruptions.
- The dispenser prototype will be available for integration testing.
- Test data will simulate actual medicine inventory and patient schedules but will not use real patient data for privacy.

### 4. TEST ITEMS

- Web-based Inventory Management System (Frontend & Backend)

- Automated Medicine Dispenser (Arduino-powered hardware)
- Notification and Alerts Module
- Reports and Analytics Module
- User Authentication and Role-based Access Control

## 5. FEATURES TO BE TESTED

- **Login & Authentication:** Secure access with role-based control.
- **Patient Management:** Add, edit, delete patient profiles with medication schedules.
- **Inventory Management:** Add, update, delete medicine records; real-time stock monitoring.
- **Dispenser Control:** Automated and manual dispensing of medicines.
- **Scheduling & Monitoring:** Real-time medicine intake tracking and missed dose detection.
- **Notifications:** Low-stock alerts, missed-dose alerts, upcoming schedules.
- **Reports:** Monthly usage, dispensing history, and stock summaries.

## 6. FEATURES NOT TO BE TESTED

- Integration with external pharmacies or vendors.
- Billing, payment processing, or financial modules (out of scope).
- Mobile app or SMS notifications (not included in this version).
- Support for liquid/injectable medicines.

## 7. APPROACH

- **Testing Philosophy:** Combination of manual testing, unit tests, and simulation of hardware integration.
- **Data Flows:** Test medicine inventory creation, scheduling, dispenser execution, and logging.
- **Execution Mode:** Both simulation (mock data, software-only test) and live execution (integrated with dispenser prototype).

## 8. ITEM PASS/FAIL CRITERIA

- **Pass:** Feature works as specified in the SRS (e.g., correct dose dispensed within  $\pm 2$  seconds of schedule).
- **Fail:** Feature deviates from requirements (e.g., incorrect medicine, failed notification).

Itemized criteria examples:

- Inventory updates reflect real-time stock levels (100% match).
- Dispenser accuracy rate  $\geq 98.8\%$ .
- Low-stock notifications triggered when threshold is reached.

## 9. SUSPENSION/RESUMPTION CRITERIA

- **Suspension:** Test may be suspended if the hardware dispenser fails, internet outage occurs, or critical bugs block test execution.
- **Resumption:** Testing may resume once issues are resolved and checkpoints (logged test cases) are revalidated.

## 10. TEST DELIVERABLES

- Test Cases & Scripts
- Test Results Log
- Bug/Defect Reports
- Final Test Report & Recommendation
- Verified Prototype with User Acceptance Test (UAT) results

### 5.1.1 11. TESTING TASKS

- Functional tasks:
  - Setup of dispenser prototype and system database
  - Execution of unit, integration, and system tests
  - User acceptance testing with caregivers
- Administrative tasks:
  - Documentation of test cases and reports
  - Scheduling test sessions with stakeholders

## 12. ENVIRONMENTAL NEEDS

- **Hardware:** Arduino Mega, stepper motors, LCD, ESP32, medicine dispenser prototype.
- **Software:** PHP, MySQL, XAMPP, local host, browser access.
- **Other:** Stable internet, laptops/desktops, training environment.

## 13. RESPONSIBILITIES

- **Project Manager (Oliveros):** Oversees test planning and approval.
- **Lead Programmer (Castelltort):** Prepares and executes technical test cases.
- **QA Analyst (Etom):** Designs test cases, tracks defects, ensures quality standards.
- **UI/UX Designer & Hardware Engineer (Leyson):** Ensures dispenser hardware integration and usability compliance.
- **Client (Ms. Margelyn Gabbac):** Provides acceptance feedback.
- **Project Adviser (Ms. Kris P. Garcia)** Provides insights to the team members
- **Subject Coordinator (Ms. Mikee Gonzaga-Obing)** Guides the team and gives suggestions to the project.

## 14. STAFFING & TRAINING

- Four-member student development team (Karton Tech Solutions).
- Training sessions for caregivers (2-day on-site, manuals + live demo).

## 15. SCHEDULE

- **System Development:** June 17 – Sept 30, 2025
- **User Testing:** Oct 1 – Oct 7, 2025
- **Deployment & Training:** Oct 7 – Oct 8, 2025
- **Final Approval:** Oct 10 – Oct 18, 2025

## 5.1.2 16. RESOURCES

- **Budget:** ₱1,833,726 (includes labor, hardware, training, documentation).
- **Tools:** Arduino IDE, PHPMyAdmin, browsers, Git for version control.



## **17. RISKS & CONTINGENCIES**

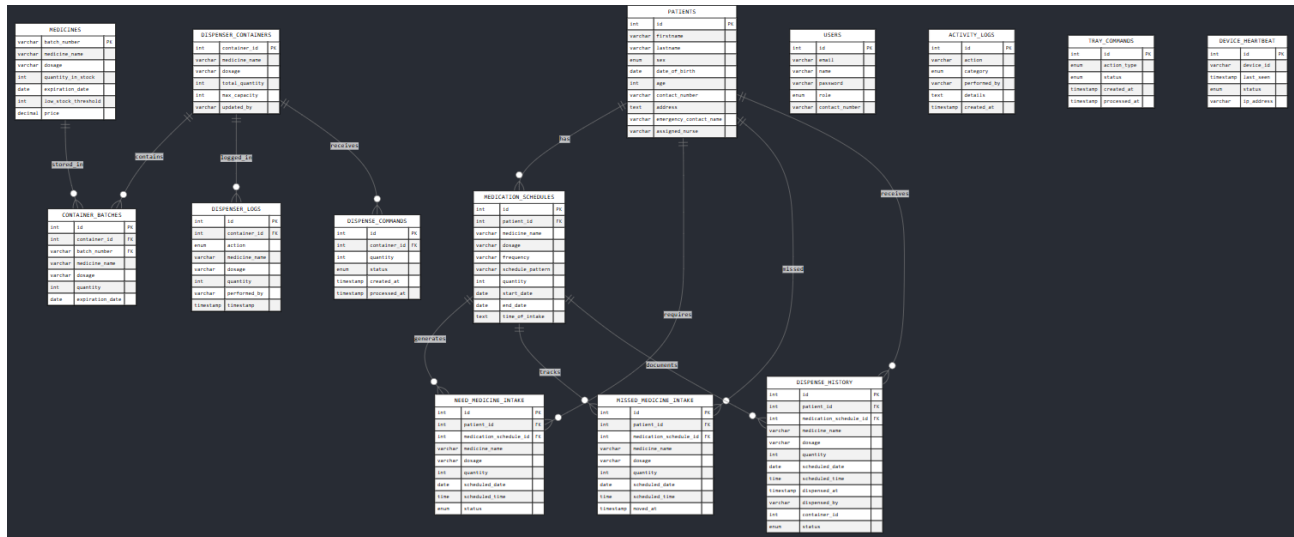
- **Power outages** → Use UPS backup.
- **Internet downtime** → Allow offline data logging, sync when online.
- **Hardware malfunction** → Spare components, maintenance checks.
- **User adoption issues** → Full training and documentation.
- **Data security breaches** → Role-based authentication & encryption.

## **18. APPROVALS**

- Client: Ms. Margelyn Gabbac (Owner, Angel's Elderly Home Care Services)
- Faculty Adviser: Ms. Kris P. Garcia
- Instructor: Ms. Mikee Gonzaga-Obing
- Development Team: Karton Tech Solutions

# Appendix B – Data Dictionary/ERD

## ENTITY RELATIONSHIP DIAGRAM:



## Curriculum Vitae

# CASTELLTORT, SEVAZTIENE IZACH

BSIT-MI Student

### PROFILE

Information Technology student at National University Fairview with a strong foundation in software development, network troubleshooting, and project management. Demonstrates proficiency in technical documentation, team collaboration, and information security. Skilled in multitasking, report-writing, presentation, and document processing. Experienced in web development and active listening, with a proven ability to make informed decisions and lead teams effectively. Aspiring to leverage these skills in a dynamic IT environment to drive innovation and efficiency.

### HOBBIES

- Working Out
- Playing Games
- Making song covers
- Sleeping

### EDUCATION

#### National University Fairview

2022 - Present

- Deans Lister (1<sup>st</sup> Year, 2<sup>nd</sup> Year, 3<sup>rd</sup> Year)

#### Mater Carmeli School of Novaliches

2019 - 2021

- With Honors
- Best in Performing Arts

### WORK EXPERIENCE

None - Present

### SKILLS

- Software development
- Network Trouble Shooting
- Project Management
- Technical Documentation
- Team Collaboration
- Information Security
- Multitasking
- Report-Writing
- Presentation Preparation
- Document Processing
- Web Development
- Active Listening
- Decision-Making
- Team Leadership

# LEYSON, VLADIMIR D.

BSIT-MI Student

## PROFILE

A highly motivated student with a strong foundation in web development, networking, and cloud computing. Certified in Generative AI and Oracle Cloud, with proven skills in problem-solving, communication, and teamwork. Experienced in administrative tasks, customer service, and multimedia editing.

## HOBBIES

- Playing Games
- Video Editing
- Sleep

## EDUCATION

### National University Fairview (College)

2022 - Present

- Deans Lister (1<sup>st</sup> Year, 2<sup>nd</sup> Year, 3<sup>rd</sup> Year)

### Loyola College of Culion, Culion, Palawan

2019 - 2021

- TECH-VOC Track Home Economics (Tourism Services) Strand
- Graduated with Honors

## WORK EXPERIENCE

### Work Immersion 2021-2021 – PNP Culion Municipality Police Station, Culion, Palawan

- Gained hands-on experience in basic law enforcement procedures, administrative tasks, and community policing initiatives

## SKILLS

- Basic Web Development
- Networking Basics
- Database Management
- Problem-Solving
- Communication Skills
- Teamwork and collaboration
- Time Management
- Customer Service
- Basic Photo and video editing
- Basic Cinematography
- Basic Drone photo and Cinematography

# ETOM, EIAN L.

BSIT-MI Student

## PROFILE

Dedicated IT Student with a passion for software development and web technologies. Proficient in [e.g.](#) Basic Java HTML/CSS and experienced with database management and cloud computing. Quick learner with strong problem-solving skills, seeking opportunities to apply knowledge in a professional environment.

## HOBBIES

- Playing Games
- Video Editing
- Sleep

## EDUCATION

### National University Fairview (College)

2022 - Present

- Deans Lister (1<sup>st</sup> Year, 3<sup>rd</sup> Year)

### Lagro High School (Senior High School)

2019 - 2021

- Humanities and Social Sciences (HUMSS)
- Graduated with Honors

## WORK EXPERIENCE

- None - Present

## SKILLS

- Basic Java
- Basic Python
- HTML, CSS
- Database
- Android Studio
- Basic Networking

## References:

- Jacobson, I., Booch, G., & Rumbaugh, J. (2005). *The Unified Modeling Language User Guide* (2nd ed.). Addison-Wesley.
- Gomaa, H. (2004). *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison-Wesley.
- IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)*. IEEE.
- IEEE Computer Society. (2018). *IEEE Systems and Software Engineering – Life Cycle Processes – Requirements Engineering (IEEE Std 29148-2018)*. IEEE.
- OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks*. Retrieved from <https://owasp.org/Top10>
- Republic of the Philippines. (2012). *Data Privacy Act of 2012 (RA 10173)*. Official Gazette. Retrieved from <https://www.privacy.gov.ph/data-privacy-act/>
- MySQL. (2024). *MySQL 8.0 Reference Manual*. Oracle. Retrieved from <https://dev.mysql.com/doc/>
- phpMyAdmin. (2024). *phpMyAdmin Documentation*. phpMyAdmin Project. Retrieved from <https://www.phpmyadmin.net/docs/>
- Arduino. (2024). *Arduino Uno Rev3*. Arduino Official. Retrieved from <https://store.arduino.cc/products/arduino-uno-rev3>
- Arduino. (2024). *Arduino Mega 2560 Rev3*. Arduino Official. Retrieved from <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- Espressif Systems. (2023). *ESP32 Technical Reference Manual*. Retrieved from <https://www.espressif.com/en/products/socs/esp32>
- Lewis, J., & Fowler, E. (2019). *Software Development Best Practices: COMET Method*. *Journal of Software Engineering Practice*, 12(3), 101–112.
- Leveson, N. (1995). *Safeware: System Safety and Computers*. Addison-Wesley.

*Angel's Elderly Home Care Services. (2024). Stakeholder Interviews and Operational Workflow. Internal documentation.*

*Karton Tech Solutions. (2024). AngelCare: Web-Based Inventory Management System with Integrated Medicine Dispenser – SOFTDEV Paper. Unpublished student paper.*