HOLY ANGEL UNIVERSITY
Object-Oriented Programming
School Year 2022-2023
1st Semester

Title:
Centralized Hospital System

Prepared by:
GOZUN JUSTINE CARL M.
Yumul, Kohl Daniel Q.
Oliver, Ira Kurt L.
Herrera, Daniel Gio

Prepared for:
De Jesus, Arnaz P.

Date
June 24, 2025

Team Profile

Oliver, Ira Kurt L.
Research

Yumul, Kohl Daniel Q.
Coding
Research

Gozun Justine Carl M.
Research
Coding

Herrera, Daniel Gio
Research

Introduction

Our Centralized Hospital System is designed to make hospital management easier and more efficient. Role-based access provide secure logins for admins, staff, and patients, ensuring everyone has the right permissions. Pharmacy management helps staff manage hospitals, process medication orders, and track inventory smoothly. Medical supplies and equipment are tracked regularly to prevent shortages and improve distribution. Staff can generate reports on patient records, finances, and resources for better decision-making. Patients can safely view their own health records anytime, reducing paperwork and wait times. With our system, hospitals can work smarter, faster, and with fewer errors.

Project’s Objectives

Our Centralized Hospital System aims to streamline healthcare operations by integrating key functionalities into a unified platform. Designed to enhance efficiency and accessibility, the system supports user accounts with distinct roles (Admin, Staff, and Patient), enabling tailored access to features. Admins can manage hospitals, while staff oversee pharmacy operations, inventory, and generate reports for analytics. Patients benefit from a dedicated hub to view their records. By consolidating hospital management, medication dispensing, stock tracking, and data reporting, the system ensures accurate, secure, and coordinated care delivery, ultimately improving patient outcomes and operational transparency.

Application Features and Description

Features	Description
User accounts with specific roles	User roles: Admin, Staff, Patient
Hospitals	The admin can add a hospital featuring, pharmacy management, stock management, and records.
Pharmacy Management	The staff manages medication orders, dispensing, and inventory, ensuring safe and accurate medication administration
Inventory/Stock Management	The staff can track the number of medical supplies, pharmaceuticals, and equipment, ensuring efficient procurement and distribution
Updates, reporting and analytics	The staff has access to reports on various aspects of hospital operations, including patient data, financial performance, and resource utilization. The staff can provide records of a patient
Patient's hub	The patient can view his/her own records provided/posted by the staff. A patient can also comment in the post

Methods/Approach

To develop the Centralized Hospital System, we adopt a modular, full-stack approach leveraging Django's Model-View-Template (MVT) architecture. Our methodology prioritizes security, scalability, and user-centric design, implemented through the following structured workflow:

1. Technology Stack

Backend: Django (Python) for robust server-side logic, authentication, and database management.

Frontend: HTML/CSS for responsive interfaces, extended with Django Templates for dynamic content rendering.

Database: SQLite (development) and PostgreSQL (production) for relational data storage.

2. Agile Development Process

We follow iterative Agile sprints (2-week cycles) to:

Prioritize high-impact modules (e.g., role-based access, pharmacy management).

Incorporate stakeholder feedback via prototypes.

Ensure continuous integration/deployment (Git/GitHub).

3. Core System Modules & Implementation

Role-Based Access Control (RBAC):

Use Django's built-in authentication system with custom user models.

Define granular permissions via Django Groups (e.g., Admin, Staff, Patient).

Secure APIs with session-based authentication and CSRF tokens.

Pharmacy & Inventory Management:

Implement Django Models (Inventory, MedicationOrder) to track stock levels, expiration dates, and orders.

Automate re-alerts for low stock using Django Signals.

Medical Equipment Tracking:

Design an Equipment model with fields for location, maintenance logs, and status.

Integrate QR-code scanning via Python libraries (qrcode, OpenCV) for quick updates.

Reporting Engine:

Generate PDF/financial reports with ReportLab and Django ORM aggregations.

Visualize resource usage via Chart.js (embedded in HTML templates).

Patient Portal:

Develop a secure patient dashboard (HTML/Django Templates) to view records.

Encrypt sensitive health data using Django's cryptography library.

4. Database Design

Relational schema with optimized tables (e.g., Patients, Appointments, Inventory).

One-to-many relationships (e.g., one Staff processes many MedicationOrders).

Index critical fields (e.g., medication_id, patient_national_id) for fast querying.

5. Security & Compliance

Apply HTTPS, input sanitization, and Django's XSS/SQL injection protections.

Store passwords as hashes (PBKDF2) and audit logs for data access (HIPAA-aligned).

6. Testing & Validation

Unit Tests: Django's test framework for models, views, and RBAC logic

User Acceptance Testing (UAT): Pilot trials with hospital staff to refine workflows.

Work Plan

OBJECTIVE	**TIME FRAME**	**ACTIVITIES**
Learning the basics	Week 1-3	Learn the basics of coding with the use of Django and Python
Learning application	Week 4-7	Apply the learnings
Coding, development phase	Week 8-12	The making of the website
Testing	Week 13-16	Debugging, testing

References